

**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE**  
**ENGINEERING AND TECHNOLOGY**

**A CONTROL AND AUTOMATION ENGINEERING APPROACH TO  
RAILWAY INTERLOCKING SYSTEM DESIGN**

**Ph.D. THESIS**

**Mustafa Seçkin DURMUŞ**

**Department of Control and Automation Engineering**

**Control and Automation Engineering Programme**

**APRIL 2014**



**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE**  
**ENGINEERING AND TECHNOLOGY**

**A CONTROL AND AUTOMATION ENGINEERING APPROACH TO  
RAILWAY INTERLOCKING SYSTEM DESIGN**

**Ph.D. THESIS**

**Mustafa Seçkin DURMUŞ**  
**(504062109)**

**Department of Control and Automation Engineering**

**Control and Automation Engineering Programme**

**Thesis Advisor: Prof. Dr. Mehmet Turan SÖYLEMEZ**

**APRIL 2014**



**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**DEMİRYOLU ANKLAŞMAN SİSTEM TASARIMINA KONTROL VE  
OTOMASYON MÜHENDİSLİĞİ YAKLAŞIMI**

**DOKTORA TEZİ**

**Mustafa Seçkin DURMUŞ  
(504062109)**

**Kontrol ve Otomasyon Mühendisliği Anabilim Dalı**

**Kontrol ve Otomasyon Mühendisliği Programı**

**Tez Danışmanı: Prof. Dr. Mehmet Turan SÖYLEMEZ**

**NİSAN 2014**



**Mustafa Seçkin DURMUŞ**, a **Ph.D.** student of ITU **Graduate School of Science Engineering and Technology** student ID **504062109**, successfully defended the thesis entitled “**A CONTROL AND AUTOMATION ENGINEERING APPROACH TO RAILWAY INTERLOCKING SYSTEM DESIGN**”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

**Thesis Advisor :**    **Prof. Dr. Mehmet Turan SÖYLEMEZ** .....  
Istanbul Technical University

**Jury Members :**    **Prof. Dr. Leyla GÖREN-SÜMER** .....  
Istanbul Technical University

**Prof. Dr. Galip CANSEVER** .....  
Yıldız Technical University

**Prof. Dr. Salman KURTULAN** .....  
Istanbul Technical University

**Assoc. Prof. Dr. Tankut ACARMAN** .....  
Galatasaray University

**Date of Submission : 31 January 2014**

**Date of Defense : 10 April 2014**





*To my loved ones,*



## **FOREWORD**

I would like to thank the following people because without their help and support this Ph.D. thesis would not have been possible. Firstly, I would like to show my gratitude to my supervisor Prof. Mehmet Turan Söylemez and my thesis progress jury Prof. Leyla Gören-Sümer and Prof. Galip Cansever for their suggestions, encouragements and guidance in approaching to different challenges during the progress in this thesis.

I would also like to thank Asst. Prof. Engin Yeşil, Asst. Prof. İlker Üstoğlu, Asst. Prof. Özgür Kaymakçı, Asst. Prof. Kürşat Yalçın, Asst. Prof. Tufan Kumbasar and Res. Asst. Kemal Uçak for their support that has helped me to improve the quality of this thesis.

I am grateful to all the people from Department of Control and Automation Engineering, Istanbul Technical University, for being a family during the many years I spent in Istanbul.

Finally, I would like to thank my parents, my friends and my dearest wife Özlem for their constant support during the time I studied.

**January 2014**

Mustafa Seçkin DURMUŞ  
(Electrical & Electronic Engineer, M.Sc.)



## TABLE OF CONTENTS

	<u>Page</u>
<b>FOREWORD</b> .....	<b>ix</b>
<b>TABLE OF CONTENTS</b> .....	<b>xi</b>
<b>ABBREVIATIONS</b> .....	<b>xiii</b>
<b>LIST OF TABLES</b> .....	<b>xv</b>
<b>LIST OF FIGURES</b> .....	<b>xvii</b>
<b>SUMMARY</b> .....	<b>xix</b>
<b>ÖZET</b> .....	<b>xxi</b>
<b>1. INTRODUCTION</b> .....	<b>1</b>
1.1 Purpose of Thesis .....	2
1.2 Structure of the Thesis.....	3
1.3 Contribution of the Thesis .....	3
<b>2. FIXED-BLOCK SIGNALING SYSTEMS</b> .....	<b>5</b>
2.1 History of the Fixed-Block Signaling Systems .....	5
2.2 Components of the Fixed-Block Signaling Systems .....	6
2.2.1 The traffic control center.....	6
2.2.2 The interlocking system .....	6
2.2.3 Point machines (railway switches).....	7
2.2.4 Signals (optical wayside signals) .....	7
2.2.5 Track circuits and axle counters.....	8
2.3 Influence of Functional Safety Standards on Fixed-Block Signaling Systems..	9
2.4 Case Study: Turkish National Railway Signaling Project .....	12
2.4.1 Automaton as a modeling tool.....	16
2.4.2 Petri nets as a modeling tool .....	17
2.4.2.1 Petri net models of the railway field components .....	19
2.5 Voting Strategy .....	24
2.5.1 Synchronization problems.....	27
2.5.2 Solutions for the synchronization problems.....	28
2.5.3 Signaling software (interlocking) design for an example railway field ...	29
2.6 Testing the signaling software.....	38
2.6.1 Signaling software module tests (Verification) .....	38
2.6.2 Signaling software simulator tests (Validation).....	38
2.6.3 Hardware simulator tests (Commissioning).....	38
2.6.4 Factory acceptance tests (FAT).....	39
2.6.5 Site acceptance tests (SAT).....	39
2.7 Results and Discussion.....	39
<b>3. MOVING-BLOCK SIGNALING SYSTEMS</b> .....	<b>41</b>
3.1 European train control system (ETCS) .....	41
3.1.1 Application Level 0.....	42
3.1.2 Application level 1 .....	42
3.1.3 Application level 2 .....	42

3.1.4 Application level 3 .....	42
3.2 GSM for railways (GSM-R).....	43
3.3 Calculating the train braking distance .....	43
3.3.1 Train braking distance calculation .....	45
3.4 Proposed speed control method for moving-block systems .....	50
3.4.1 Suggested adaptive PD controller based on online LSSVR.....	51
3.4.2 Simulation results.....	56
3.4.2.1 Noiseless case.....	57
3.4.2.2 Noisy case with disturbances .....	60
3.5 Results and Discussion .....	63
<b>4. CONCLUSION.....</b>	<b>65</b>
<b>REFERENCES .....</b>	<b>67</b>
<b>CURRICULUM VITAE .....</b>	<b>75</b>

## ABBREVIATIONS

<b>APN</b>	: Automation Petri net
<b>BD</b>	: Braking Distance
<b>CBTC</b>	: Communication Based Train Control
<b>CENELEC</b>	: European Committee for Electrotechnical Standardization
<b>COTS</b>	: Commercial Off-The-Shelf
<b>DES</b>	: Discrete Event System
<b>DMI</b>	: Driver Machine Interface
<b>EBD</b>	: Emergency Brake Deceleration
<b>EBI</b>	: Emergency Brake Invention
<b>E/E/EP</b>	: Electrical/Electronic/Programmable Electronic
<b>EIRENE</b>	: European Integrated Radio Enhanced Network
<b>EOA</b>	: End of Authority
<b>ERA</b>	: European Railway Agency
<b>ERTMS</b>	: European Rail Traffic Management System
<b>ETCS</b>	: European Train Control System
<b>FAT</b>	: Factory Acceptance Tests
<b>GSM</b>	: Global System for Mobile Communications
<b>GSM-R</b>	: GSM for Railways
<b>ITU</b>	: Istanbul Technical University
<b>LSSVR</b>	: Least Squares Support Vector Regression
<b>MA</b>	: Movement Authority
<b>MORANE</b>	: Mobile Radio for Railways Networks in Europe
<b>MSB</b>	: Moving Space Block
<b>MTB</b>	: Moving Time Block
<b>MTTF</b>	: Mean Time to Failure
<b>NARX</b>	: Nonlinear AutoRegressive eXogenous
<b>PLC</b>	: Programmable Logic Controller
<b>PMB</b>	: Pure Moving Block
<b>PN</b>	: Petri net
<b>RBC</b>	: Radio Block Center
<b>SAT</b>	: Site Acceptance Tests
<b>FD</b>	: Following Distance
<b>SIL</b>	: Safety Integrity Level
<b>SNR</b>	: Signal to Noise Ratio
<b>TCC</b>	: Traffic Control Center
<b>TCDD</b>	: Turkish State Railways
<b>TNRSP</b>	: Turkish National Railway Signaling Project
<b>TMR</b>	: Triple Modular Redundancy
<b>TUBITAK</b>	: The Scientific and Technological Research Council of Turkey
<b>UIC</b>	: International Union of Railways
<b>UNIFE</b>	: The European Rail Industry
<b>UNISIG</b>	: Union Industry of Signaling





## LIST OF TABLES

	<u>Page</u>
<b>Table 2.1</b> : The types and the definitions of common signals.....	8
<b>Table 2.2</b> : Relationship between SIL, $\lambda$ and MTTF.....	10
<b>Table 2.3</b> : Definitions of places and transitions in figure 2.12. ....	20
<b>Table 2.4</b> : Definitions of places and transitions in figure 2.13. ....	21
<b>Table 2.5</b> : Definitions of places and transitions in figure 2.14. ....	23
<b>Table 2.6</b> : Definitions of places and transitions in figure 2.15. ....	23
<b>Table 2.7</b> : The safe-states of the variables. ....	32



## LIST OF FIGURES

	<u>Page</u>
<b>Figure 1.1</b> : Typical emission and energy consumption. ....	1
<b>Figure 2.1</b> : General block diagram of the fixed-block signaling system. ....	6
<b>Figure 2.2</b> : Schematic representation of a railway switch. ....	7
<b>Figure 2.3</b> : Movement of trains in a single railway line (fixed-block). ....	9
<b>Figure 2.4</b> : (a) Unoccupied track circuit, (b) Occupied track circuit. ....	9
<b>Figure 2.5</b> : The V-model. ....	11
<b>Figure 2.6</b> : A sample railway field. ....	12
<b>Figure 2.7</b> : The interlocking table of the railway field given in figure 2.6. ....	13
<b>Figure 2.8</b> : Determination of SIL. ....	13
<b>Figure 2.9</b> : The general framework of the railway signaling system. ....	16
<b>Figure 2.10</b> : An example automaton. ....	17
<b>Figure 2.11</b> : An example <i>PN</i> . ....	18
<b>Figure 2.12</b> : Route reservation model of a single route. ....	20
<b>Figure 2.13</b> : A single PM model. ....	21
<b>Figure 2.14</b> : Signal models (Four-aspect tall signal (a), Three-aspect tall signal (b), Two-aspect dwarf signal (c), Three-aspect dwarf signal (d)). ....	22
<b>Figure 2.15</b> : Railway block model. ....	23
<b>Figure 2.16</b> : The type 1 problem. ....	27
<b>Figure 2.17</b> : Situation where the type 1 problem do not cause any failure (a, b), another situation where the type 1 problem causes a failure (c). ....	27
<b>Figure 2.18</b> : The type 2 problem. ....	28
<b>Figure 2.19</b> : A possible solution for the type 1 problem. ....	28
<b>Figure 2.20</b> : Possible solutions for the type 2 problem. ....	29
<b>Figure 2.21</b> : The general representation of the interlocking software. ....	30
<b>Figure 2.22</b> : The sequence diagram for reservation of the route A-B. ....	33
<b>Figure 2.23</b> : An example sequence diagram for the type 1 problem. ....	35
<b>Figure 2.24</b> : The sequence diagram of possible solution for the type 1 problem. ...	36
<b>Figure 2.25</b> : An example data sequence diagram for the type 2 problem. ....	37
<b>Figure 2.26</b> : The data sequence diagram of possible solution for the type 2 problem. .....	37
<b>Figure 3.1</b> : ERTMS application levels from 1 to 3. ....	43
<b>Figure 3.2</b> : ERTMS communication architecture. ....	45
<b>Figure 3.3</b> : Average gradient concept and limitations [60]. ....	46
<b>Figure 3.4</b> : Fixed-block vs. moving-block. ....	48
<b>Figure 3.5</b> : Stopping distance calculation for a high-speed train. ....	49
<b>Figure 3.6</b> : Online adaptive PD controller based on LSSVR. ....	53
<b>Figure 3.7</b> : The change of velocity ( $V_{Rf}$ : Velocity of the rear train, $V_L$ : Velocity of the leading train) and acceleration value of the rear train ( $a_R$ ) with respect to time. ....	57

<b>Figure 3.8 :</b> The change of velocity ( $V_{Rr}$ : Velocity of the rear train, $V_L$ : Velocity of the leading train) and acceleration value of the rear train ( $a_R$ ) with respect to position.....	58
<b>Figure 3.9 :</b> The change of FD, BD and the difference between FD and BD with respect to time.....	58
<b>Figure 3.10 :</b> The change of FD, BD and the difference between FD and BD with respect to position.....	59
<b>Figure 3.11 :</b> $K_p$ , $K_d$ values of the PD controller.....	59
<b>Figure 3.12 :</b> The change of velocity ( $V_L$ : Velocity of the leading train, $V_{Rm}$ : Measured velocity of the rear train, $V_{Rr}$ : Output velocity of the rear train) and acceleration ( $a_R$ ) with respect to time.....	60
<b>Figure 3.13 :</b> The change of velocity ( $V_L$ : Velocity of the leading train, $V_{Rm}$ : Measured velocity of the rear train, $V_{Rr}$ : Output velocity of the rear train) and acceleration ( $a_R$ ) with respect to position.....	61
<b>Figure 3.14 :</b> The change of velocity ( $V_L$ : Velocity of the leading train, $V_{Rm}$ : Measured velocity of the rear train, $V_{Rr}$ : Output velocity of the rear train) and acceleration ( $a_R$ ) with respect to time.....	61
<b>Figure 3.15 :</b> The change of FD, BD and the difference between FD and BD with respect to time.....	62
<b>Figure 3.16 :</b> The change of FD, BD and the difference between FD and BD with respect to position.....	62

# **A CONTROL AND AUTOMATION ENGINEERING APPROACH TO RAILWAY INTERLOCKING SYSTEM DESIGN**

## **SUMMARY**

Despite the high initial costs of railway constructions, railway systems are more economic, safer and more environment friendly than other ways of transport. Notwithstanding all these positive features, the investments in Turkey had been rather limited in comparison with other European countries until recent years. A need for reducing the dependency to other countries has arisen in line with the development in railway transportation after a visible increase in the investments on railway sector in recent years. Owing to high costs and problems in adaptation, development of local signaling systems has been required by TCDD. As a result, the first signaling system has been developed by the partnership of TCDD, TUBITAK and ITU.

The most important component of railway systems that enables a safe transportation is interlocking. The safety basis of developing an interlocking system is described by standards developed by international committees like CENELEC. In order to provide the required SIL, using these techniques, methods and architectures has a great importance. Besides the international safety standards, the needs and safety rules of the country where the signaling system to be applied have to be considered.

In this thesis, functional safety requirements related to fixed-block railway signaling systems are described, and formal modeling methods and software architectures used for a minimum SIL 3 railway interlocking system has discussed in detail. In particular, some of the problems that arise in using the diverse programming technique, which is used in developing failsafe software, have been determined and solution methods to these problems have been proposed. .

Besides fixed-block signaling systems, moving-block signaling systems are also explained within the thesis. The implementation levels of European Rail Traffic Management System (ERTMS) are explained, and application of an online adaptive controller design method that guarantees the trains to follow each other within safe distances is proposed.



## DEMİRYOLU ANKLAŞMAN SİSTEM TASARIMINA KONTROL VE OTOMASYON MÜHENDİSLİĞİ YAKLAŞIMI

### ÖZET

Demiryollarının ilk inşaat maliyetleri oldukça yüksek olmasına rağmen diğer ulaşım sistemleri ile karşılaştırıldığında, demiryolu sistemleri daha ekonomik, güvenli ve çevrecidir. Tüm bu özelliklerine rağmen, diğer Avrupa ülkeleri ile karşılaştırıldığında yakın zamana kadar Türkiye’de demiryollarına yapılan yatırım oldukça kısıtlı kalmıştır. Son yıllarda demiryolu sektörüne yapılan yatırımların gözle görülür bir şekilde artmasıyla gelişimi hız kazanan demiryolu ulaşım sistemlerinde dışarıya bağımlılığın azaltılması gündeme gelmiştir.

Bu bağlamda gerek uyum ve güncelleme sıkıntıları gerekse yüksek maliyetler nedeni ile yerli sinyalizasyon sistemlerinin geliştirilmesi Türkiye Cumhuriyeti Devlet Demiryolları (TCDD) tarafından bir ihtiyaç olarak belirtilmiştir. Bunun bir sonucu olarak Türkiye’nin ilk yerli demiryolu sinyalizasyon projesi TCDD, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) ve İstanbul Teknik Üniversitesi (İTÜ) ortaklığında tamamlanarak TCDD’ye teslim edilmiştir.

Demiryolu sistemlerinde ulaşım ve taşımanın güvenli olarak gerçekleştirilmesini sağlayan en önemli bileşen anklaşman (interlock) sistemidir. Anklaşman sisteminin geliştirilmesinde izlenilecek olan temel adımlar Avrupa Elektroteknik Standardizasyon Komitesi (European Committee for Electrotechnical Standardization - CENELEC) gibi uluslararası komiteler tarafınca hazırlanan güvenlik standartlarında tanımlanmıştır.

Geliştirilen sinyalizasyon sisteminin istenilen Güvenlik Bütünlüğü Seviyesi (Safety Integrity Level - SIL) seviyesini sağlayabilmesi için bu güvenlik standartlarıncı tavsiye edilen yöntem, teknik ve mimarilerin kullanılması yüksek önem arz etmektedir. Uluslararası güvenlik standartlarının gereksinimlerine ek olarak, sinyalizasyon sisteminin kurulacağı ülkeye ait ihtiyaçlar ve güvenlik kriterleri de göz önünde bulundurulmalıdır.

Yazılım geliştirme süreci başlangıcında yazılımdan beklenen çıktılar veya başka bir deyişle yazılım isterleri oluşturulmalıdır. Sonrasında güvenlik standartlarıncı tavsiye edilen yöntem ve mimarilerin istenilen SIL seviyesinin sağlanabilmesi için uygun bir şekilde seçilmesi gerekmektedir. Seçilen yöntem ve mimariler yazılım isterlerini eksiksiz sağlayacak şekilde tasarımı gerçekleştirecek olan grup tarafından yazılım geliştirme sürecinde kullanılmalıdır.

Yazılım geliştirme sürecinde tasarımı gerçekleştirilen yazılım blok ve alt blokları (veya modülleri) yazılım test grubu tarafından oluşturulan test prosedürüne göre test edilerek doğrulanmalıdır. Test prosedüründe uygulanan adımlar yazılım isterlerini doğrulayacak şekilde oluşturmalıdır. Yazılım geliştirme ve test süreçleri eksiksiz

şekilde tamamlandıktan sonra doğrulanmış yazılım Fabrika Kabul Testi ve Saha Kabul Testleri ile de doğrulanmalıdır. Yazılım testlerinde herhangi bir hata ile karşılaşılması durumunda hatalar raporlanmakta ve gerekli düzeltmelerin yapılması amacıyla yazılım geliştirme gruplarına sunulmaktadır. Gerekli düzeltmeler gerçekleştirildikten sonra tüm testler en baştan tekrar gerçekleştirilmelidir.

Bu tezde, sabit-blok (fixed-block) demiryolu sinyalizasyon sistemlerine ilişkin fonksiyonel güvenlik gereksinimleri tanımlanmış ve en az SIL 3 seviyesine sahip bir demiryolu anlaşılan sistemi tasarımında kullanılan biçimsel modelleme yöntemleri, yazılım mimarileri detaylı olarak anlatılmıştır.

Özellikle, güvenlik-kritik yazılım geliştirme sürecinde kullanılan çoklu programlama (Diverse Programming, N-version Programming) tekniğinin kullanımında ortaya çıkan eşzamanlama (synchronization) problemleri tanımlanmış ve bunlara yönelik çözüm yöntemleri önerilmiştir. Karşılaşılan eşzamanlama problemleri tip-1 ve tip-2 olarak iki farklı gruba ayrılmıştır. Bu iki farklı problem için iki farklı çözüm önerilmiş ve programlanabilir mantıksal kontrolörler üzerinde uygulanarak doğrulanmıştır.

1800'lü yılların ortalarından bugüne kadar kullanılmakta olan sabit-blok sinyalizasyon sistemleri, yolcu ve taşıma yoğunluğunun artması, trenler arası yolculuk sürelerinin (headway time) gerekenden yüksek olması nedeniyle özellikle metro sistemlerinde yerini hareketli-blok (moving-block) sistemlere bırakmaktadır. Hareketli-blok sistemlerin en bilinen örneği olan Haberleşme Tabanlı Tren Kontrolü (Communication Based Train Control - CBTC) uygulamaları ile mevcut metro ve şehir içi demiryolu hatları daha etkin ve verimli bir şekilde kullanılabilir hale gelmiştir.

Buna ek olarak, farklı Avrupa ülkelerinde uygulanan tren kontrol, sinyalizasyon yöntemleri ile güvenlik kriterlerinin tek bir çatı altında toplanması, bu kriter ve uygulamaların şehir içi şehirlerarası demiryolu hatlarında da kullanılabilmesi amacıyla Avrupa Raylı Ulaşım Yönetim Sistemi (European Rail Traffic Management System - ERTMS) tanımlanmıştır.

ERTMS, Avrupa Demiryolu Trafik Kontrolü (European Rail Traffic Control - ETCS) ve Demiryolu Mobil İletişim için Küresel Sistem (Global System for Mobile communications - Railway - GSM-R) uygulamalarının bir araya getirilmesi sonucunda oluşturulmuştur.

ETCS uygulama seviyesi 1 ve 2, ek güvenlik kriterleri getirilmiş olan sabit-blok sistemlerdir. ETCS uygulama seviyesi 3 ise seviye 1 ve 2'den farklı olarak hareketli-blok sistemler olarak tanımlanmaktadır. ETCS uygulama seviye 3'ün en önemli bir diğer avantajı ise, sabit-blok sistemlerde kullanılan yol boyu sinyallerin ve ray devrelerinin kaldırılmış olmasıdır. Güvenli tren hareketi için gereken tüm bilgiler tren üzeri bilgisayar ve trafik kumanda merkezi arasında GSM-R vasıtasıyla iletilmektedir.

Sabit-blok demiryolu sinyalizasyon sistemlerinde gerçekleştirilen çözüm ve uygulamalara ek olarak, hareketli-blok sinyalizasyon sistemlerinin temel bileşenleri ve kavramları da tez kapsamında açıklanmıştır. Hareketli-blok sistemlerde trenlerin



birbirini güvenli bir şekilde takip edebilmeleri için göz önünde bulundurulması gereken kriterler tanımlanmıştır.

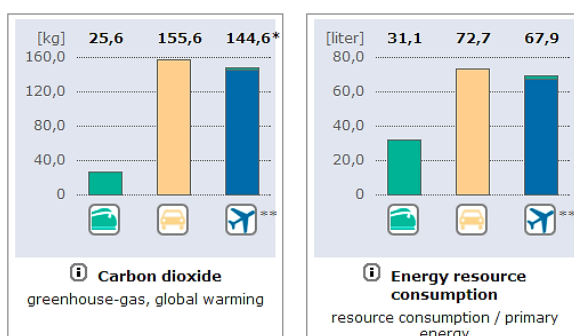
Avrupa Demiryolu Ajansı (European Railway Agency - ERA) ve Uluslararası Demiryolu Sendikası (International Union of Railways - UIC) gibi kurumların konu hakkında tavsiye ve önerilerine uygun olarak tren frenleme eğrilerinin hesaplanması ve güvenli tren takip mesafesi gibi kavramlar da açıklanmıştır.

ERA ve UIC tarafından tanımlanmış güvenli tren takip mesafeleri ve tren frenleme mesafeleri göz önünde bulundurularak, ETCS uygulama seviye 3 kriterlerine göre aynı demiryolu hattı üzerinde aynı yönde ilerleyen trenlerin birbirlerini güvenli olarak takip etmesini sağlayan bir çevrimiçi uyarlamalı kontrolör tasarım yönteminin uygulaması önerilmiştir. Önerilen bu kontrolörün doğrulanması amacıyla benzetim çalışmaları yapılmış ve kontrolörün doğruluğu gösterilmiştir.



## 1. INTRODUCTION

Railways provide more economical and environmental solutions in spite of the fact that their initial costs are relatively high. Among different transportation alternatives, for a passenger who travels from Rome to Paris, traveling by train causes less carbon dioxide emission and energy consumption as illustrated in figure 1.1 [1]. Researches also prove that the lifetime risk of death caused by railways is very low in comparison to different transportation alternatives.



**Figure 1.1** : Typical emission and energy consumption.

Such low risk values are ensured by strictly obeying the safety related standards such as CENELEC (European Committee for Electrotechnical Standardization) standards in Europe. For instance, the European standard IEC 61508 [2] is developed for functional safety requirements of all kinds of electronic and programmable devices. In addition to the umbrella standard IEC 61508, EN 50126 describes the functional safety requirements related with all kinds of railway applications where Reliability, Availability, Maintenance and Safety analysis (RAMS) are determined. EN 50128 (similar to EN 61508-3) determines methodologies for building software for railway control and applications and EN 50129 (similar to EN 61508-2) defines requirements for the hardware of electric, electronic and programmable devices to be used in railways.

Railway systems can be grouped as conventional (fixed-block) railway systems or moving-block railway systems. In fixed-block railway systems, the trains are moving according to route reservation procedure in fixed-length railway blocks, whereas in

moving-block railway systems, each train is regarded as a moving-block. Although, the use of fixed-block signaling in railway systems reduces the overall capacity and increases the headway time of the railway lines, it has been used widely the world since mid-1800s. As an alternative, the moving-block railway systems are introduced to increase the transportation capacity and to reduce the headway times. ERTMS, application level 3 can be given as an example of moving-block railway systems.

Developing signaling systems for unsignaled railway lines and the renovation of old (conventional) railway lines in Turkey has an increasing trend in past few years. Development of new railway lines and application of ERTMS are still ongoing. Because of these advantages and the rise in population, the demand on railways increases day by day.

## **1.1 Purpose of Thesis**

The main aim of this thesis is to examine the development process of an interlocking system for fixed-block railway signaling systems from the control and automation engineering point of view. For this purpose, the basic principles of railway signaling systems are explained. At first, the signaling systems are classified in to two main groups known as fixed-block and moving-block railway signaling systems. The influence of functional safety standards to fixed-block railway signaling systems and the use of recommended methods in the Turkish National Railway Signaling Project (TNRSP) are examined in the second section of the thesis. Especially, the software development process and the use of several programming techniques in determination of the software architectures are discussed. In particular, a new voting strategy is proposed for the solution of the problems encountered in the development of the TNRSP.

Moreover, basic concepts of moving-block signaling systems are given. The movement of trains in a single railway line is studied from the control and automation engineering point of view as a velocity and acceleration control problem. The use of an online adaptive controller design is proposed to solve this problem. Results of simulation studies are given to show the applicability of the proposed controller design.

## **1.2 Structure of the Thesis**

The structure of this thesis is as follows; after giving a short introduction, the main components of fixed-block signaling systems, related functional safety standards and the use of them in TNRSP, signaling software (interlocking software) design, the proposed voting strategy, development of interlocking software for a sample railway field and system tests are explained in section 2. In section 3, moving-block signaling systems and the main concepts of ERTMS is explained. Calculation of train braking distance and design of an online adaptive PD controller to control the speed of trains are given in section 3. The thesis ends with a conclusion in section 4.

## **1.3 Contribution of the Thesis**

The contribution of this thesis can be summarized as follows:

Interlocking software development process and the use of railway related safety standards are explained in detail in this thesis. In particular representation of the railway field components in a formal way and the steps of the interlocking software development process are explained with a case study.

A new voting strategy related with diverse programming technique is proposed for safety-critical systems with binary decisions. Despite its many advantages, a disadvantage of the proposed voting strategy is the possibility of occurrence of some synchronization problems. A formal approach to determine such problems and possible solution methods has been proposed.

Consideration of the train movements in a single railway line as moving-blocks is dealt as a speed and acceleration control problem and the application of an online adaptive controller to solve this problem can be considered as another contribution of this thesis.



## **2. FIXED-BLOCK SIGNALING SYSTEMS**

### **2.1 History of the Fixed-Block Signaling Systems**

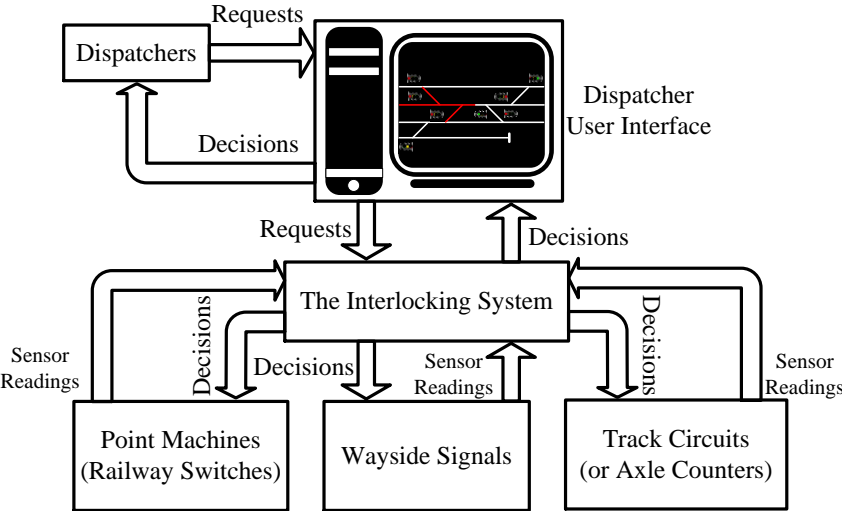
The main idea of the fixed-block signaling systems is to provide safe travel and transportation by keeping trains adequately apart from each other while moving on railways. The railway lines (tracks) are divided into fixed-length railway blocks where each railway block has an entrance and an end signal. These signals inform the train drivers about the occupation of the next railway block. Each railway block is permitted to be occupied by only one train at a time. Although there are some disadvantages such as the reduction in railway line capacity, fixed-block signaling systems are still in use since 1800's. In the early day when the railway traffic and density were not as much as today, railway systems did not need any signaling systems. Therefore, the train movements were managed by the help of the railway guards. The railway guards stand at the beginning of each railway block and warn the train drivers about the obstruction in front of their way [3]. By the development of railways in those years, many accidents occurred because of either human (railway guards or even drivers themselves) or nonhuman (component malfunctions) errors. In order to eliminate all these problems, the first interlocking system installation was built in UK 1843 [4]. The need for reliable and safe signaling system is much more today than in the past because of high demand and failures while satisfying such demands may result in fatal accidents [5]. In addition to mechanical interlocking systems where the railway traffic operations were realized by signalboxes manually [4], electronic interlocking systems such as SMILE [6], STERNOL [7], ELEKTRA [8] and other microprocessor-based systems [9] were also used. Many studies on modeling, design and verification of railway signaling and interlocking systems can be found in the literature [10-19]. Nowadays, as an alternative way, already certified COTS (Commercial off-the-shelf) solutions are also available in the market provided by private companies [20].

Even though the name of the signaling system varies from country to country, the basic principles remain almost the same. For instance, the basic principles of the

British Absolute Block Signaling (ABS) [21] and the basic principles the North American Centralized Traffic Control (CTC) [22] system are very similar to each other. The main components used in fixed-block signaling systems are explained in the next section.

**2.2 Components of the Fixed-Block Signaling Systems**

The general block diagram of a fixed-block signaling systems is illustrated in figure 2.1.



**Figure 2.1 :** General block diagram of the fixed-block signaling system.

**2.2.1 The traffic control center**

The Traffic Control Center (TCC) is directly responsible for all train traffic in its liability region. Additionally, the scheduling of trains, monitoring of the train movements and observing the conditions of the railway field components are under the responsibility of the TCC. Since the safe train movements in fixed-block signaling systems is mainly relies on route reservation procedures, the responsible officers or namely the dispatchers request routes for incoming and outgoing trains.

**2.2.2 The interlocking system**

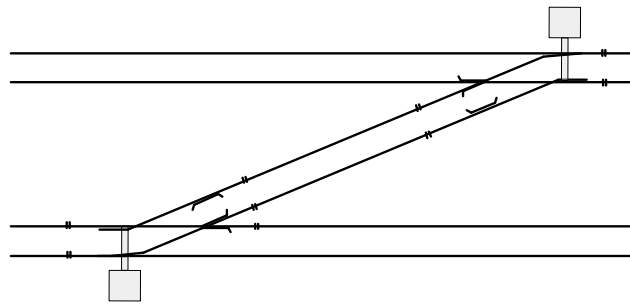
As given in figure 2.1, the interlocking system receives the dispatcher requests and compares these requests with the actual condition of the railway field including the situation of the railway field components and the railway traffic. After comparison, an incoming request is accepted if all safety criteria are satisfied or rejected. The



dispatcher cannot give commands to the railway field components directly. The main task of the interlocking system is to ensure safety at all times.

### 2.2.3 Point machines (railway switches)

The point machines (railway switches or turnouts) enable trains to pass from one railway track to another. A point machine usually has two positions namely normal and reverse. Position of the point machines can be changed from the TCC as long as the interlocking system allows such a move. Point machines also take proper positions after an incoming route request from the TCC is accepted by the interlocking system. An example schematic representation of a point machine is given in figure 2.2.







**Figure 2.2 :** Schematic representation of a railway switch.

### 2.2.4 Signals (optical wayside signals)

In fixed-block signaling systems, the wayside signals are established along the rail line on certain locations to inform the train drivers about the next railway block. The train drivers have to pay attention to the signals on the right side with respect to their direction of movement. Even though the use of colors on signals and the physical types of the signals differ from country to country, usually red color means the next railway block is occupied, yellow color indicates that the next railway block is free but not the railway block after that and green color means at least the next two railway blocks are free. Depending on the regional requirements, Turkish State Railways uses an additional aspect near station areas. The fourth aspect (a bottom yellow sign) designates a line change ahead (so the train should proceed with reduced speed on switch regions). Movement of two trains in a single railway line is illustrated in figure 2.3 and the definitions and the types of common signals used by the Turkish State Railways are given in table 2.1.

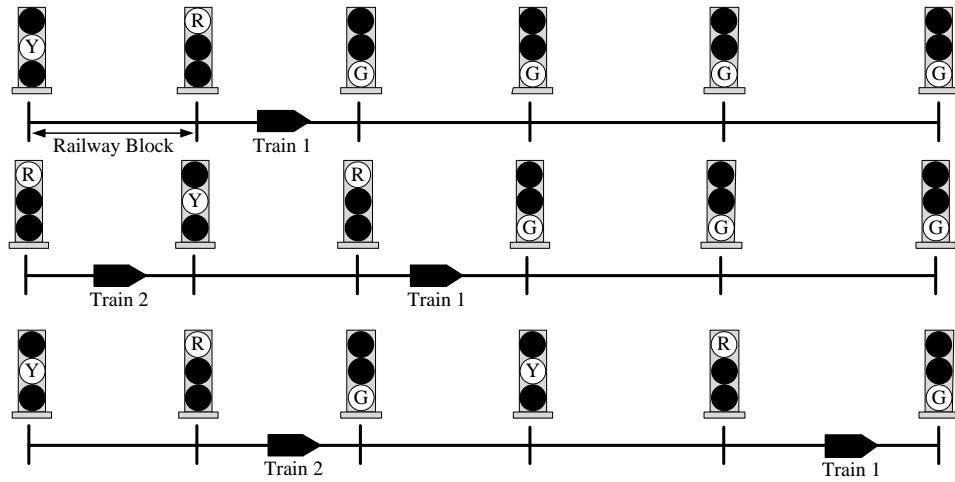
**Table 2.1 :** The types and the definitions of common signals.

Type of the Signal		Colors and Definitions	
	Four-aspect Tall Signal	Green (G)	: Next two blocks are free; the train can proceed.
		Yellow (Y)	: The next block is free, but the second block is occupied. Proceed carefully.
		Red (R)	: Stop, the next block is occupied.
		Yellow-Green (YG)	: There is a turning (a line change through a switch or switches) ahead and the next two blocks are free.
		Yellow-Yellow (YY)	: There is a turning ahead and the next block is free, but the second block is busy.
	Three-aspect Tall Signal	Yellow-Red (YR)	: Proceed very carefully (stop when necessary).
		Green	: Next two blocks are free; the train can proceed.
		Yellow	: The next block is free, but the second block is occupied. Proceed carefully.
	Three-aspect Dwarf Signal	Red	: Stop, the next block is occupied.
		Green	: Next two blocks are free; the train can proceed.
		Yellow	: The next block is free, but the second block is occupied. Proceed carefully.
	Two-aspect dwarf Signal	Red	: Stop, the next block is occupied.
		Yellow	: Proceed very carefully (stop when necessary).
		Yellow-Red	: The next block is free, but the second block is occupied. Proceed carefully.

*Note: The dwarf signals are located at the exits of secondary lines of the railway fields. They indicate that the train will be changing lines through a switch (or a set of switches).*

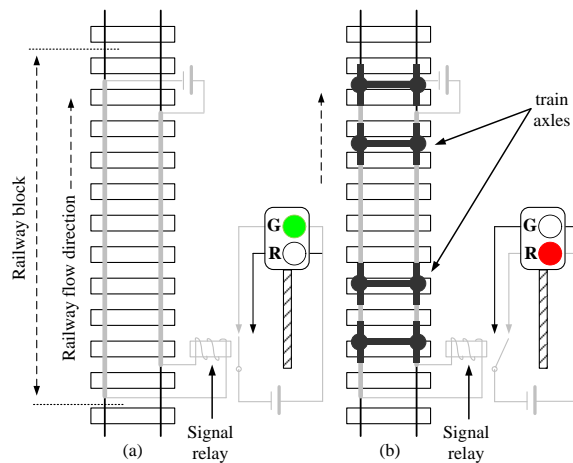
### 2.2.5 Track circuits and axle counters

The location of trains in fixed-block signaling systems are detected by the help of track circuits or axle counters. These equipment are connected in series depending on the length of the railway block because a railway block can contain more than one track circuit. When a train enters the railway block, the train axles short-circuit the track circuit (see figure 2.4). In this situation, the interlocking system assumes the related block as occupied by a train. Turkish State Railways use three different types of track circuits such as DC-type, AC-type and Jointless-type track circuits [4].



**Figure 2.3 :** Movement of trains in a single railway line (fixed-block).

Alternatively, axle counters can be used to detect the train locations. The counter heads of the axle counters are located at the intersection points of the railway blocks and counts the train axles. The railway block is assumed as occupied until the total number of the incoming and outgoing axles becomes equal.



**Figure 2.4 :** (a) Unoccupied track circuit, (b) Occupied track circuit.

### 2.3 Influence of Functional Safety Standards on Fixed-Block Signaling Systems

The European safety standard [23] defines failure as the termination of the capability of a functional unit to provide a required function in any way other than as desired. If the results of a failure are vital and may lead to death of many people, systems such as industrial plants, air traffic control systems or railway systems can be regarded as safety-critical systems [24], [25].

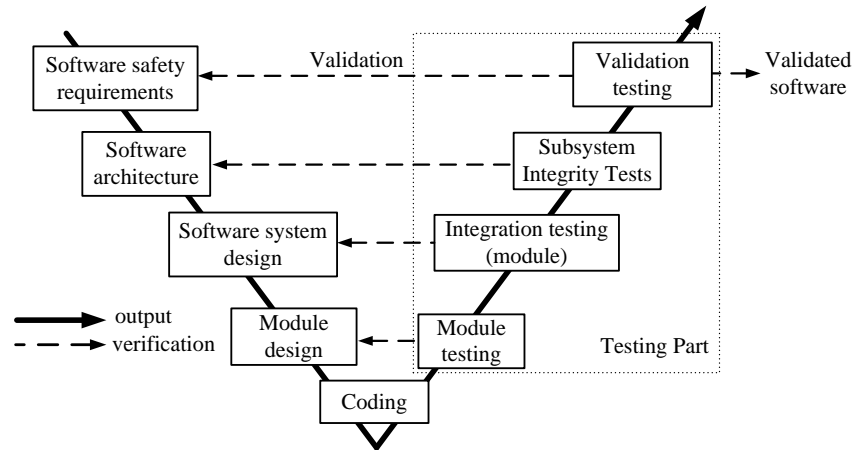
In this context, railway signaling systems are expected to satisfy the recommendations of safety related standards such as IEC 61508 where functional safety requirements of all kinds of Electrical/Electronic/Programmable Electronic (E/E/PE) devices. In addition to this umbrella standard, EN 50126 describes the functional safety requirements related with all kinds of railway applications where Reliability, Availability, Maintenance and Safety analysis (RAMS) are determined. EN 50128 (similar to EN 61508-3) determines methodologies for building software for railway control and applications and EN 50129 (similar to EN 61508-2) defines requirements for the hardware of E/E/PE devices [26].

These standards also bring out a widely known definition named Safety Integrity Level (SIL), which is a discrete level for specifying the safety integrity requirements of the safety functions to be allocated to the E/E/PE safety-related systems [23]. SIL definition is made in [27] for Software SIL as a classification number that determines the techniques those have to be applied to reduce software faults to an appropriate level and System SIL as a classification number that determines the required rate of confidence, separately. As an example, for a SIL 3 system in high demand mode of operation or continuous mode of operation [23], average frequency of a dangerous failure of the safety function per hour (failure rate -  $\lambda$ ) is between  $10^{-8}$  and  $10^{-7}$  [28]. The corresponding value of the mean time to failure (MTTF) is roughly between 1000 and 10000 years. In another words, a SIL 3 system is expected to work between 1000 to 10000 years without falling into the hazardous state. The relation between the expected range of failure per hour ( $\lambda$ ) and SIL can be seen from table 2.2 [29].

**Table 2.2 :** Relationship between SIL,  $\lambda$  and MTTF.

SIL	High Demand or Continuous Operation Mode (Range of $\lambda$ )	Low Demand Operation Mode (Range of MTTF)
4	$10^{-9} \leq \lambda < 10^{-8}$	$100000 \geq \text{MTTF} \geq 10000$
3	$10^{-8} \leq \lambda < 10^{-7}$	$10000 \geq \text{MTTF} \geq 1000$
2	$10^{-7} \leq \lambda < 10^{-6}$	$1000 \geq \text{MTTF} \geq 100$
1	$10^{-6} \leq \lambda < 10^{-5}$	$100 \geq \text{MTTF} \geq 10$

In order to satisfy the desired safety integrity level, several methods and techniques are recommended for the software development process. While developing software in accordance with the safety standards, the designers should follow the software development lifecycle (the V-model) given in figure 2.5 [2].



**Figure 2.5 :** The V-model.

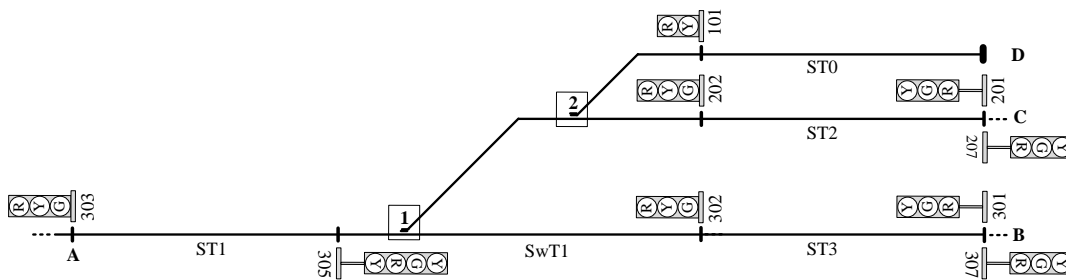
As an initial step, the safety requirements of the software and the required SIL of the software have to be determined. In addition to the international safety requirements such as the train must not pass the signal while it is red, additional safety requirements can be determined by the competent authorities which is the Turkish State Railways in Turkey. The required SIL for a given system can be determined by using the figure E.1, the figure E.2 and the table E.1 in EN 61508-5 [30]. In the second step, combination of several recommended software architectures by [27] in table A.3 should be chosen in order to provide the determined SIL in the first step. Later, related system sub models (the modules) have to be obtained. Suitable methods for system modeling according to the determined SIL can be found in table A.16 in [27].

After modeling, the required components of the safety-critical system, the obtained models have to be transformed into code snippets (or software function blocks). Later, the developed software have to be tested for verification, validation and commissioning. Due to the fact that software development process requires much more concentration and work, hardware solutions that have the appropriate SIL can be provided using certified commercial off-the-shelf (COTS) products from the market. A case study including the above definitions and the use of techniques and measures will be given in detail in the next section.

## 2.4 Case Study: Turkish National Railway Signaling Project

Some of the details and a brief explanation of the Turkish National Railway Signaling Project (TNRSP) are given in this section. The use of recommended methods and techniques by the railway related safety standards are also discussed. Please note that, the safety software developed within the TRNSP will be called as the interlocking software.

As mentioned in section 2.3, as an initial step the interlocking table of the railway field is constructed by taking into account the international safety requirements and the requirements mentioned by the Turkish State Railways. Interlocking table consist of all routes in the railway field including the entrance and exit signals, their color indication information, related point machine positions and other prohibitions [31], [32]. A sample railway field and its interlocking table are given in figure 2.6 and figure 2.7, respectively.



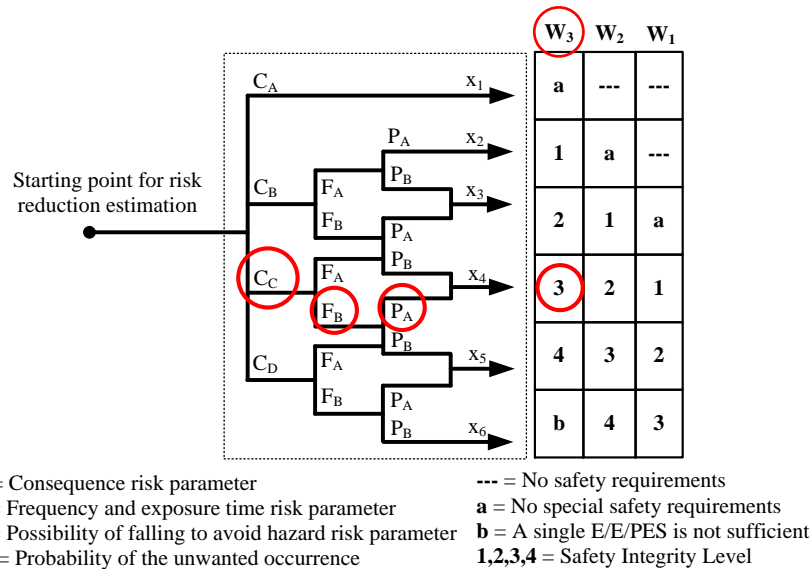
**Figure 2.6 :** A sample railway field.

The given railway field in figure 2.6 consist of 2 point machines (1 and 2), 4 three-aspect tall signals (201, 207, 301, 307), 3 three-aspect dwarf signals (202, 302, 303), 1 two-aspect dwarf signal (101), 1 four-aspect tall signal (305) and 5 railway blocks (ST0, ST1, ST2, ST3, SwT1). For instance, to reserve the route 1 (A-D), the railway block SwT1 must be unoccupied; the signals that are in the opposite direction such as 302 and 202 must be red, the point machines 1 and 2 must be in reverse position. After all these conditions are satisfied, the entrance signal of the route 1, signal 305, can be yellow-yellow (YY). If another train occupies the railway block ST0, the entrance signal 305 will be yellow-red (YR).

Name	Definition		Signal ID	Signal Color		Lock		Block Control	Route Lock	
	Route Number	Route		Point Machines	Incoming Signals					
Entrance Signal	Route 1 (A-D)	ST1-ST0	305	YY	-	1,2	101, 202, 302	SwT1, ST0	(SwT1)	
				YR	-			SwT1, ST0	(SwT1)	
	Route 2 (A-C)	ST1-ST2		YG	207 - Y, G	1,2		101, 202, 302, 201, 301	SwT1, ST2	(SwT1)
				YY	207 - R				SwT1, ST2	(SwT1)
				YR	-				SwT1, ST2	(SwT1)
	Route 3 (A-B)	ST1-ST3		101	G	307 - Y, G		1	101, 202, 302, 201, 301	SwT1, ST3
			Y		307 - R	SwT1, ST3	(SwT1)			
	Route 4 (D-A)	ST0-ST1	Y		303 - Y, G, R	1,2	202, 302, 305	SwT1, ST1		(SwT1)
			YR		-			SwT1, ST1		(SwT1)
	Route 5 (C-A)	ST2-ST1	202	G	303 - G	1,2	101, 302, 305	SwT1, ST1	(SwT1)	
				Y	303 - Y, R, YR			SwT1, ST1	(SwT1)	
				YR	-			SwT1, ST1	(SwT1)	
Route 6 (B-A)	ST3-ST1	302		G	303 - G	1		101, 202, 305	SwT1, ST1	(SwT1)
				Y	303 - Y, R, YR				SwT1, ST1	(SwT1)
				YR	-	SwT1, ST1			(SwT1)	

**Figure 2.7 :** The interlocking table of the railway field given in figure 2.6.

The required SIL is determined by using the risk graph given in [30]. The consequence risk parameter (C) can be chosen as  $C_C$  level for the interlocking systems since the system failures can result in death of several people. Similarly, the risk parameter (F) is chosen as frequent to permanent exposure in the hazardous zone ( $F_B$ ). Since avoiding from hazardous events are possible under certain conditions, possibility of falling to avoid hazard risk (P) is chosen as  $P_A$  and lastly, due to high probability of unwanted risk occurrence, probability of the unwanted occurrence of risk (W) is chosen as  $W_3$ . As a result of these observations, the required SIL should be **at least** 3 for the interlocking systems. The SIL determination process can be seen from figure 2.8.



**Figure 2.8 :** Determination of SIL.

Among several recommendations of [27], Defensive Programming, Failure Assertion Programming and Diverse Programming techniques to achieve a SIL 3 software in the TNRSP. The aim of Defensive Programming is to detect abnormal control flow, data flow or data values during their execution and react to these in a predetermined and acceptable manner [27]. In other words, designer has to put several extra control points to check the validity of results and/or variables of the program. For instance, if a railway block is occupied, the entrance signal of this railway block cannot be yellow or green. Similarly, movement of a point machine must not be allowed when its railway block is occupied. Such simple general rules can be checked independently, regardless of the current state of the program, just before sending output signals to the railway field.

In Failure Assertion Programming, the key point is to check the initial conditions for validity before execution of a command and to check all results after the execution of a command while executing safety-critical commands. For example, position of a point machine should be checked to ensure correct functioning of the point machine before and after a movement request. Similarly, a signal cannot be red and green at the same time. If red and green color indications are received from a signal at the same time, this must be considered as a failure by the developed software [27].

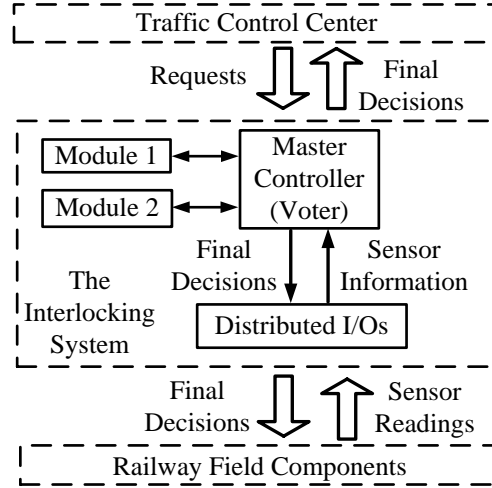
Finally, the aim of Diverse Programming (or namely, N-version Programming) is to detect and mask software design faults during execution of a program in order to prevent safety-critical failures of the system and continue operation with high reliability. The main concept of N-version programming is to develop N-different algorithms that have the same input-output specifications built up by N-different (ideally non-interacting) workgroups. In other words, while execution of a program, N-version programming detects residual software faults to prevent system from getting into fatal failures and provides high reliability for execution continuity [27]. The outputs of these N-different software versions are then sent to another unit, which is usually known as the voter. The voter can be in hardware, software, or both, depending on the application requirements [33]. The voter receives outputs of N-different modules and compares them depending on the voting strategy. Several voting algorithms (or strategies) and their comparison can be found in the literature [34-41]. If the safe-state of the system is not determined (or if the system is not



safety-critical) generalized voting strategies [34], [35] can be used where generally  $N$  is chosen as 3 (Triple modular redundancy - TMR) [42-44]. In such topologies, the voter usually applies the majority (or a weighted average) of the decisions given by the modules. Nevertheless, for fail-safe systems, where safety is the most important concern, the complete agreement of the modules can be sought before getting into an “unsafe” state. Hence, usually  $N$  is chosen as 2 [27]. The system gets into safe-state and produces predetermined fail-safe output, when there is a disagreement between the modules. The use of such a strategy is usually encountered in high level safety systems. However, the main disadvantage of this strategy is a considerable reduction on system availability, since disagreements between modules, which result in system locks, can happen frequently.

While comparing the incoming responses from the modules two possible strategies exist: either all decisions are compared as a whole, or every decision bit can be compared separately. If the decisions are compared as a whole the system gets into the safe-state (all bits are set to their safe values) when there is an incompatibility between the decisions. In contrast, only the bits for which incompatibilities exist are set to their safe values in bitwise comparison of the decisions. In this sense, bitwise voting strategy based on safe-states of variables can provide higher safety in comparison to the bitwise TMR architecture, while providing higher availability rates in comparison to the strategies that take the decisions as a whole.

For instance, in a majority voting strategy with three modules, the voter accepts the decision whenever two modules accept it even if these modules give wrong decisions. A more realistic example can be given for the interlocking systems. While a train is moving on a railway block containing a point machine, the point machine must be kept in its position. In the TMR architecture with a majority voting strategy, the point machine position can be changed if two modules accept it, which may cause the derailment of the train. However, in the proposed bitwise voting strategy, if one of the modules denies the position change, the position of the point machine will not change due to the safe-state of the position variable of the point machine. More information about the proposed bitwise voting strategy will be given in section 2.5. The inner structure of the interlocking system given in figure 2.1 can be seen from figure 2.9.



**Figure 2.9 :** The general framework of the railway signaling system.

As mentioned before, the main aim of N-version programming is to develop N-different algorithms that have the same input-output specifications. Since the fixed-block signaling systems can be considered as Discrete Event Systems (DES) due to having features like non-determinism, asynchronism, event-driven and simultaneity in their structures and to achieve the diversity in software design, two highly recommended modeling methods for SIL 3 software given in table A.16 in [27] are used in TNRSP. Brief definitions about the modeling methods to construct the software blocks of the modules given in figure 2.9 and the models of the railway field components will be given in subsection 2.4.2.1.

### 2.4.1 Automata as a modeling tool

A deterministic automaton (or a deterministic finite state automaton), denoted by  $G$ , is a six tuple [45]:

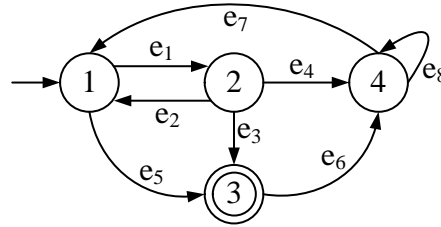
$$G = (Q, \Sigma, f, \Gamma, q_0, Q_m) \quad (2.1)$$

where

- $Q$  : is the set of states,
- $\Sigma$  : is the finite set of events,
- $f$  :  $Q \times \Sigma \rightarrow Q$  is the partial transition function on its domain,
- $\Gamma$  :  $Q \rightarrow 2^\Sigma$  is the active event function,
- $\Gamma(q)$  : is the set defined for every state of  $G$  and represents the feasible events of  $q$ ,

- $q_0$  : is the initial state,
- $Q_m$  : is the set of marked states representing a completion of a given task or operation.

The given automaton by (2.1) is assumed as deterministic because  $f$  is a function of  $Q \times \Sigma$  to  $Q$ . In other words, there cannot be two transitions with the same event label out of a state. For the nondeterministic automaton the transition function  $f$  is a function of  $Q \times \Sigma$  to  $2^Q$ . In such a case, there can be multiple transitions with the same event label [45]. In order to apply the method based on the Automaton model, first the events have to be identified and a state transition graph have to be obtained [46]. An example automaton is illustrated in figure 2.10.



**Figure 2.10** : An example automaton.

For the automaton model given in figure 2.10,  $Q = \{1, 2, 3, 4\}$ ,  $q_0 = 1$ ,  $Q_m = 3$ ,  $\Sigma = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$ ,  $f(1, e_1) = 2$ ,  $f(1, e_5) = 3$ ,  $f(2, e_2) = 1$ ,  $f(2, e_3) = 3$ ,  $f(2, e_4) = 4$ ,  $f(3, e_6) = 4$ ,  $f(4, e_7) = 1$ ,  $f(4, e_8) = 4$ . For more definitions about the specifications of the automata modeling method and the use of automata, the reader is referred to [45] and [47].

### 2.4.2 Petri nets as a modeling tool

A marked Petri net, denoted by  $PN$ , is a five tuple [45]:

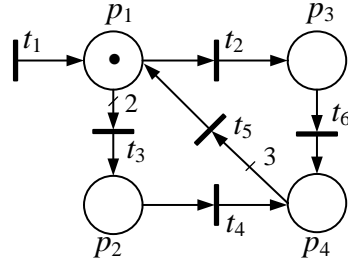
$$PN = (P, T, Pre, Post, W, M_0) \quad (2.2)$$

where

- $P$  :  $\{p_1, p_2, \dots, p_n\}$ , is the finite set of places,
- $T$  :  $\{t_1, t_2, \dots, t_k\}$ , is the finite set of transitions,
- Pre :  $P \times T \rightarrow \mathbb{N}$  directed ordinary arcs from places to transitions,
- Post :  $T \times P \rightarrow \mathbb{N}$  directed ordinary arcs from transitions to places,

- $W : \text{Pre, Post} \rightarrow \{1, 2, 3, \dots\}$  weight function on the arcs,
- $M_0 : P \rightarrow \mathbb{N}$  initial marking,
- $P \cap T = \emptyset$  and  $P \cup T \neq \emptyset$ .
- $\mathbb{N}$  is a set of nonnegative integers.

An example  $PN$  is illustrated in figure 2.11.



**Figure 2.11** : An example  $PN$ .

For the  $PN$  model given in figure 2.11,  $P = \{p_1, p_2, p_3, p_4\}$ ,  $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ ,  
 $M_0 = \{1, 0, 0, 0\}$ ,  $\text{Pre} = \{(p_1, t_2), (p_1, t_3), (p_2, t_4), (p_3, t_6), (p_4, t_5)\}$ ,  
 $\text{Post} = \{(t_1, p_1), (t_2, p_3), (t_3, p_2), (t_4, p_4), (t_5, p_1), (t_6, p_4)\}$ ,  $W(p_1, t_2) = 1$ ,  
 $W(p_1, t_3) = 2$ ,  $W(p_2, t_4) = 1$ ,  $W(p_3, t_6) = 1$ ,  $W(p_4, t_5) = 3$ ,  $W(t_1, p_1) = 1$ ,  
 $W(t_2, p_3) = 1$ ,  $W(t_3, p_2) = 1$ ,  $W(t_4, p_4) = 1$ ,  $W(t_5, p_1) = 1$ ,  $W(t_6, p_4) = 1$ .

For a marking  $M : P \rightarrow \{1, 2, \dots\}$ ,  $M(p_n) = k$  means that the  $n$ th place has  $k$  tokens [48]. A marking  $M$  can also be represented by a vector with  $l$  elements where  $l$  is the total number of places. Some basic properties of  $PN$ s are as follows [45], [48]:

**1.** A transition  $t_k$  is said to be enabled at a marking  $M$  if each input place  $p_n$  of  $t_k$  has at least  $W(p_n, t_k)$  tokens, where  $W(p_n, t_k)$  is the weight of the arc from place  $p_n$  to transition  $t_k$ , that is,  $M(p_n) \geq W(p_n, t_k)$  for all  $p_n \in I(t_k)$ . Note that,  $I(t_k)$  and  $O(t_k)$  represents the sets of input places and output places of transition  $t_k$ , respectively; and if  $I(t_k) = \emptyset$ , transition  $t_k$  is always enabled.

$$\begin{aligned}
 I(t_k) &= \{p_n \in P : (p_n, t_k) \in \text{Pre} \cup \text{Post}\} \\
 O(t_k) &= \{p_n \in P : (t_k, p_n) \in \text{Pre} \cup \text{Post}\}
 \end{aligned}
 \tag{2.3}$$

2. An enabled transition may or may not fire (depending on whether or not the event actually takes place).

3. The firing of an enabled transition  $t_k$  removes  $W(p_n, t_k)$  tokens from each  $p_n \in I(t_k)$  and adds  $W(t_k, p_n)$  tokens to each  $p_n \in O(t_k)$ , where  $W(t_k, p_n)$  is the weight of the arc from  $t_k$  to  $p_n$ . That is,

$$M'(p_n) = M(p_n) - W(p_n, t_k) + W(t_k, p_n) \quad (2.4)$$

where  $M'(p_n)$  is the number of tokens in the  $n$ th place after the firing of transition  $t_k$ .

4. A marking  $M_k$  is reachable from the initial marking  $M_0$  in a  $PN$  if there exists a sequence of transitions  $t_1 t_2 \dots t_m$  such that  $M_0[t_1 > M_1[t_2 > \dots M_{k-1}[t_k > M_k$  and  $R(M_0)$  denotes the set of all reachable markings from  $M_0$ .

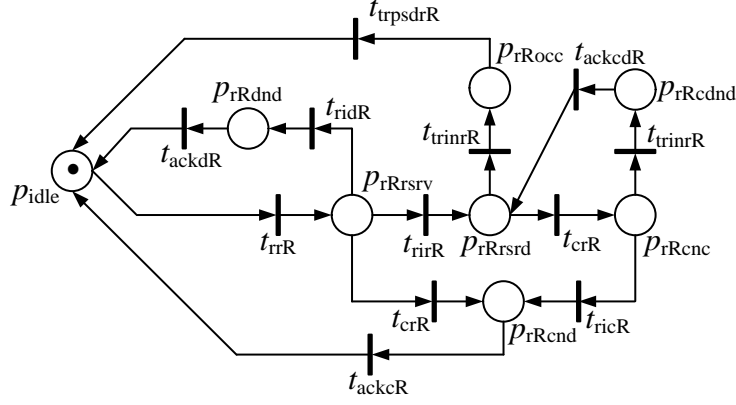
5. A Petri net  $PN$  is said to be  $m$ -bounded if the number of tokens in each place does not exceed a finite number  $m$ , that is,  $\forall M_k \in R(M_0), \forall p_i \in P: M_k(p_i) \leq m$ . Additionally,  $PN$  is *safe* if it is 1-bounded.

6. A Petri net  $PN$  is said to be *deadlock-free* if at least one transition is enabled at every reachable marking  $M_k \in R(M_0)$ .

#### 2.4.2.1 Petri net models of the railway field components

In this subsection, the  $PN$  models of the railway field components are explained by the help of the railway field given in figure 2.6. It is important to note that,  $PN$  models given in this section are compact models that just represent the basic operational behavior of the railway field components. These  $PN$  models can be converted to software codes (software blocks) by using several approaches [49-51].

Route reservation model of a single route is given in figure 2.12 with the meanings of the places and transitions in table 2.3, respectively.



**Figure 2.12** : Route reservation model of a single route.

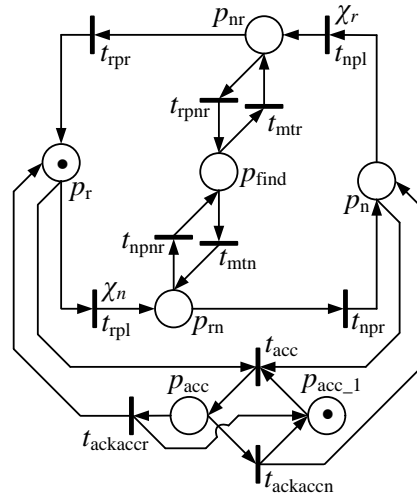
**Table 2.3** : Definitions of places and transitions in figure 2.12.

Places	Meaning	Transitions	Meaning
$p_{idle}$	Routes are idle	$t_{rrR}$	Reserve route
$p_{rRdnd}$	Route is denied	$t_{rirR}$	Route is reserved
$p_{rRrsrv}$	Route is reserving	$t_{crR}$	Cancel route
$p_{rRrsrd}$	Route is reserved	$t_{ricR}$	Route is canceled
$p_{rRcnc}$	Route is canceling	$t_{ridR}$	Route is denied
$p_{rRcnd}$	Route is canceled	$t_{trinrR}$	Train in route
$p_{rRocc}$	Route is occupied	$t_{trpsdrR}$	Train has passed route
$p_{rRcdnd}$	Route cancellation is denied	$t_{ackcR}$	Route cancellation acknowledged
		$t_{ackdR}$	Route denial acknowledged
		$t_{ackcdR}$	Route cancellation denial acknowledged

When the dispatchers ( $t_{rrR}$ ) make a route reservation request, the interlocking system evaluates the request with respect to the actual situation of the railway field components ( $p_{rRrsrv}$ ). If a cancellation request is received before the route is reserved ( $t_{crR}$ ), the route will be cancelled immediately ( $p_{rRcnd}$ ). After the comparison, if all safety criteria are met ( $t_{rirR}$ ) then the route will be reserved otherwise the route will be denied ( $p_{rRdnd}$ ). If a cancellation request is received after the route is reserved ( $t_{crR}$ ), the route cancellation process will be evaluated ( $p_{rRcnc}$ ). Finally, after the route is reserved, the route will be released when a train has passed ( $t_{trpsdrR}$ ). If a train has entered ( $t_{trinrR}$ ) the route while it is canceling then the route cancellation process will be denied ( $p_{rRcdnd}$ ) and the route will be kept as reserved ( $p_{rRrsrd}$ ).

Next, a single point machine (PM) model is given in figure 2.13 with the meanings of the places and transitions in table 2.4, respectively. A PM can be controlled either

manually or by a route request by the dispatchers. After a movement request has received ( $\chi_n$  or  $\chi_r$ ) if all safety criteria are satisfied, the PM starts to move from one position to another ( $p_m$  or  $p_{nr}$ ). As an operational requirement of Turkish state Railways, a PM is expected to reach the desired position in 7 seconds. If the PM did not reach to desired position ( $t_{rpnr}$  or  $t_{npnr}$ ), this is regarded as an indication fault by the interlocking system ( $p_{find}$ ).



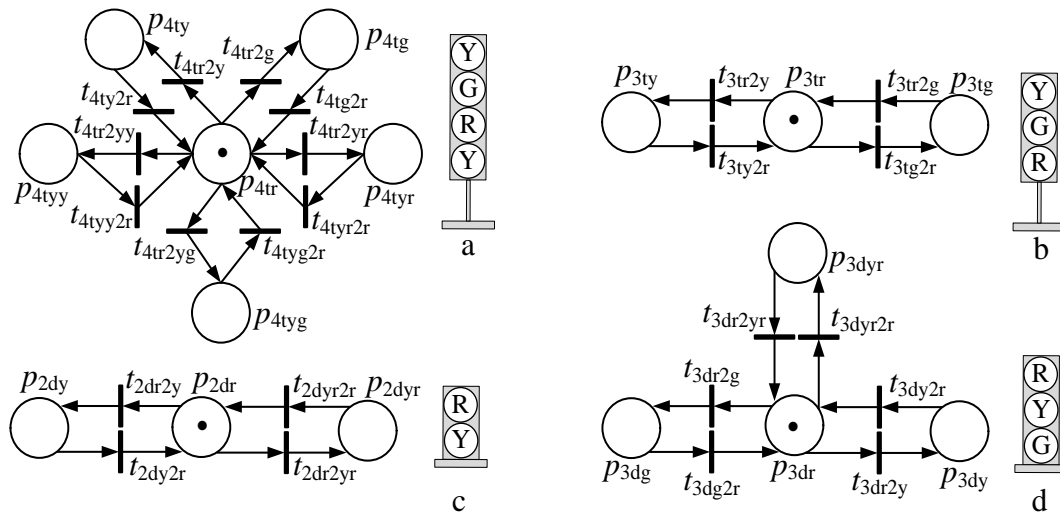
**Figure 2.13** : A single PM model.

**Table 2.4** : Definitions of places and transitions in figure 2.13.

Places	Meaning	Transitions	Meaning
$p_r$	PM is in reverse position	$t_{npl}$	PM left normal position
$p_n$	PM is in normal position	$t_{npr}$	PM reached to normal position
$p_mn$	PM is moving from reverse to normal position	$t_{ackaccn}$	Data congruence fault acknowledged to normal position
$p_nr$	PM is moving from normal to reverse position	$t_{ackaccr}$	Data congruence fault acknowledged to reverse position
$p_{find}$	Indication fault	$t_{mtn}$	Move PM to normal position
$p_{acc}$	Data congruence fault	$t_{mtr}$	Move PM to reverse position
$p_{acc\_1}$	Control place for data congruence fault	$t_{rpnr}$	PM did not reach to reverse position
$\chi_n$	Movement request to normal position	$t_{npnr}$	PM did not reach to normal position
$\chi_r$	Movement request to reverse position	$t_{acc}$	Occurrence of data congruence fault
		$t_{rpr}$	PM reached to reverse position
		$t_{rpl}$	PM left reverse position

This fault can be overcome by another position request ( $t_{mtr}$  or  $t_{mntn}$ ). Additionally, another malfunction occurs when the PM sensors send the position signal at the same time ( $p_{acc}$ ). In this case, the PM sensors should be checked and an acknowledgment signal should be sent to the interlocking system after the fault is fixed ( $t_{ackaccn}$  or  $t_{ackaccr}$ ).

Moreover, the PN models of the signals described in table 2.1 are given in figure 2.14 with the meanings of the places and transitions in table 2.5, respectively (G-Green, Y-Yellow, R-Red, YY-Yellow-Yellow, YR-Yellow-Red, YG-Yellow-Green). The interlocking system sends color information to the related signals according to the interlocking table after the PM(s) took proper positions. All signals in the railway field should be red when there is no reserved route. The interlocking system should also check the signals in the railway field for possible signal faults. These faults can be classified as no color indication fault or multiple color indications fault. In the first case, the signal do not show any color information whereas in the latter case, the signal shows more than one color information (etc. red and green). When any of these faults occur, the interlocking system denies the route request or cancels the reserved routes that are related with the faulty signal. These signal faults are not mentioned in the signal models to decrease the complexity in the figures.



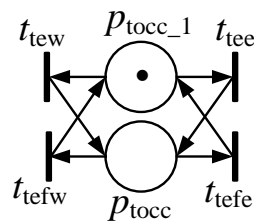
**Figure 2.14 :** Signal models (Four-aspect tall signal (a), Three-aspect tall signal (b), Two-aspect dwarf signal (c), Three-aspect dwarf signal (d)).



**Table 2.5 :** Definitions of places and transitions in figure 2.14.

Places	Meaning	Transitions	Meaning
$p_{4tr}$	Signal is red	$t_{4tr2y}$	Red to yellow
$p_{4ty}$	Signal is yellow	$t_{4ty2r}$	Yellow to red
$p_{4tg}$	Signal is green	$t_{4tr2g}$	Red to green
$p_{4tyy}$	Signal is YY	$t_{4tg2r}$	Green to red
$p_{4tyg}$	Signal is YG	$t_{4tr2yr}$	Red to YR
$p_{4tyr}$	Signal is YR	$t_{4tyr2r}$	YR to red
$p_{3tr}$	Signal is red	$t_{4tyg2r}$	YG to red
$p_{3tg}$	Signal is green	$t_{4tr2yg}$	Red to YG
$p_{3ty}$	Signal is yellow	$t_{4tr2yy}$	Red to YY
$p_{2dr}$	Signal is red	$t_{4tyy2r}$	YY to red
$p_{2dy}$	Signal is yellow	$t_{3tr2g}$	Red to green
$p_{2dyr}$	Signal is YR	$t_{3tg2r}$	Green to red
$p_{3dr}$	Signal is red	$t_{3tr2y}$	Red to yellow
$p_{3dy}$	Signal is yellow	$t_{3ty2r}$	Yellow to red
$p_{3dg}$	Signal is green	$t_{2dr2y}$	Red to yellow
$p_{3dyr}$	Signal is YR	$t_{2dy2r}$	Yellow to red
Transition	Meaning	$t_{2dr2yr}$	Red to YR
		$t_{2dyr2r}$	YR to red
		$t_{3dr2g}$	Red to green
		$t_{3dg2r}$	Green to red
		$t_{3dr2y}$	Red to yellow
		$t_{3dy2r}$	Yellow to red
		$t_{3dr2yr}$	Red to YR
		$t_{3dyr2r}$	YR to red

Finally, the *PN* models of the track circuits are given in figure 2.15 with the meanings of the places and transitions in table 2.6, respectively.



**Figure 2.15 :** Railway block model.

**Table 2.6 :** Definitions of places and transitions in figure 2.15.

Places	Meaning	Transitions	Meaning
$p_{tocc}$	Railway block is occupied	$t_{tew}$	Train enters from west
$p_{tocc\_1}$	Railway block restriction	$t_{tee}$	Train enters from east
		$t_{tefw}$	Train exits from west
		$t_{tefe}$	Train exits from east

Since only one train is allowed to occupy a railway block at the same time, each place ( $p_{\text{tocc}}$ ) is restricted by another place ( $p_{\text{tocc}_1}$ ). Trains can enter from west ( $t_{\text{tew}}$ ) or east ( $t_{\text{tee}}$ ) depending on their direction of movement. When a train enters to the railway block, the token in place  $p_{\text{tocc}_1}$  moves to place  $p_{\text{tocc}}$  by the firing of related transitions. It should be noted once again that the *PN* models given in this section only describes the general operation logic of the railway field components. A case study about signaling system design for an example railway field will be given in subsection 2.5.3 .

## 2.5 Voting Strategy

In this section, the decision-making strategy of the developed interlocking system is described. In the TNRSP, the proposed bitwise voting strategy is used where the safe-state values of the variables are taken into account but some synchronization problems have been encountered. These problems and their solutions are also described in detail in this section.

The safe-state of a variable can be considered as a state that prevents the whole system not to fall into a dangerous situation due to that variable. The proposed voting strategy (hence the proposed voter) demands the complete agreement of all modules to set a variable into its “unsafe” state. The modules given in figure 2.9 are chosen as fail-safe programmable logic controllers. It is obvious from figure 2.9 that the voter also works as a communication unit between the field, the control center and the modules (the modules do not interact with each other and the control center directly). A fail-safe communication protocol such as safe-ethernet must be used for communication between all components (except for the communication between the voter and the control center) and the field. When a request is received from the control center (e.g. reserve route x or move switch 1 to its normal position) the voter sends this request to the modules concurrently and waits for their reply. After the voter receives the responses of the modules, it compares the responses according to a table where the safe-state of each variable is defined. The safe-state of each variable is determined at the “Software Safety Requirements Specification” level in the V-model (see figure 2.5) as an initial step before developing safety-critical software. N-different workgroups then develop their software using these specifications. Typically, voters are designed as simple as possible to reduce the probability of

possible faults in the voter. In this study, some simple but effective logic is also added to the voter as explained below.

The voter is designed as a state-independent machine. In other words, the decisions of the designed voter are independent from its previous decisions. This allows testing the voter for all possible inputs. In contrast, the modules are designed as state-dependent machines, as they have to make decisions depending on the previous inputs and states. In general, there can be infinite number of possible evolutions of the states in such machines. Therefore, it is usually not possible to test the modules for all possible input combinations. Frequently, the modules do not send their decisions at the same time due to their cycle times and communication delays. In such cases, after the first decision is received from a module, the voter waits along for a specified time called as the consistency time for the other module to give its decision. If the other module does not give any decision (no output to the voter) then the voter produces an error called consistency error. The voter keeps the corresponding output in its safe-state when there is an inconsistency. Moreover, the module that tries to drive a variable into its unsafe state, cancels its decision upon receiving the consistency error for that variable.

The consistency time is usually determined depending on the cycle times of each module and expected communication delays. It is a nice practice to choose the consistency time greater than the sum of 2-fold of the communication time, the cycle time of the voter and the module with longer cycle time. Assume that the cycle times of the modules and the voter are 220ms, 350ms and 100ms, respectively. The communication between the voter and the modules are assumed as 300ms. For this example, the consistency time should better be selected greater than  $(600\text{ms} + 100\text{ms} + 350\text{ms}) = 1050\text{ms}$ .

Similarly, when a module sends its decision after an incoming request from the voter, the module waits for a specified time called as the synchronization time for the answer of the voter about its decision. If the module does not receive any answer back within this time the module rejects the request, and so the voter. It is a nice practice to choose the synchronization time greater than the 2-fold of the sum of the communication time, the cycle time of the voter and the module with longer cycle time, and the consistency time. For the given cycle time values in the previous

paragraph, the synchronization time should better be selected greater than  $((300\text{ms} + 100\text{ms} + 350\text{ms} + 1050\text{ms}) \times 2 =) 3600\text{ms}$ . Actually, the determination of the synchronization time also depends on the time constraints of the process under consideration.

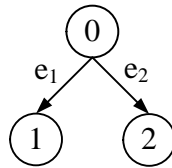
The voter demands complete agreement of all modules before getting into an unsafe state [46], [52]. In order to move from an unsafe state to a safe-state, however, decision of just one module towards this direction is enough. For instance, reserving a route for an incoming train needs more attention and the interlocking system has to check several conditions, therefore, reserving a route demands complete agreement of all modules. Similarly, keeping a signal color red is safer than the other color information, thus, the signal color will always be red unless all modules produce the same color other than red.

If an incoming request is accepted or rejected then the voter sends proper signals to the field and informs the control center and the modules about the result. The voter also records discrepancies between the modules. The process between the voter and the modules can be considered as a kind of handshaking process. This process of handshaking brings up the issue of synchronization between the modules, which is one of the main problems related with the N-version programming. Normally, the voter sends the requests synchronously to the modules. However, the modules receive the requests at different times depending on their cycle times and communication delays. This may lead a module not to produce an output in time or even fall into an irrelevant state, if the design has not been done carefully. In other words, synchronization problems between the two modules can occur. To prevent from this kind of situations, the voter sends proper signals after it receives a response from a module. This allows re-synchronizing the other module, which has not produced the expected response in time. There are also some other solutions to such problematic cases.

One possible solution is to wait both modules to recover themselves (each module waits until the end of synchronization time for the other module). Another possible solution is to operate the modules in an asynchronous manner by using safety precautions.

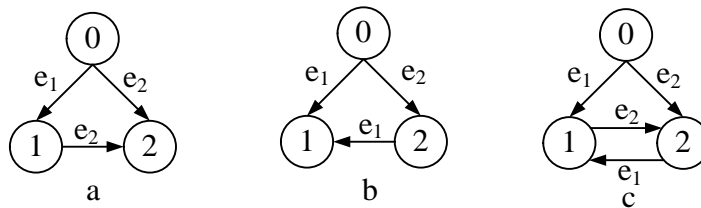
### 2.5.1 Synchronization problems

A problematic situation can arise if the modules can move into different states depending on the order of two incoming events. Consider the case as illustrated in figure 2.16, where the state of each module changes from state 0 to state 1 by the event  $e_1$  and to state 2 by the event  $e_2$ . If there is a possibility for the modules to receive the events in different order, then the modules can move into separate states. This kind of problem will be called as *the type 1 problem*.



**Figure 2.16 :** The type 1 problem.

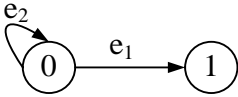
A possible solution to this type of problems is letting each module to change its state as shown in figure 2.17a or figure 2.17b. This will resolve the problem since the order of occurrence of the events  $e_1$  and  $e_2$  do not change the resulting state of the modules. It should be noted, however, that in this situation, if the event  $e_2$  changes the state of the modules from state 1 to state 2 and the event  $e_1$  changes the state of the modules from state 2 to state 1 (see figure 2.17c), this may also cause the modules to move into different states depending on the events they receive.



**Figure 2.17 :** Situation where the type 1 problem do not cause any failure (a, b), another situation where the type 1 problem causes a failure (c).

Sometimes the appearance and then the disappearance of a signal occur in a very short period of time. This situation may lead one of the modules to change its state, while the other module keeping its state. As is illustrated in figure 2.18, the occurrence of the event  $e_1$  for a very short period of time leads one of the modules to state 1, while the other module waits in state 0. In such a case, the modules divaricate at the end of the synchronization time. This case especially arises when the event  $e_1$

depends on internal states of the module (such as time-out situations). This kind of problem will be called as *the type 2 problem*.

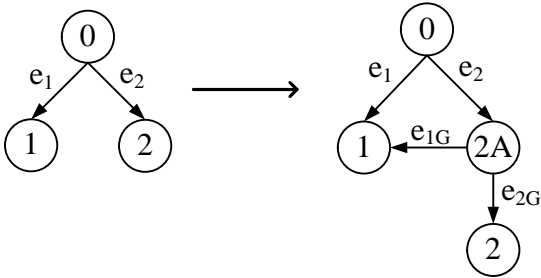


**Figure 2.18 :** The type 2 problem.

**2.5.2 Solutions for the synchronization problems**

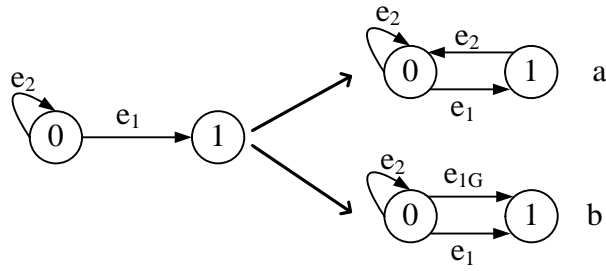
A possible solution for type 1 problem is to convert the state diagram given in figure 2.16 to figure 2.17a or figure 2.17b. Nevertheless, it is not always possible to convert a model to another one. An alternative way is to add extra states to synchronize the modules as given in figure 2.19.

Before passing to state 2 from state 0, an extra state is added and named as 2A. In this solution, the modules have to pass to state 2A before passing to state 2; and they can only pass to state 2 by the event  $e_{2G}$ , which happens when all modules are in state 2A. If a module passes to state 2A and receives a signal from the voter indicating that the other module passed to state 1 (denoted by the event  $e_{1G}$ ), it also passes to state 1 from state 2A.



**Figure 2.19 :** A possible solution for the type 1 problem.

Additionally, there are two possible solutions for the type 2 problem. The first solution proposed for this kind of problem is to take back the module that passes to state 1 into state 0 again by the event  $e_2$ . The second proposed solution is to take a module into state 1 from state 0 after receiving a signal from the voter indicating the passing of the other module into state 1 (denoted by the event  $e_{1G}$ ). These solutions are illustrated in figure 2.20a and figure 2.20b, respectively.



**Figure 2.20 :** Possible solutions for the type 2 problem.

A case study about signaling system design for an example railway field including these problems and the solutions will be given in the next section.

### 2.5.3 Signaling software (interlocking) design for an example railway field

In this section, signaling software design for the given railway field in figure 2.6 is explained. The *PN* models given in section 2.4.2.1 are used as software main blocks and possible synchronization problems given in section 2.5.1 are also discussed in detail. By the help of the interlocking table in figure 2.7 and the *PN* models of the railway field components, software blocks can be constructed easily. Since there are two PM in the railway field, the interlocking software must include two PM *PN* models. The other railway field component *PN* models can be added in a similar manner. As a result, the interlocking software should include six *PN* models for the route reservations (which are intersecting with each other), two *PN* models for the PMs, nine *PN* models for the signals and five *PN* models for the railway blocks. Later, the *PN* models or namely the software blocks must connect with each other according to the interlocking table. For instance, while reserving the route 5 (C-A), after the PM positions are adjusted (1-reverse position, 2-normal position) the signal 202 should be yellow when the signal 303 is yellow, red or yellow-red. In other words, the output of the *PN* model of the route 5 should be connected to *PN* models of the point machines 1 and 2, the related signals such as 202 and 303 and the track circuits ST1, SwT1 and ST2. The general representation of the interlocking software with respect to *PN* models is given in figure 2.21. The connections of the *PN* models are not given here due to reduce the complexity. As it is obvious from the figure 2.21 that, as an initial condition, all routes are idle (pidle), both PM s are in reverse position, al signals are red and all railway blocks are unoccupied.





By an incoming route request (e.g. route A-B) from the dispatcher, the token in ( $p_{idle}$ ) moves to the place ( $p_{rABrsrv}$ ) by the firing of the transition ( $t_{rrAB}$ ). Later, the token in the place ( $p_{r1}$ ) of the point machine 1 PN model, moves to the place ( $p_{rn1}$ ) by the firing of the transition ( $t_{rp11}$ ) after all firing conditions ( $\chi_{n1}$ ) are satisfied. When the PM 1 reaches to the desired position in a predetermined time interval (e.g. 7 seconds), the token in the place ( $p_{rn1}$ ) will pass to the place ( $p_{n1}$ ) over the transition ( $t_{npr1}$ ). Otherwise, the token in the place ( $p_{rn1}$ ) will pass to the place ( $p_{find1}$ ) over the transition ( $t_{pnr1}$ ) where the interlocking system denies the route request A-B due to the indication fault of the PM.

After the PM reaches to the desired position ( $p_{n1}$ ), the entrance signal of the route will inform the train driver by showing the proper color information according to the interlocking table. The entrance signal of the route A-B (signal 305) will be green if the next signal in the route (signal 307) is yellow or green; or the entrance signal of the route A-B will be yellow if the next signal in the route is red. After the entrance signal of the route A-B shows proper color indication, the token in the place ( $p_{rABrsrv}$ ) moves to the place ( $p_{rABrsrd}$ ) by the firing of the transition ( $t_{rirAB}$ ). This also means that the route A-B is reserved and the railway field components related with this route are locked electronically.

The token in the place ( $p_{rABrsrd}$ ) will also move to the place ( $p_{rABocc}$ ) by the firing of the transition ( $t_{trinrAB}$ ) after the entrance of the train in the first track circuit (SwT1) of the route A-B. After the train exits from the track circuit SwT1, the token in the place ( $p_{tocc\_SwT1}$ ) will pass to the place ( $p_{tocc\_1SwT1}$ ) by the firing of the transition ( $t_{tefeSwT1}$ ). In other words, the train reaches to the station 3 (ST3) and the route reservation A-B will be terminated by the passing of the token in the place ( $p_{rABocc}$ ) to the place ( $p_{idle}$ ) over the transition ( $t_{trpsdrAB}$ ).

Briefly, in addition to the explanations given above, different requests can be also made by the dispatchers. For instance, if a cancellation request is made before a train enters in the reserved route, the interlocking software send red color information to the entrance signal of the route and then cancels the reserved route immediately. If the cancellation request is made after the train enters in the reserved route, the interlocking software should deny the cancellation request. The interlocking software

is responsible of taking decisions by comparing the requests of the dispatchers and the actual situation of the railway field.

As given in figure 2.9, the requests of the dispatcher are first sent to the voter and then to the modules for evaluation. The voter and the modules are chosen fail-safe PLCs. After evaluation, each module sends its decision back to the voter. The voter gives the final decision by considering both module decisions and the safe-state of the request. After that, the voter sends its final decision both to the control center and to the modules. In case of the acceptance of the request, the voter also sends its final decision to the related railway field components. The data sequence diagram for the reservation of the route A-B is given in figure 2.22.

The final decision is always made by the voter and the voter sends this decision to the modules, the control center and the field, if necessary. As mentioned before, the voter also takes into account the safe-state of each variable while making its decisions. The safe-states of the variables of concern are given in table 2.7.

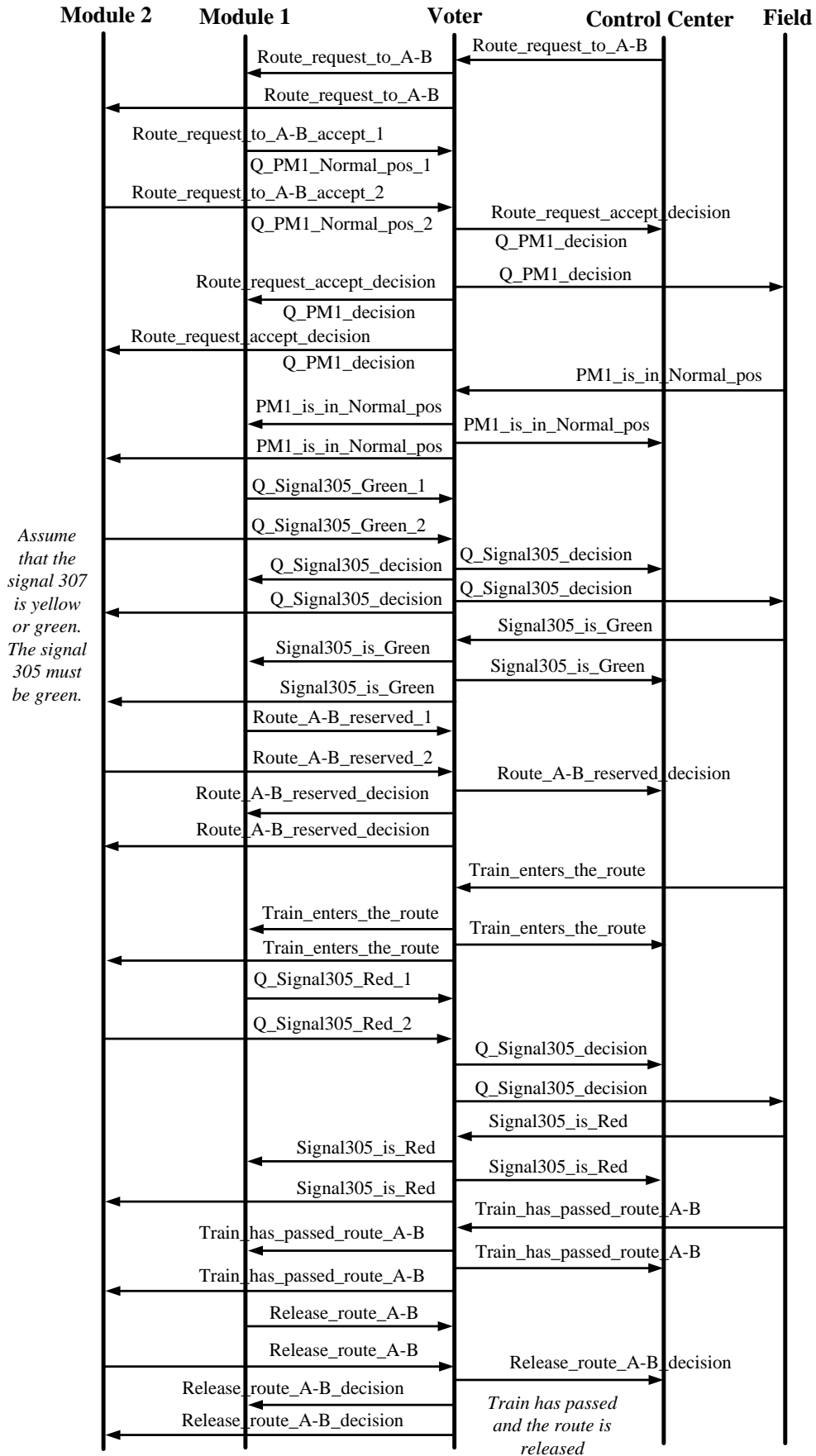
**Table 2.7 :** The safe-states of the variables.

Variable that has logic “1” safe-state	Variable that has logic “0” safe-state
Route request accepted*	Route request accepted*
Route is reserved*	Route is reserved*
Cancel route reservation	Q_switch_normal_pos**
Route request denied	Q_switch_reverse_pos**
Route cancellation is denied	Q_signal_Green
Switch movement is denied	Q_signal_Yellow
Q_signal_Red	Route is cancelled

\* Before the route reservation, the safe-states of these two variables are assumed as “0” whereas the safe-states of these variables are assumed as “1” after a route is reserved. For example, before reservation of a route, the voter demands full agreement from the modules about “route request accepted bit”. After the reservation, the voter keeps this bit in “1” as long as at least one module keeps it in “1”.

\*\* In addition to the module decisions, the voter also checks the states of some other bits for the final decision. For instance, if the track circuit of a switch is busy, the voter does not change the position of the switch even both of the modules agree, and once the movement of the switch is started, due to safety-requirements, the voter does not leave the switch in a middle position even if both modules stop sending switch movement commands.

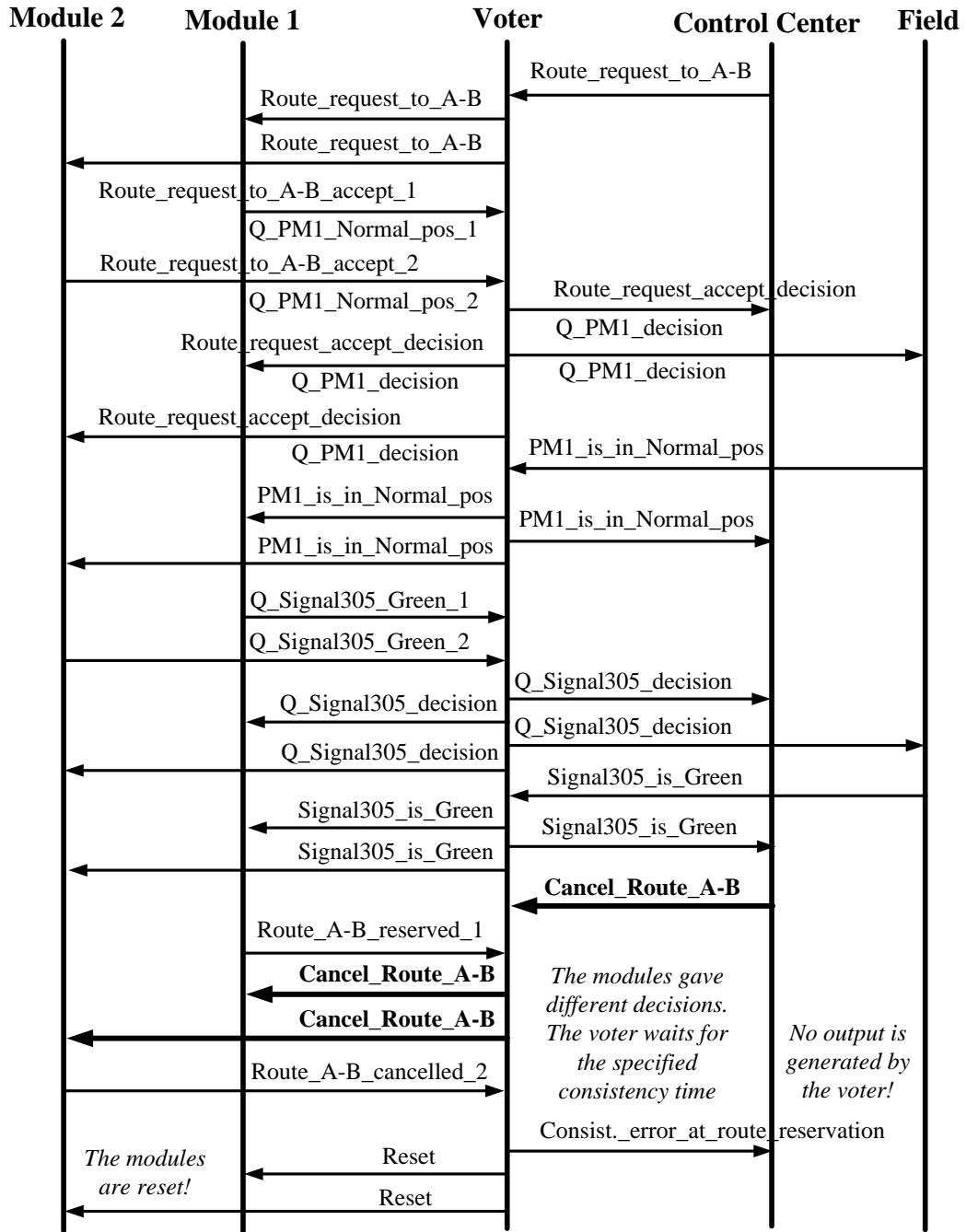
As mentioned above, for the route reservation or moving the PM from one position to another, the voter demands complete agreement of the modules. In other words, reserving a route is much safer than not to reserve it and similarly, keeping a PM in a position is much safer than moving it.



**Figure 2.22 :** The sequence diagram for reservation of the route A-B.

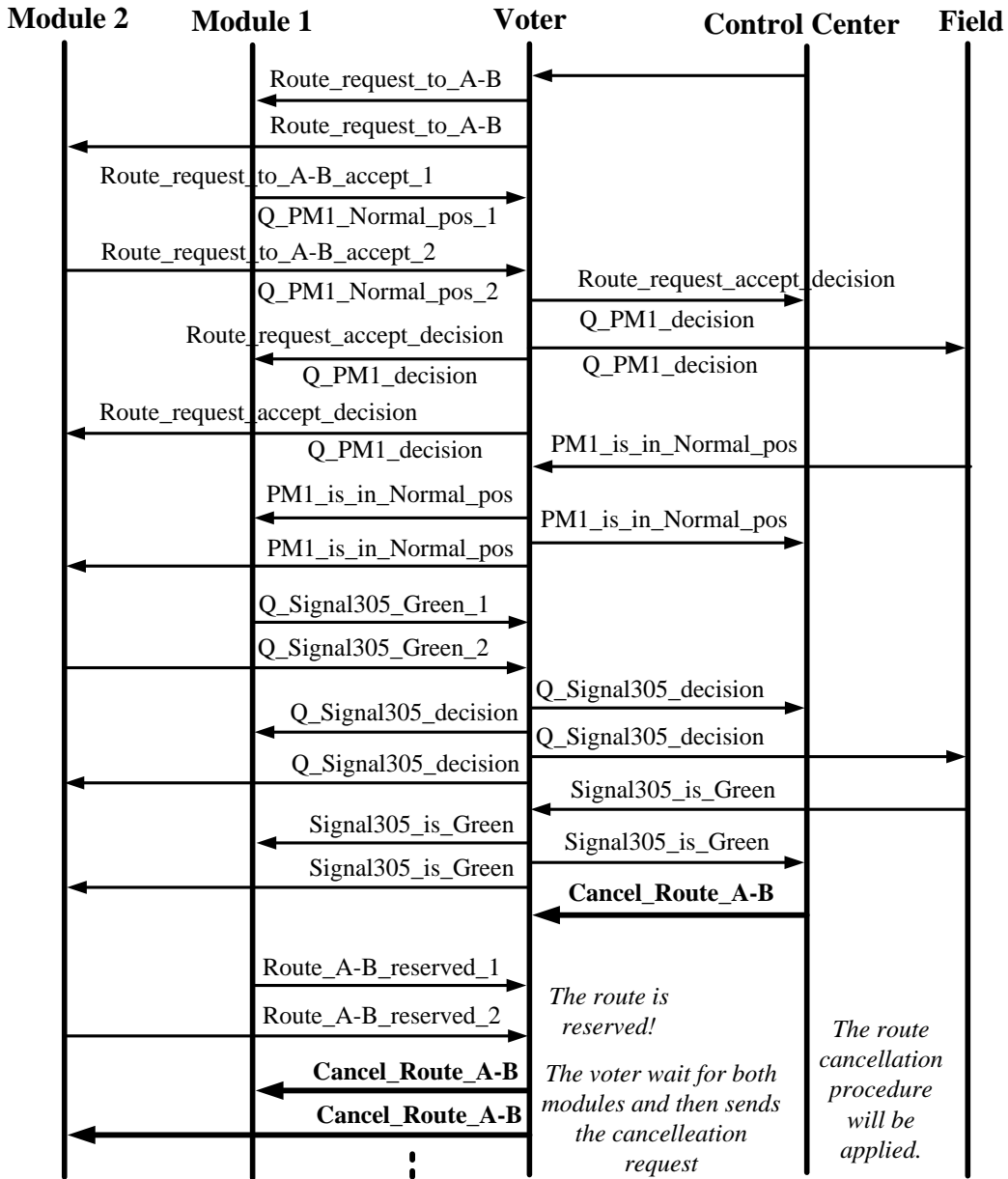
According to the definition of *defensive programming* in [28] (where detection of abnormal control flow, data flow or data values during their execution is determined and software reacts to these problems in a predetermined and acceptable manner), the voter sometimes can give decisions with respect to the decisions of the modules. When a train is moving in a railway region containing a PM, the interlocking software must not change the position of the PM even both modules are in agreement to change the position of the PM.

Likewise, when a PM has begun to change its position after the complete agreement of all modules, the voter has to keep moving of PM until it reaches to its new position even if one or all of the modules change(s) its/their decision. Because leaving a PM in the middle position is forbidden by the operational conditions and safety-requirements of almost all the railways in the worldThe type 1 problem mentioned in section 2.5.1 , can be confronted while reserving a route. The route cancellation request can be made in any time instance, assume that, a route cancellation request is made from the control center after the modules accept the route request and send proper signals to the voter. If Module 1 has shorter cycle time, then it receives the position indication of the point machine and reserves the route before Module 2. If Module 2 receives the route cancellation request in this interval, it can cancel the route request before the route is reserved. Briefly, parallel running modules divaricate to different states. Sequence diagram of this example can be seen from figure 2.23.



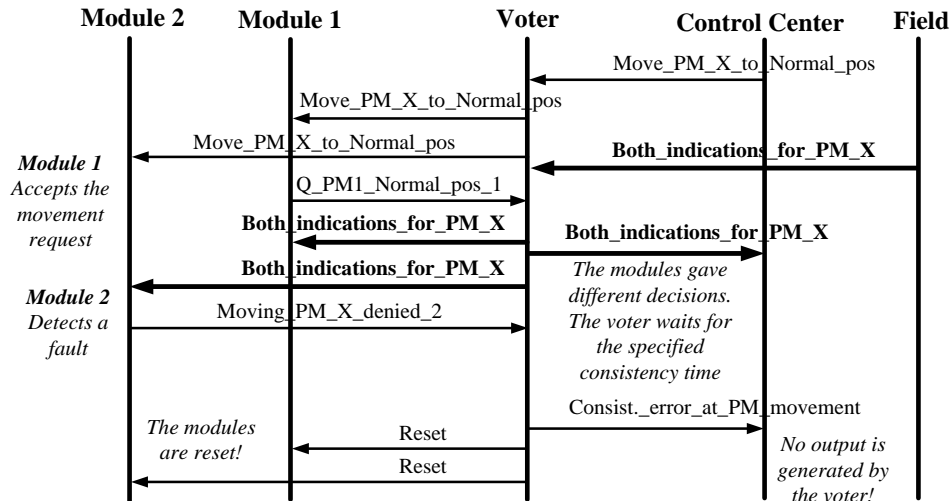
**Figure 2.23** : An example sequence diagram for the type 1 problem.

For the sequence diagram given in figure 2.23, the solution given in section 2.5.2 can be applied. If a module accepts a request, the voter waits until the end of consistency time for the response of the other module and does not send any incoming cancellation request to the modules. This is illustrated in figure 2.24.



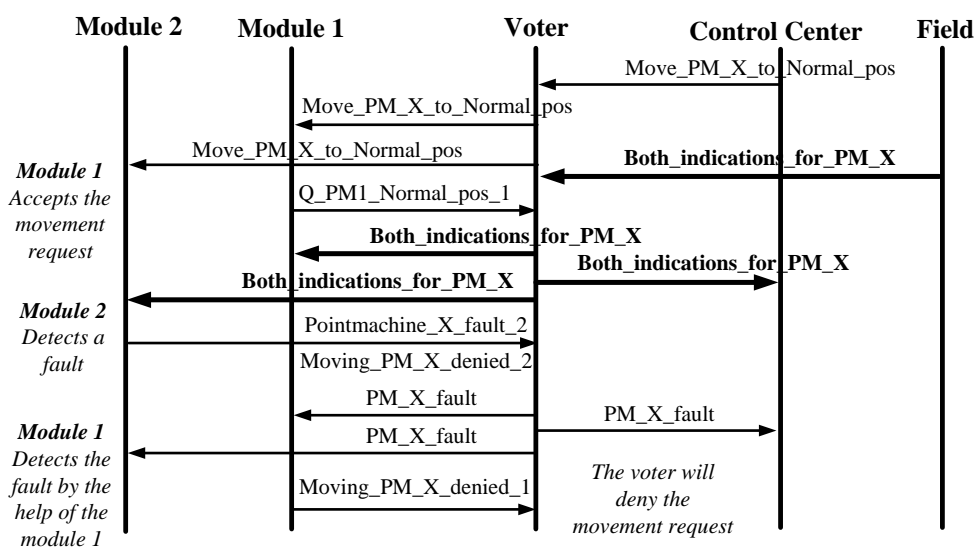
**Figure 2.24 :** The sequence diagram of possible solution for the type 1 problem.

The type 2 problem can be encountered when detecting the faults of the railway field components. Assume that, when a field component malfunction has occurred, the module with shorter cycle time detects this fault first and moves into the failure state. In this state, the module will reject all incoming requests related with the faulty field component, whereas the other module with longer cycle time might not catch this fault. Briefly, parallel running modules can divaricate to different states. An example sequence diagram for this case is given in figure 2.25.



**Figure 2.25 :** An example data sequence diagram for the type 2 problem.

By using the table 2.7 and the solution given in section 2.5.2 , the type 2 problem given in figure 2.25 can be solved as follows: If a module detects a failure, it sends an identification signal to the voter about the failure. When the voter receives such a failure signal from any module, the voter sends proper signals to both modules. This signal enables the other module to detect the fault. In such a condition, the module that detects the fault by the help of the voter does not change its state and does not produce any output related with the faulty situation but give decisions by taking into the fault. Therefore all incoming requests from the control center related with this faulty component will be rejected by both modules. This is illustrated in figure 2.26.



**Figure 2.26 :** The data sequence diagram of possible solution for the type 2 problem.

## **2.6 Testing the signaling software**

Testing signaling software is at least as important as developing it. The related standards [2] and [27] provide several guidelines for testing such crucial software. Before explaining the signaling software test steps, it is useful to remind that, all tests have to be logged and must begin from the very beginning whenever a test step failed. In TNRSP, testing of the developed signaling software is achieved in five steps.

### **2.6.1 Signaling software module tests (Verification)**

Module tests [53] or in other words, testing the developed program modules separately as mentioned in the V-model are realized by using the Interlocking Test Program (ITP) developed by the Department of Control Engineering of Istanbul Technical University (ITU). In this program, a basic simulator is used to imitate the signals in real railway field. An automatic testing procedure was applied to each signaling software modules which are developed by independent groups (2~3 weeks) [46].

### **2.6.2 Signaling software simulator tests (Validation)**

Validation was realized by using a complex software simulator which is connected to the signaling software. Simulator was connected to the signaling software through a second PLC set using digital I/O modules, and all possible signals in the field are imitated by this second PLC. Complex train movements with several stress tests can be performed by the software simulator [54].

### **2.6.3 Hardware simulator tests (Commissioning)**

A hardware simulator, which is 1:87 scaled model of the real railway field, is used to provide an even more realistic simulation. The signals collected from the model railway field are delivered to the developed signaling software using a set of PLCs in this simulation model [54]. The hardware simulator was established in the Industrial Automation Laboratory of the Control Engineering Department of ITU.



#### **2.6.4 Factory acceptance tests (FAT)**

These tests are also realized in the Industrial Automation Laboratory of the Control Engineering Department of ITU, under the observation of a group of delegates from the Turkish State Railways. The whole system (including the signaling software and TCC software, which was developed by TUBITAK) has been validated in these tests.

#### **2.6.5 Site acceptance tests (SAT)**

The final site acceptance tests are realized by a group of delegates from the Turkish State Railways in the real railway field in Adapazarı region in Sakarya, Turkey in January, 2012. All the tests are passed successfully. It should also be noted that the tests are executed by independent groups and documented carefully as suggested by the CENELEC standards.

### **2.7 Results and Discussion**

Consequently, while designing an interlocking software, the use of formal methods and several software architectures are highly recommended by the railway related safety standards to achieve the desired SIL. In this section the use of highly recommended methods, design steps and implementation of these methods to fixed-block signaling systems are discussed.

An example railway field is also examined to increase the clarity of the definitions. The proposed control architecture includes two parallel modules that are supervised by a main controller named as the voter. The voter takes into account the safe-states of each variable for making final decisions. Moreover, possible deadlocks and synchronization problems between the modules due to differences in their cycle times and design strategy are determined. Solutions to overcome these problems are proposed as one of the main contributions of this thesis.

A signaling system using this architecture has been realized in Adapazarı region, Sakarya, Turkey. Lastly, TNRSF is the first signaling system development project for Turkish State Railways and supported by The Scientific and Technological Research Council of Turkey (TUBITAK) with the project number 108G186. Additionally, designing signaling software for the fixed-block signaling systems by

considering the fault diagnosis approaches can be considered as future works. Fault diagnosis approach enables the designers to check the correctness of their component models before developing the signaling software.

### **3. MOVING-BLOCK SIGNALING SYSTEMS**

Since the train movement in conventional railway systems mainly relies on route reservation procedure, the trains have to leave at least one railway block between them while moving on the same railway line. As a result of this, overall capacity of the railway lines are not used efficiently. This is particularly important in metro and urban lines. Moving-block signaling system is defined to increase transport capacity and reduce headways, where trains including their length and safe braking distances are assumed as moving blocks. European Rail Traffic Management System (ERTMS) is the combination of European Train Control System (ETCS) and Global System for Mobile Communications for Railways (GSM-R) which can be also assumed as a standard for safety, signaling and communication system for railways across Europe (and also World-wide). ERTMS increases railway capacity, decreases energy consumption and optimizes train speeds. Another main purpose of ERTMS is to unify different national signaling and train control systems in Europe. In addition to European countries, ERTMS is also in use in Mexico, South Korea, China, Thailand, Taiwan, Australia and Turkey [55].

ERTMS has three application levels from 1 to 3. Application level 1 and 2 can be regarded as fixed-block signaling systems with ATS (Automatic Train Stop) and ATP (Automatic Train Protection) features [3] whereas the application level 3 is considered as moving-block signaling systems [56]. This application level also allows reducing headways when compared with fixed-block signaling systems. In this section, ERTMS and its application levels, the calculation of the train braking distance and the train dynamics are defined briefly and an intelligent adaptive control technique is applied to control the speed of a trailing train.

#### **3.1 European train control system (ETCS)**

ETCS was established by cooperation of railway people in Europe such as UIC (International Union of Railways), UNIFE/UNISIG (European Rail Industry / Union

Industry of Signaling) and ERA (European Railway Agency). ETCS has mainly three application levels as explained in detail below.

### **3.1.1 Application Level 0**

If an ETCS equipped vehicle is used on a route without ETCS equipment then this is considered as level 0. The national rules and requirements should be obeyed by the train drivers.

### **3.1.2 Application level 1**

Wayside optical signal lights inform train drivers about the occupation of the track in front of them. Train - track communication is achieved over balises (Eurobalise<sup>®</sup>). Trains cannot pass the balise as long as the next signal light is red. On-board train computer named Eurocab<sup>®</sup> receives the permitted speed limit (movement authority - MA) over balises, compares with the actual speed of the train and calculates the train braking distance, if necessary. All essential information is displayed to the driver over DMI (Driver Machine Interface) [57]. Track circuits are used to detect the train occupation in railway blocks. If the train passes the related balise while the related signal light is red, then it will stop automatically by the Eurocab<sup>®</sup>, or if the driver does not react in time for a signal change then the train will slow down by its own.

### **3.1.3 Application level 2**

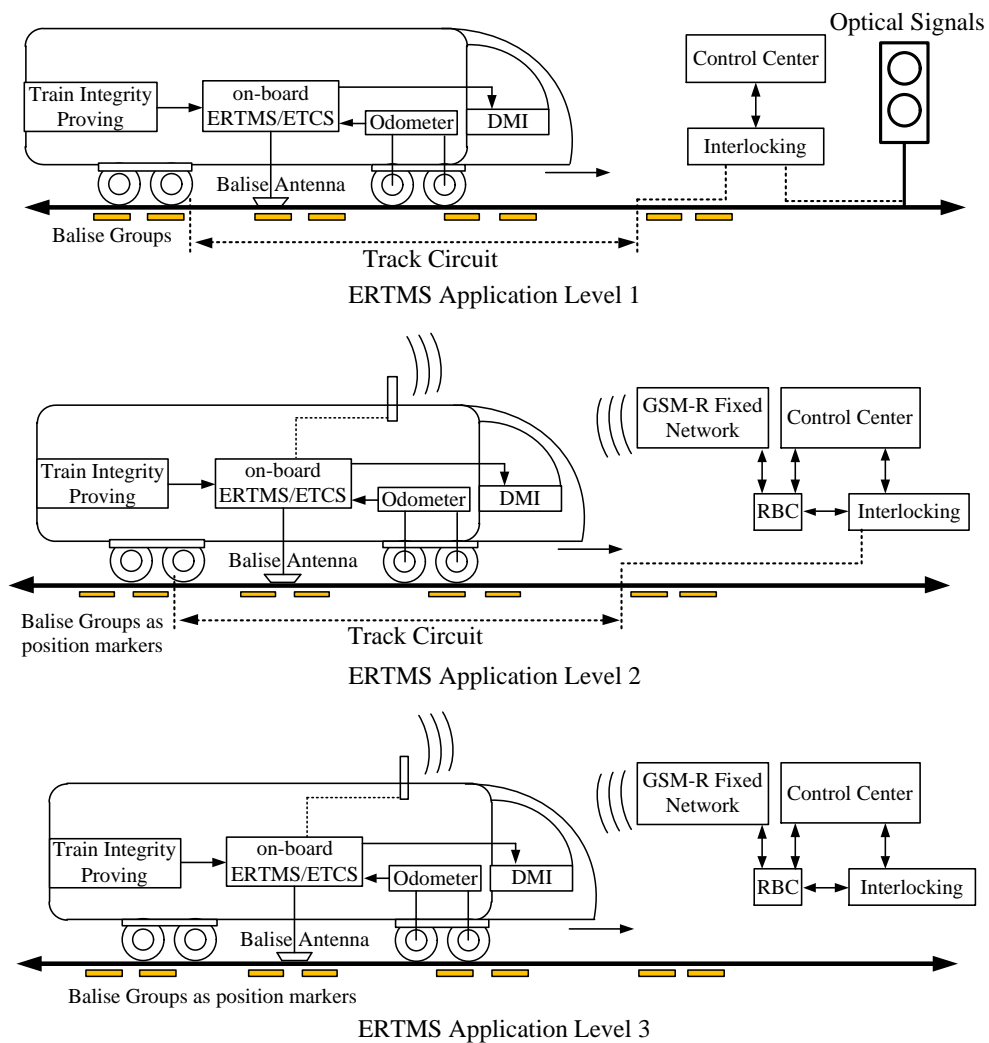
In this application level, movement authority is sent to on-board train computer directly from a RBC (Radio Block Center) via GSM-R instead of balises. There is no need to wayside signals and Eurocab<sup>®</sup> is always up to date. Balises are used as position markers and sends fixed messages such as location and gradient.

### **3.1.4 Application level 3**

This level is a kind of a moving-block signaling system. In addition to the optical wayside signals, track circuits are also removed to decrease infrastructure maintenance and costs. Train itself is considered as a moving-block and allows a continuous train movement. Similar to level 2, balises send fixed messages and all other information is send directly to on-board train computer over GSM-R. ERTMS levels are illustrated in figure 3.1.

### 3.2 GSM for railways (GSM-R)

GSM-R [58] standard combines all past experiences and key functions from systems that used previously in Europe. GSM-R enables communication between RBC and trains without any data loss up to very high speeds (500km/h). GSM-R is mainly based on EIRENE (European Integrated Railway Radio Enhanced Network) and MORANE (Mobile Radio for Railway Networks in Europe) specifications determined by UIC [59].



**Figure 3.1 :** ERTMS application levels from 1 to 3.

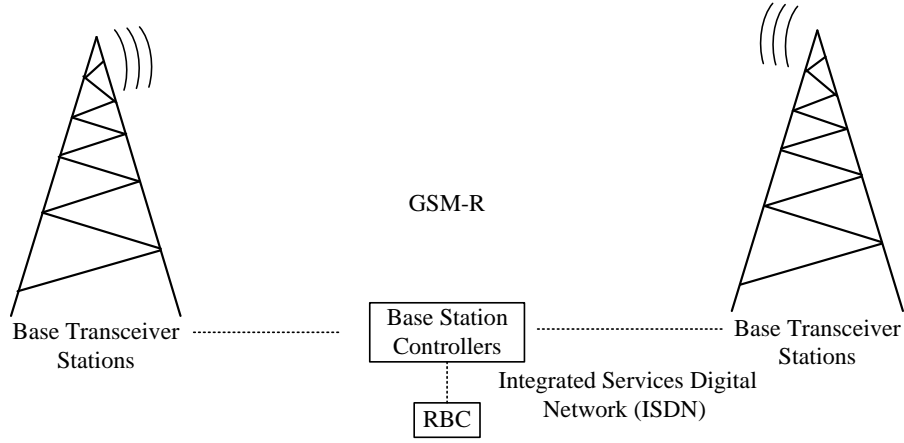
### 3.3 Calculating the train braking distance

In order to prohibit collisions in moving-block systems, trains have to leave adequate distance (safe stopping distance) while following each other. In fixed-block signaling systems, the length of the railway blocks are fixed and same braking distance is used

for all kinds of trains. While calculating braking distance in moving-block systems, the factors including the speed of the train when brakes are applied, brake delay time, the railway track gradient, the mass distribution of the train and etc. have to be considered [60]. For example, for a train with length 410m that has 300km/h speed cannot get closer than 4000m to the train in front (if all conditions run perfectly) [61]. In ERTMS, the braking curves are also updated depending on the train speed and the MAs and Eurocab<sup>®</sup> always keeps the maximum allowed speed limits by communicating with the lineside equipment, interlocking system and etc. [62].

MA first uploaded to Eurocab<sup>®</sup> before leaving the station and while train is moving it communicates over GSM-R to the nearest RBC and sends essential information (speed, location etc.) about the train. This information is evaluated by the interlocking system and then the new MA is sent to the rear train's on-board computer to update the DMI of the rear train. While moving in the railway line, End of Authority (EOA) messages could be received depending on the conditions and new MAs can be uploaded to trains.

Every railway line has a permitted speed limit depending on the environmental and operational conditions that the driver has to obey while moving in that railway line. If the driver increases the train speed and exceeds the permitted speed limit (warning limit) a warning will be screened on the DMI. This warning will remain on the DMI until train's speed is equal or below the permitted speed limit. If driver does not care the warning limit and keeps the train speed over the limit, the service brake will be triggered until the train's speed is equal or below the permitted speed limit. Another speed prevention limit is known as emergency brake limit. If the train exceeds this limit, an emergency brake will be triggered until the train's speed is equal or below the permitted speed limit. This prevention is used when the service brake is not available or the train passes the EOA [63] (the train has to remain at a standstill until a new MA is available). GSM-R network and the communication scheme are given in figure 3.2 [62].



**Figure 3.2** : ERTMS communication architecture.

### 3.3.1 Train braking distance calculation

As it is mentioned before, several factors including track gradient, mass of train and etc. influences the train braking distance. An example calculation including the average gradient is given by [60]:

$$maS + \frac{1}{2}mU^2 + mg(h_1 - h_2) = 0 \quad (3.1)$$

where

- $m$  : is the mass of the train,
- $a$  : is the acceleration rate ( $m/s^2$ ) (deceleration is negative acceleration),
- $S$  : is the stopping distance (m),
- $U$  : is the speed at which the deceleration began (m/s),
- $h_1$  : is the height at which the deceleration began (m),
- $h_2$  : is the height at the stopping point (m),
- $g$  : is the acceleration due to gravity ( $9.79 m/s^2$ ),
- $\alpha$  : is the angle of slope.

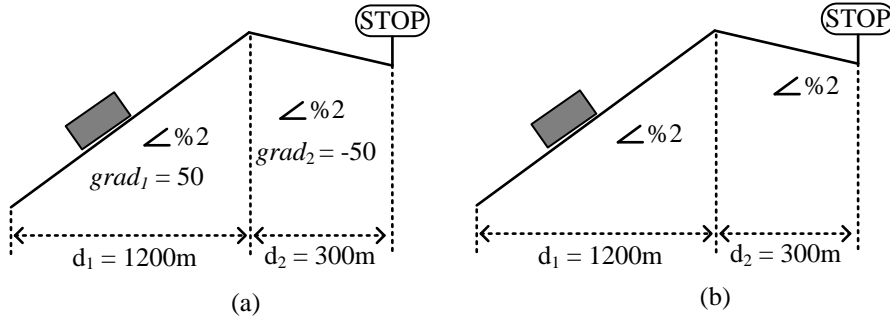
For the railway lines, for small  $\alpha$  values,  $\tan(\alpha)$  equals  $\sin(\alpha)$  and  $\sin(\alpha)$  is the change in height over the stopping distance  $S$ .

$$(h_2 - h_1) = S \sin(\alpha) = S \tan(\alpha) \quad (3.2)$$

Substituting (3.2) into (3.1)

$$S = -\frac{U^2}{2(a - g \tan(\alpha))}, \text{ for } a < 0 \quad (3.3)$$

The term  $-g \tan(\alpha)$  is the gravitational acceleration. For uphill track gradients, where  $h_2 \geq h_1$ , gravity assists deceleration. Examples on the calculation of the braking distance are given in figure 3.3 [60].



**Figure 3.3 :** Average gradient concept and limitations [60].

The calculation of the average gradient ( $G$ ) for the path given in figure 3.3 (a) is as follows:

$$G = \frac{D}{\frac{d_1}{grad_1} + \frac{d_2}{grad_2} + \dots + \frac{d_n}{grad_n}}, \quad D = d_1 + d_2 + \dots + d_n \quad (3.4)$$

$$G = \frac{(1200 + 300)}{\left(\frac{1200}{50}\right) + \left(\frac{300}{-50}\right)} = 83.33(\%1.2)$$

In figure 3.3 (b), the braking distance of a train going with 80 km/h speed and  $-0.3 \text{ m/s}^2$  acceleration rate is calculated by,

$$S = \frac{V^2 - U^2}{2(a - g \tan(\alpha))} \quad (3.5)$$

where  $V^2$  is the final speed of the train. By using (3.5), the braking distance can be calculated as follows:

$$S = \frac{0^2 - (80/3.6)^2}{2(-0.3 + (9.79/83.33))} = 591 \text{ m.} \quad (3.6)$$



The speed of the train at the change of gradient so as to stop at the stop point can be also calculated by using (3.3),

$$U_{300} = \sqrt{-2(-0.3 + (9.79 * 0.02)300)} = 7.9 \text{ m/s.} \quad (3.7)$$

The distance to slow down to 7.9 m/s from 80 km/h is:

$$S_{300} = \frac{7.9^2 - (80/3.6)^2}{2(-0.3 - (9.79 * 0.02))} = 435 \text{ m.} \quad (3.8)$$

$$\text{Braking distance} = 435 + 300 = 735 \text{ m}$$

In all the above equations, brake delay term is ignored. By considering the brake delay time in the above equations and assuming that the gradient is constant, the braking distance can be calculated as follows [60]:

$$S = -\frac{(U + bt_d)^2}{2(a + -g \tan(\alpha))} - Ut_d - (-g \tan(\alpha)) \frac{t_d^2}{2} \quad (3.9)$$

where

- $S$  : is the stopping distance (m),
- $U$  : is the speed of the train when brake command was received (m/s),
- $a$  : is the acceleration provided by the braking system ( $\text{m/s}^2$ ),
- $g$  : is the acceleration due to gravity ( $9.79 \text{ m/s}^2$ ),
- $t_d$  : is the train's brake delay time (sec).

Moreover, a braking distance calculation given by UIC 546 [62] is as follows:

$$S = \frac{\Phi V^2}{1.09375 \left( \frac{\% \lambda}{100} \right) + 0.127 - 0.235(\pm i) \Phi} \quad (3.10)$$

where

- $S$  : is the stopping distance (m),
- $V$  : is the speed of the train (km/h),
- $\% \lambda$  : is the braking efficiency (has no physical meaning),
- $i$  : is the gradient (%),
- $\Phi$  : is the speed dependent constant.

If there is a constant deceleration, the deceleration value can be determined by using

$$a = \frac{V^2}{2S} \quad (3.11)$$

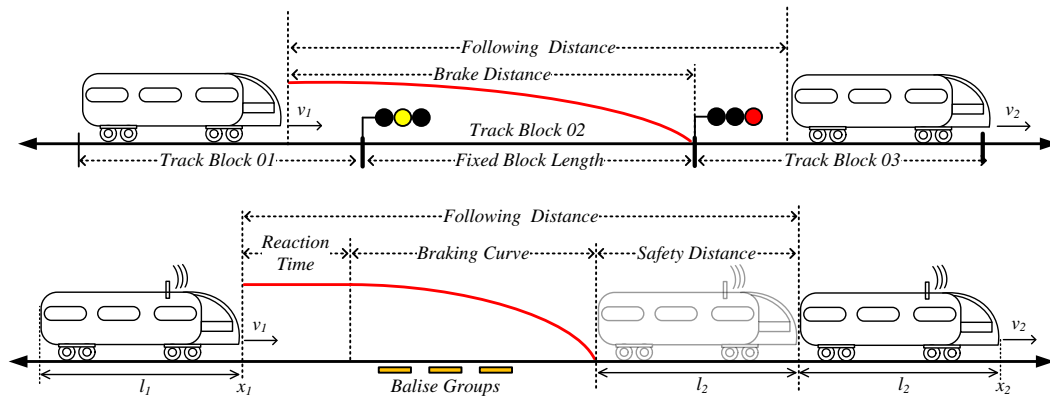
Using (3.10) in (3.11) the following equation is obtained:

$$a_c = \frac{1}{2375} \left( \frac{\% \lambda}{\Phi_c} \right) + \left( \frac{1}{200\Phi_c} \right) - \frac{1}{2375} i \quad (3.12)$$

where

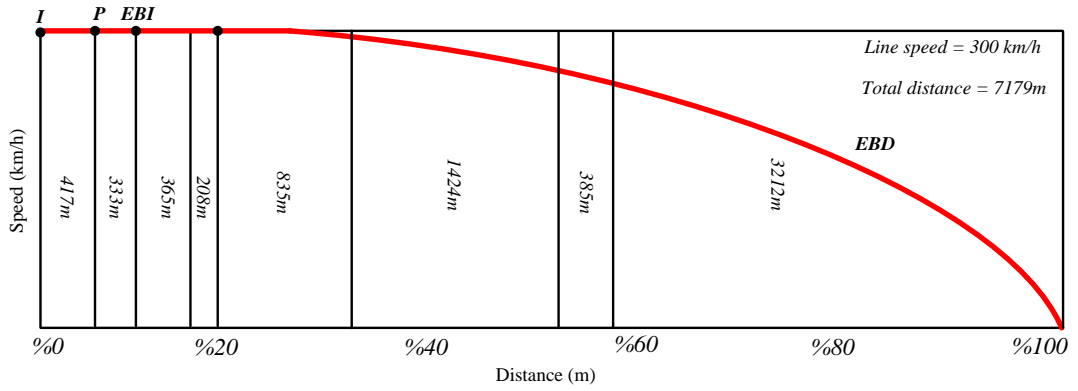
- $S$  : is the stopping distance (m),
- $V$  : is the speed of the train (km/h),
- $a_c$  : is the constant acceleration value ( $m/s^2$ ),
- $\Phi_c$  : is the mean value of  $\Phi$ .

In moving-block systems, more than one train can occupy the same railway block by leaving adequate following distance between them. This following distance can be calculated as the sum of the safe stopping distance, safety margin and the distance corresponding to the brake reaction time. The difference between the fixed-block and the moving-block systems are given in figure 3.4.



**Figure 3.4 : Fixed-block vs. moving-block**

An example braking curve calculation is given in figure 3.5 for a high-speed train with 300km/h speed [61].



**Figure 3.5 :** Stopping distance calculation for a high-speed train.

In figure 3.5, *I* and *P* represents the supervision limits. For the *I* supervision limit, a predetermined time is given to the driver to act for the service brake so that the train does not pass the permitted speed limit. For the *P* supervision limit, a predetermined time is given to the driver after the train reaches to overspeed to act for the service brake so that the train will not overpass the point beyond which ETCS will trigger the command of the brakes [64]. The braking curve related to the speed decrease due to the emergency brake is called *EBD* (Emergency Brake Deceleration) curve. The shape of the EBD curve, for a given piece of track, will therefore vary according to the type of rolling stock. The *EBI* (Emergency Brake Intervention) supervision limit is the point in which ETCS will bypass the human in charge [64].

In addition to [64], several train braking distance calculations can be found in the literature [65-68]. Other practical safe stopping distance calculations explained in [69] are Moving Space Block (MSB), Moving Time Block (MTB) and Pure Moving Block (PMB). These safe stopping distance calculations are as follows [70]:

$$d_{MSB} = \frac{v_{\max}^2}{2(a_s)} + SM \quad (3.13)$$

$$SM = T_r * v_{\max}$$

$$d_{MTB} = \frac{v_{\max} v_f}{2(a_s)} + SM \quad (3.14)$$

$$d_{PMB} = \frac{v_f^2}{2(a_s)} + SM \quad (3.15)$$

where

- $d_{MSB}$  : is the safe stopping distance between two trains for MSB,
- $d_{MTB}$  : is the safe stopping distance between two trains for MTB,
- $d_{PMB}$  : is the safe stopping distance between two trains for PMB,
- $v_{max}$  : is the maximum speed (km/h),
- $v_f$  : is the speed of the rear train (km/h),
- $a_s$  : is the deceleration of the rear train ( $m/s^2$ ),
- $SM$  : is the marginal distance (m),
- $T_r$  : is the react time (sec).

### 3.4 Proposed speed control method for moving-block systems

In this section, the control mechanism proposed in [71] is improved with the use of an adaptive online learning algorithm and applied to the speed control of the trailing train so as to conserve the adequate following distance between trains. In addition to PMB calculation given by (3.15), the following train dynamics are used in simulation studies [72].

$$\begin{aligned}
F(g, v_f) &= aM + R(g, v_f) \\
R(g, v_f) &= R_r + R_g \\
R_r &= 6.374M + 129n + bv_f M \\
R_g &= M \sin(\theta) = M \sin\left[\tan^{-1}(grad/100)\right] \\
P(g, v_f) &= F(g, v_f)v_f
\end{aligned} \tag{3.16}$$

where

- $F(g, v_f)$  : is the traction force (N),
- $P(g, v_f)$  : is the total traction power (W),
- $R_g$  : is the gradient resistance (N),
- $R_r$  : is the rolling resistance (N),
- $v_f$  : is the speed of the train (km/h),
- $a$  : is the acceleration value ( $m/s^2$ ),
- $grad$  : is the gradient of the railway line (%),
- $M$  : is the mass of the train (tons),
- $b$  : is the flange coefficient of the train between 0.0914 and 0.137,
- $n$  : is the total number of axles.

### 3.4.1 Suggested adaptive PD controller based on online LSSVR

This section explains the online adaptive algorithm and the design of proposed PD controller. Given a training data set:

$$(y_1, x_1) \dots (y_k, x_k), \quad x \in R^n, y \in R \quad k=1, 2, \dots, N \quad (3.17)$$

where  $N$  is the size of the training data set and  $n$  is the dimension of the input matrix, respectively. The main aim is to find a function  $f(x)$  that has at most  $\varepsilon$  deviation from the targets  $y_i$  for all training data and as flat as possible [73]. The errors are ignored if they are less than  $\varepsilon$ . A linear regression function  $f(x)$  has the form,

$$f(x) = \langle w, x \rangle + b \quad \text{with } w \in R^n, b \in R \quad (3.18)$$

where  $\langle \cdot, \cdot \rangle$  denotes the dot product in  $R^n$ .

The solution for this training data set can be found by the minimum of the following optimization problem [73], [74]:

$$\min_{(w,b,e)} \frac{1}{2} w^T w + \frac{1}{2} C \sum_{k=1}^N e_k^2 \quad (3.19)$$

subject to

$$y_k - w^T \Phi(x_k) - b - e_k = 0, \quad k=1, 2, \dots, N \quad (3.20)$$

where  $w$  is a vector in feature space  $R$ ,  $C$  is a regularization parameter that provides a compromise between the model complexity and the degree of tolerance to the errors larger than  $\varepsilon$ ,  $\Phi(x_k)$  is a mapping from input space to the feature space and  $b$  is the bias term. In LSSVR, it is aimed to find the global solution by maximizing the geometric margin and minimizing the training error. The minimization problem presented in (3.19) and (3.20) are called the primal form of the optimization problem given in [73] and [74]. Since the objective function in (3.19) is non-convex, dual form for the problem ensuring global minima, can be derived using Lagrangian ( $L$ ) function.

$$L(w, b, e, \alpha) = \frac{1}{2} w^T w + \frac{1}{2} C \sum_{k=1}^N e_k^2 - \sum_{k=1}^N \alpha_k (w^T \Phi(x_k) + b + e_k - y_k) \quad (3.21)$$

where  $\alpha_k$  are the Lagrangian multipliers. For optimality, primal variables have to vanish at the saddle point:

$$\begin{aligned}
\frac{\partial L}{\partial b} &= \sum_{k=1}^N \alpha_k = 0 \\
\frac{\partial L}{\partial w} &= \underline{w} - \sum_{k=1}^N \alpha_k \Phi(x_k) = 0 \rightarrow \underline{w} = \sum_{k=1}^N \alpha_k \Phi(x_k) \\
\frac{\partial L}{\partial e_k} &= C \sum_{k=1}^N e_k - \sum_{k=1}^N \alpha_k = 0 \rightarrow \alpha_k = C e_k \\
\frac{\partial L}{\partial \alpha_k} &= 0 \rightarrow y_k = w^T \Phi(x_k) + b + e_k, \quad k = 1, 2, \dots, N.
\end{aligned} \tag{3.22}$$

As a result of dual formulation, the optimal solution of the problem can be obtained as follows:

$$\begin{bmatrix} \underline{0} & \underline{1} \\ \underline{1}^T & \Omega_{km} + \frac{I}{C} \end{bmatrix} \begin{bmatrix} b \\ \underline{\alpha}^T \end{bmatrix} = \begin{bmatrix} 0 \\ \underline{y}^T \end{bmatrix} \tag{3.23}$$

with

$$\begin{aligned}
\underline{y} &= [y_1 \quad y_2 \quad \dots \quad y_N], \quad \underline{\alpha} = [\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_N], \\
\underline{1} &= [1 \quad 1 \quad \dots \quad 1], \quad \Omega_{km} = K(x_k, x_m), \quad k, m = 1, 2, \dots, N
\end{aligned} \tag{3.24}$$

The solution in (3.23) is the solution that can be utilized in online learning processes by introducing a time index as in (3.25):

$$\begin{bmatrix} \underline{0} & \underline{1} \\ \underline{1}^T & \Omega_{km} + \frac{I}{C} \end{bmatrix} \begin{bmatrix} b_n \\ \underline{\alpha}_n^T \end{bmatrix} = \begin{bmatrix} 0 \\ y_n \end{bmatrix} \tag{3.25}$$

where

$$\begin{aligned}
\underline{\alpha}_{N_n} &= [\alpha_{1_n} \quad \alpha_{2_n} \quad \dots \quad \alpha_{N_n}], \quad \underline{1} = [1 \quad 1 \quad \dots \quad 1] \\
\Omega_{km_n} &= K(x_{k_n}, x_{m_n}), \quad k, m = 1, 2, \dots, N
\end{aligned} \tag{3.26}$$

The dynamics of a non-linear system, can be represented by the Nonlinear AutoRegressive with eXogenous inputs (NARX) model as in (3.27),

$$y_n = f(u_n, \dots, u_{n-n_u}, y_{n-1}, \dots, y_{n-n_y}) \quad (3.27)$$

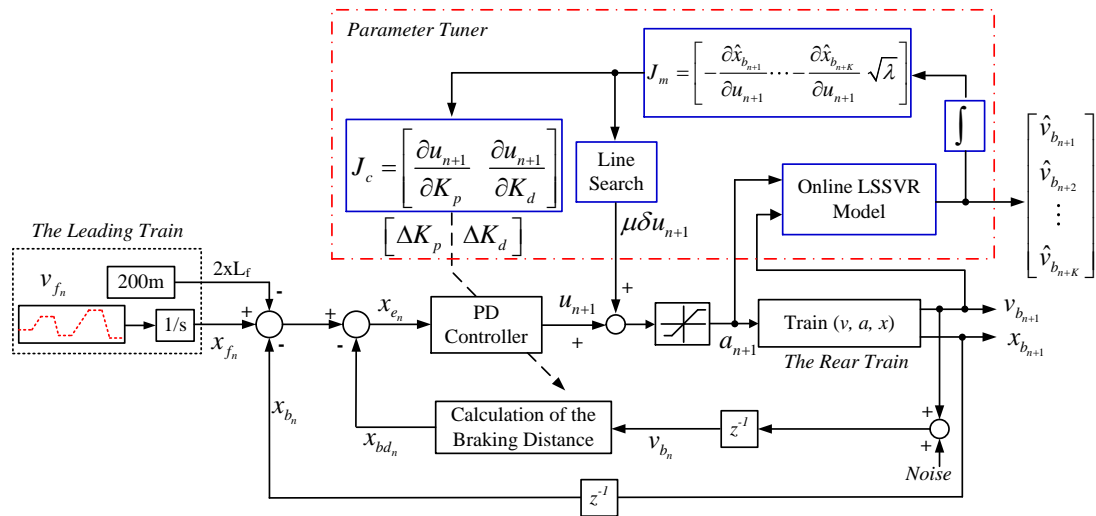
where  $u_n$  is the control input applied to the plant at time  $n$ ,  $y_n$  is the output of the plant, and  $n_u$  and  $n_y$  stand for the number of past control inputs and the number of past plant outputs involved in the model, respectively [71], [75]. The state vector of the system at time index  $n$  is represented as follows:

$$\underline{c}_n = [u_n, \dots, u_{n-n_u}, y_{n-1}, \dots, y_{n-n_y}] \quad (3.28)$$

The output of the model can be written by using equations (3.25), (3.27) and (3.28),

$$\hat{y}_n = \sum_{i=1}^L \alpha_i(n) K(\underline{c}_n, \underline{x}_i) + b_n \quad (3.29)$$

where  $K(\underline{c}_n, \underline{x}_i)$  is a kernel function given by  $K(\underline{c}_n, \underline{x}_i) = \Phi^T(\underline{c}_n)\Phi(\underline{x}_i)$ . The kernel function handles the inner product in the feature space. Detailed information on the online identification procedure of nonlinear systems via online LSSVR can be found in [76-78]. The adaptation mechanism proposed in [71] has been applied so as to conserve the following distance between two trains.



**Figure 3.6 :** Online adaptive PD controller based on LSSVR.

The adaptive structure for PD controller is illustrated in figure 3.6, where  $X_b$  is back train,  $X_f$  is the position of the leading train,  $X_{bd}$  is the bra the position of the king distance, and  $X_{error}$  is the tracking error. The adaptive PD controller consists of four parts: an incremental PD controller, an online LSSVR model of the plant, a line search block (where the golden section algorithm was used) and two jacobian blocks. PMB calculation given by (3.15), is used to calculate the braking distance of the rear train.

The classical incremental PD controller produces a control signal as follows [71], [79].

$$u_{n+1} = u_n + K_{p_n} [e_n - e_{n-1}] + K_{d_n} [e_n - 2e_{n-1} + e_{n-2}] \quad (3.30)$$

where  $e_n = x_{f_n} - x_{b_n} - 2L_f - x_{bd_n}$ .  $K_{p_n}$  and  $K_{d_n}$  are the controller parameters to be tuned. The controller parameters are not at optimal values initially. Owing to this, Online LSSVR model of the plant is utilized to predict the system jacobian information needed to tune controller parameters. The model produces a trajectory vector  $[\hat{v}_{b_{n+1}} \quad \hat{v}_{b_{n+2}} \quad \cdots \quad \hat{v}_{b_{n+K}}]$  that is composed of K-step ahead future behavior of the plant when the output of the PD controller is repeatedly applied to the plant exactly K times [71]. Based on these predictions, parameters of the controller are updated in a manner such that the sum of the squared K-step ahead prediction errors is minimized with minimum deviation in the control action [71]. The objective function to be minimized is selected as:

$$E(u_{n+1}) = \frac{1}{2} \sum_{k=1}^K [x_{f_{n+k}} - \hat{x}_{b_{n+k}} - 2L_f - \hat{x}_{bd_{n+k}}]^2 + \frac{1}{2} \lambda [u_{n+1} - u_n]^2 \quad (3.31)$$

where  $K$  is the prediction horizon and  $\lambda$  is the penalty coefficient. The second term in the summation in (3.31) is used to restrict the deviation of the control signal. Levenberg-Marquardt [80], [81] algorithm has been utilized to tune the controller parameters as in (3.32).



$$\begin{bmatrix} K_p^{new} \\ K_d^{new} \end{bmatrix} = \begin{bmatrix} K_p^{old} \\ K_d^{old} \end{bmatrix} - (\mathbf{J}^T \mathbf{J} + \mu \mathbf{I})^{-1} \mathbf{J}^T \hat{\underline{\mathbf{X}}}_{error} \quad (3.32)$$

where  $\mu$  is a parameter that yields a compromise between the steepest descent and the Gauss-Newton algorithms [71],  $\mathbf{I}$  is 2x2 identity matrix,  $\mathbf{J}$  is a  $(K+1) \times 2$  Jacobian matrix given by (3.33).

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \tilde{x}_{e_{n+1}}}{\partial K_{p_n}} & \frac{\partial \tilde{x}_{e_{n+1}}}{\partial K_{d_n}} \\ \vdots & \vdots \\ \frac{\partial \tilde{x}_{e_{n+K}}}{\partial K_{p_n}} & \frac{\partial \tilde{x}_{e_{n+K}}}{\partial K_{d_n}} \\ \sqrt{\lambda} \frac{\partial \Delta u_n}{\partial K_{p_n}} & \sqrt{\lambda} \frac{\partial \Delta u_n}{\partial K_{d_n}} \end{bmatrix} = \begin{bmatrix} -\frac{\partial \hat{x}_{b_{n+1}}}{\partial K_{p_n}} & -\frac{\partial \hat{x}_{b_{n+1}}}{\partial K_{d_n}} \\ \vdots & \vdots \\ -\frac{\partial \hat{x}_{b_{n+1}}}{\partial K_{p_n}} & -\frac{\partial \hat{x}_{b_{n+1}}}{\partial K_{d_n}} \\ \sqrt{\lambda} \frac{\partial \Delta u_n}{\partial K_{p_n}} & \sqrt{\lambda} \frac{\partial \Delta u_n}{\partial K_{d_n}} \end{bmatrix} \quad (3.33)$$

and  $\hat{\underline{\mathbf{X}}}_{error}$  is prediction error vector given by (3.34).

$$\hat{\underline{\mathbf{X}}}_{error} = \begin{bmatrix} \tilde{x}_{e_{n+1}} \\ \vdots \\ \tilde{x}_{e_{n+K}} \\ \sqrt{\lambda} \Delta u_n \end{bmatrix} = \begin{bmatrix} x_{f_{n+1}} - \hat{x}_{b_{n+1}} - 2L_f - \hat{x}_{bd_{n+1}} \\ \vdots \\ x_{f_{n+K}} - \hat{x}_{b_{n+K}} - 2L_f - \hat{x}_{bd_{n+K}} \\ \sqrt{\lambda} \Delta u_n \end{bmatrix} \quad (3.34)$$

In order to eliminate drawbacks resulting from modeling inaccuracies in transient-state and external disturbance in steady-state and to force system toward the desired reference, a control signal correction term ( $\delta u_{n+1}$ ) has been proposed in [71]. The corrector term is computed using gradient vector as follows:

$$\delta u_{n+1} = \frac{-\mathbf{J}_m^T \hat{\underline{\mathbf{X}}}_{error}}{\mathbf{J}_m^T \mathbf{J}_m} \quad (3.35)$$

where

$$\begin{aligned}
\mathbf{J}_m &= \begin{bmatrix} \frac{\partial \hat{x}_{b_{n+1}}}{\partial u_{n+1}} & \dots & -\frac{\partial \hat{x}_{b_{n+K}}}{\partial u_{n+1}} & \sqrt{\lambda} \end{bmatrix}^T \\
&= \begin{bmatrix} \frac{\partial \hat{x}_{b_{n+1}}}{\partial \hat{v}_{b_{n+1}}} \frac{\partial \hat{v}_{b_{n+1}}}{\partial u_{n+1}} & \dots & -\frac{\partial \hat{x}_{b_{n+K}}}{\partial \hat{v}_{b_{n+K}}} \frac{\partial \hat{v}_{b_{n+K}}}{\partial u_{n+1}} & \sqrt{\lambda} \end{bmatrix}^T \\
&= \begin{bmatrix} -T_s \frac{\partial \hat{v}_{b_{n+1}}}{\partial u_{n+1}} & \dots & -T_s \frac{\partial \hat{v}_{b_{n+K}}}{\partial u_{n+1}} & \sqrt{\lambda} \end{bmatrix}^T
\end{aligned} \tag{3.36}$$

The Jacobian matrix can be split up into two blocks and calculations can be simplified by using chain rule.

$$\mathbf{J} = \begin{bmatrix} -T_s \frac{\partial \hat{v}_{b_{n+1}}}{\partial u_{n+1}} \\ \vdots \\ -T_s \frac{\partial \hat{v}_{b_{n+K}}}{\partial u_{n+1}} \\ \sqrt{\lambda} \end{bmatrix} \begin{bmatrix} \frac{\partial u_{n+1}}{\partial K_{p_n}} & \frac{\partial u_{n+1}}{\partial K_{d_n}} \end{bmatrix} = \mathbf{J}_m \mathbf{J}_c \tag{3.37}$$

Control signal corrector term and controller parameters can be easily obtained by computing  $\mathbf{J}_m$  and  $\mathbf{J}_c$ .

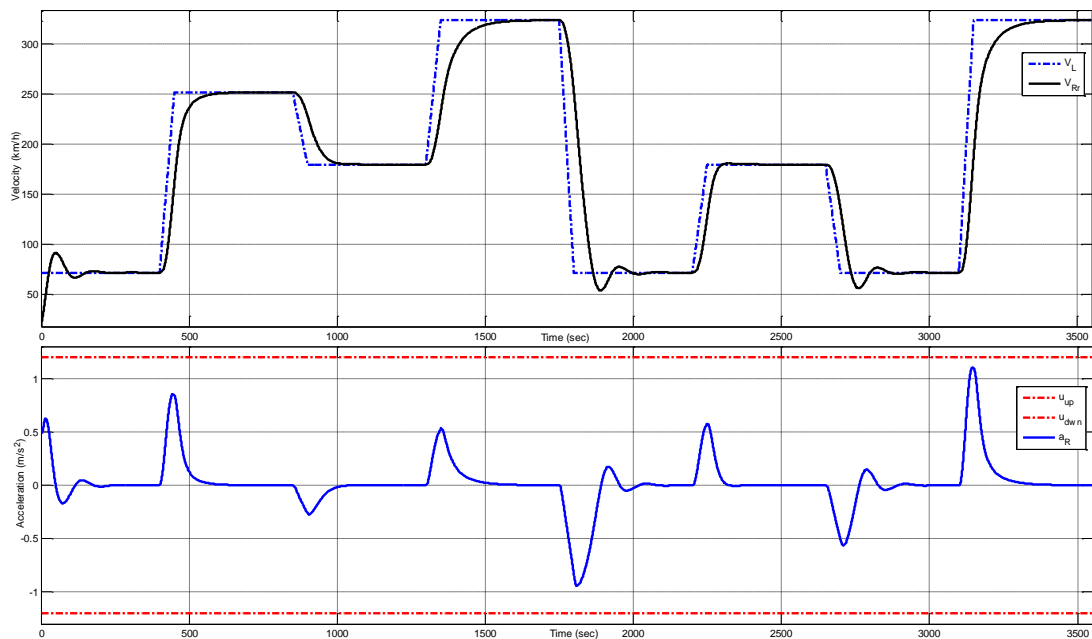
### 3.4.2 Simulation results

The performance of the online adaptive PD controller is evaluated on a moving-block train system. The main aim is to keep the trains safely away from each other. Firstly, it is assumed that there is no measurement noise or any disturbances applied to the system. Secondly, measurement noise and disturbances are also added to the system. These simulations are made for fixed prediction horizon ( $K$ ) and penalty constant ( $\lambda$ ) values, which are chosen as 3 and 0.25, respectively. As mentioned before, to provide a comfortable journey acceleration values have to be between  $-1.2$  and  $1.2 \text{ ms}^{-2}$  which is also satisfied in the simulation studies. The initial values of the PD controller parameters are chosen to be equal to zero, the number of the past inputs ( $n_u$ ) and outputs ( $n_y$ ), which are the order of NARX model, have been selected as 3. The length of the sliding window, which is also the number of the online training data pair, is chosen as 25. The length of each train is assumed as 200 m and

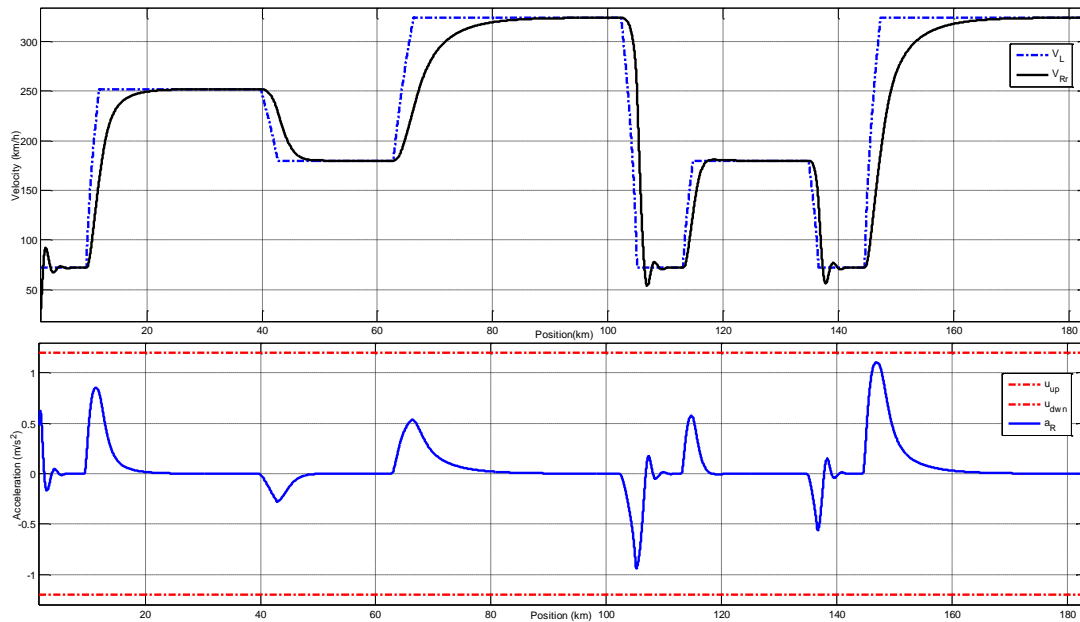
the initial distance between trains is 1500 m. The mass of the train is chosen as 100 tons, the flange coefficient is chosen as 0.13, the number of axles are chosen as 6. Additionally, for the calculation of the braking distance of the rear train, the maximum deceleration is chosen as  $1.4 \text{ m/s}^2$ , the reaction time is chosen as 2.5 sec and the maximum allowed speed is chosen as 110 m/s.

### 3.4.2.1 Noiseless case

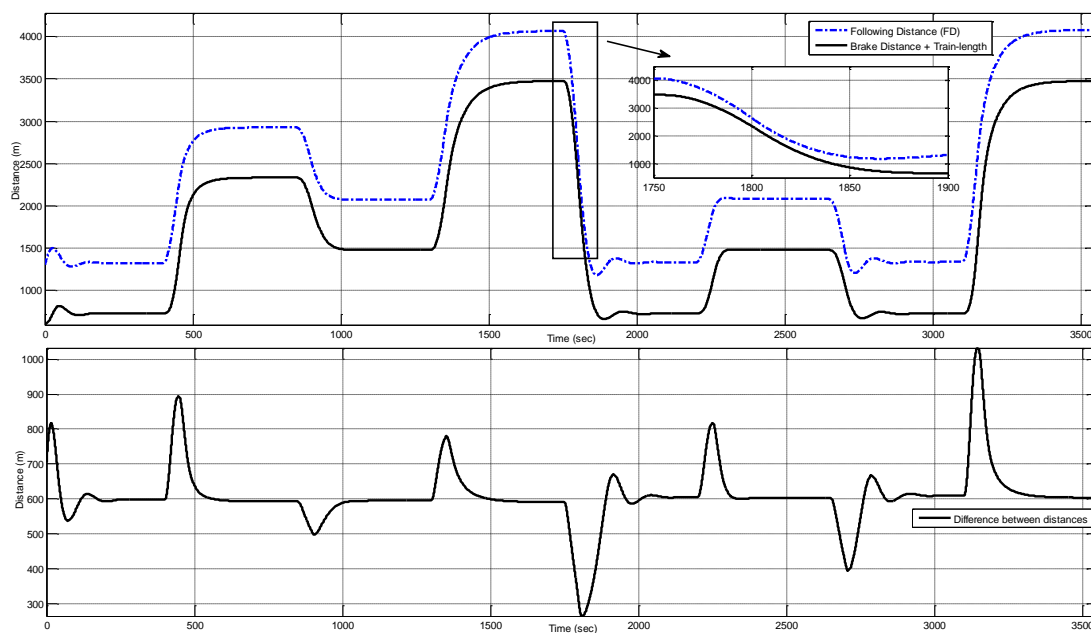
Firstly, it is assumed that there is no measurement noise or any disturbances affecting the system. Simulations for the noiseless case are realized for fixed prediction horizon ( $K$ ) and penalty constant ( $\lambda$ ), which are chosen as 3 and 0.25, respectively. The simulation results are given in figure 3.7, figure 3.8, figure 3.9, figure 3.10 and figure 3.11, respectively.



**Figure 3.7 :** The change of velocity ( $V_{Rr}$ : Velocity of the rear train,  $V_L$ : Velocity of the leading train) and acceleration value of the rear train ( $a_{Rr}$ ) with respect to time.



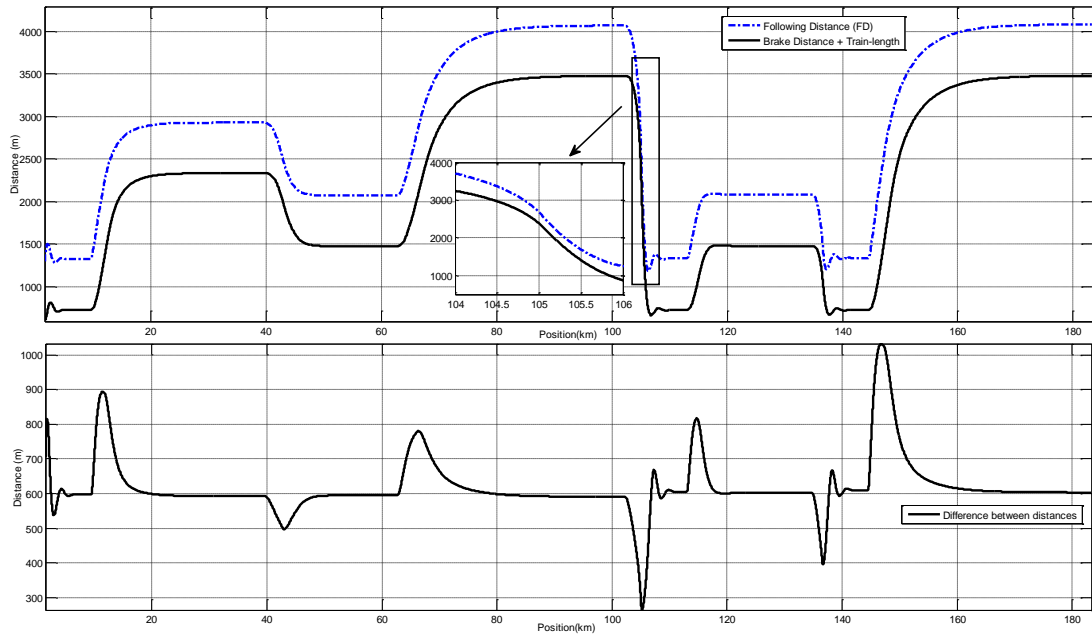
**Figure 3.8 :** The change of velocity ( $V_{Rr}$ : Velocity of the rear train,  $V_L$ : Velocity of the leading train) and acceleration value of the rear train ( $a_{Rr}$ ) with respect to position.



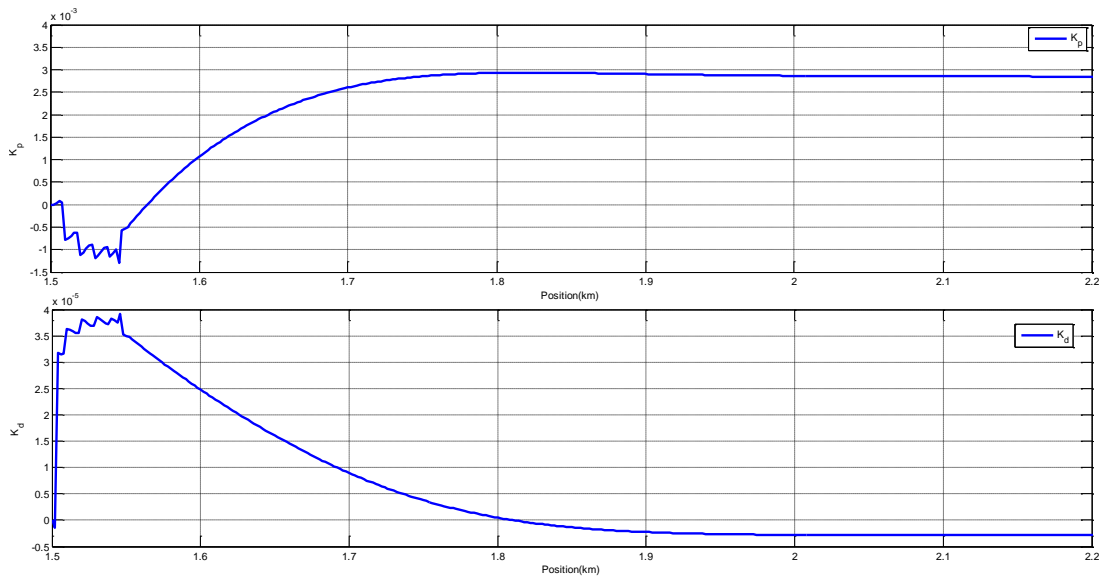
**Figure 3.9 :** The change of FD, BD and the difference between FD and BD with respect to time.

Figure 3.7 (and figure 3.8) indicates the velocities of the trains and the control signal ( $a_{Rr}$ ) applied to the rear train versus time and position, respectively and it is obvious from figure 3.7 and figure 3.8 that the adaptive PD controller attains good tracking performance for the noiseless case. The acceleration of the rear train (the control

signal produced by the adaptive PD controller) is within the limits for a comfortable journey.



**Figure 3.10 :** The change of FD, BD and the difference between FD and BD with respect to position.



**Figure 3.11 :**  $K_p$ ,  $K_d$  values of the PD controller.

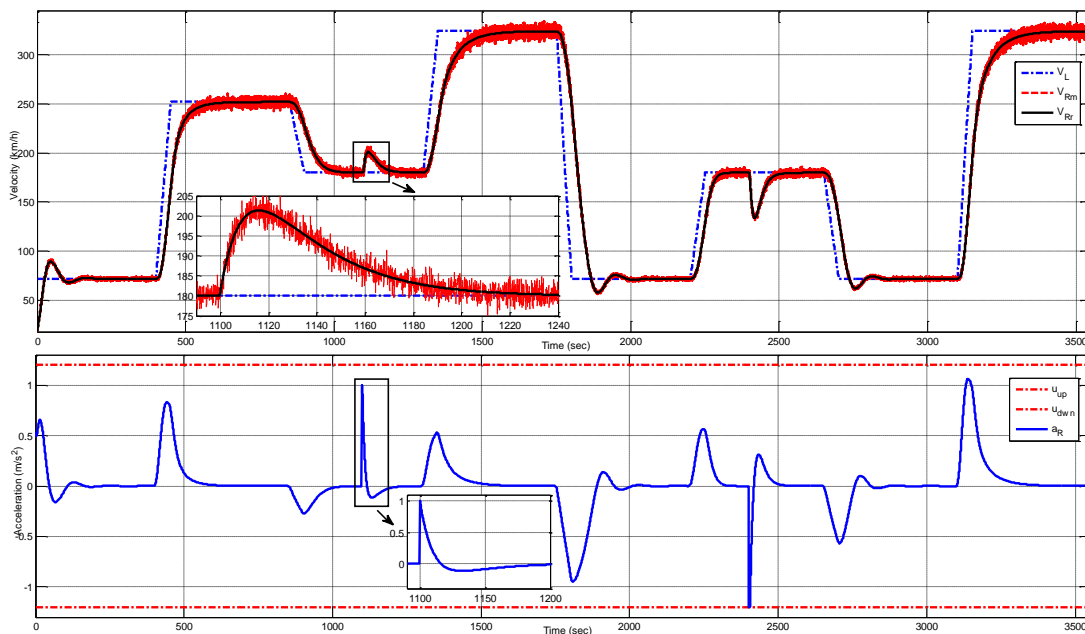
The position between trains called following distance (FD) is bigger than the braking distance (BD) of the rear train, that is, the rear train preserve the FD during journey. The following distance between trains is also kept for the noiseless case (see figure 3.9 and figure 3.10). Finally, the adaptation of the controller parameters are depicted in figure 3.11.

### 3.4.2.2 Noisy case with disturbances

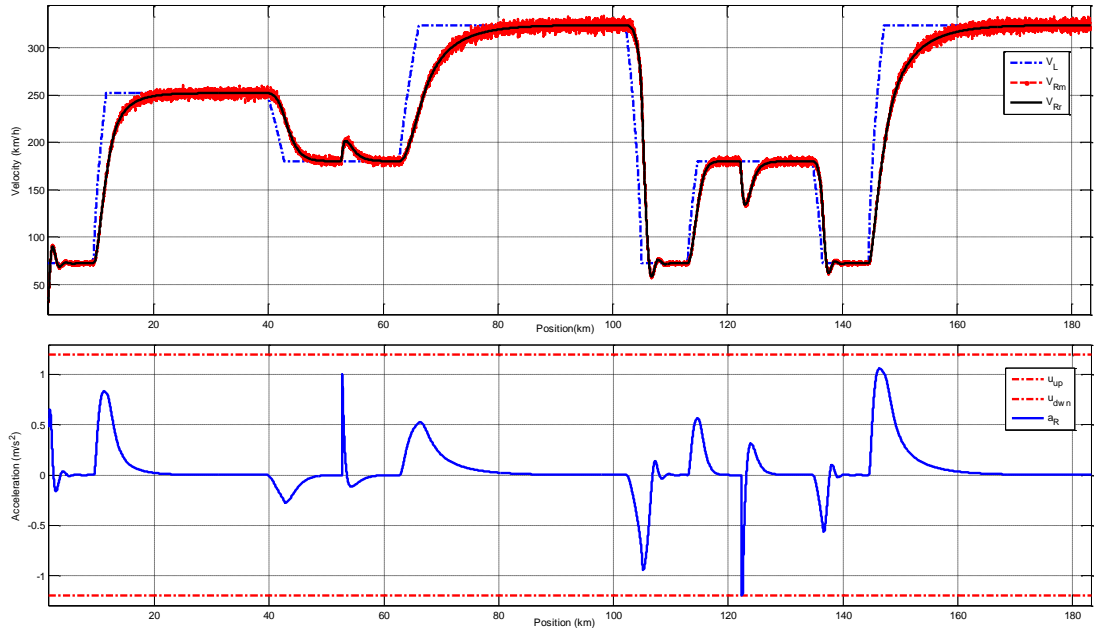
Secondly, in order to evaluate the robustness of the controller with respect to measurement noise and input disturbances; the measured velocity of the rear train is corrupted by an additive zero mean Gaussian noise. The  $K$  and the  $\lambda$  are chosen as 3 and 0.25, as before. The signal-to-noise ratio (SNR) of the additive noise is 40 dB. Moreover, step type disturbances are added to the control signal at some time intervals. SNR is given by the following equation

$$SNR = 10 \log_{10} \left( \frac{\sigma_y^2}{\sigma_v^2} \right) \text{ dBa} \quad (3.38)$$

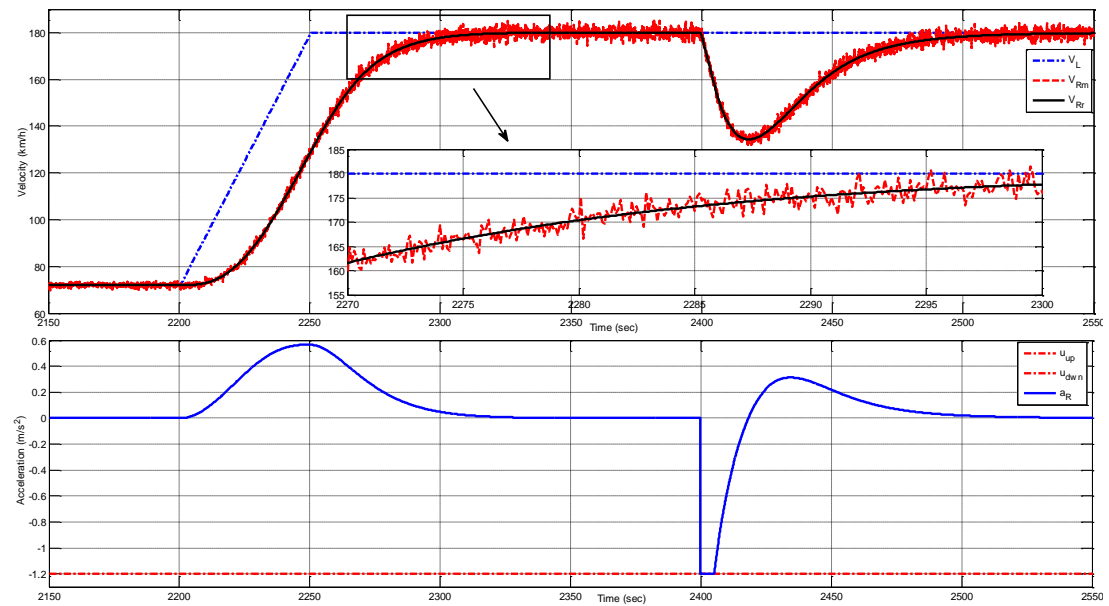
where  $\sigma_y^2$  and  $\sigma_v^2$  are the variances of the measured output of the underlying system and the additive noise, respectively [71]. Simulation results for noisy case with step and sinusoidal type disturbance are given in figure 3.12, figure 3.13, figure 3.14, figure 2.15 and figure 3.16, respectively.



**Figure 3.12 :** The change of velocity ( $V_L$ : Velocity of the leading train,  $V_{Rm}$ : Measured velocity of the rear train,  $V_{Rf}$ : Output velocity of the rear train) and acceleration ( $a_R$ ) with respect to time.

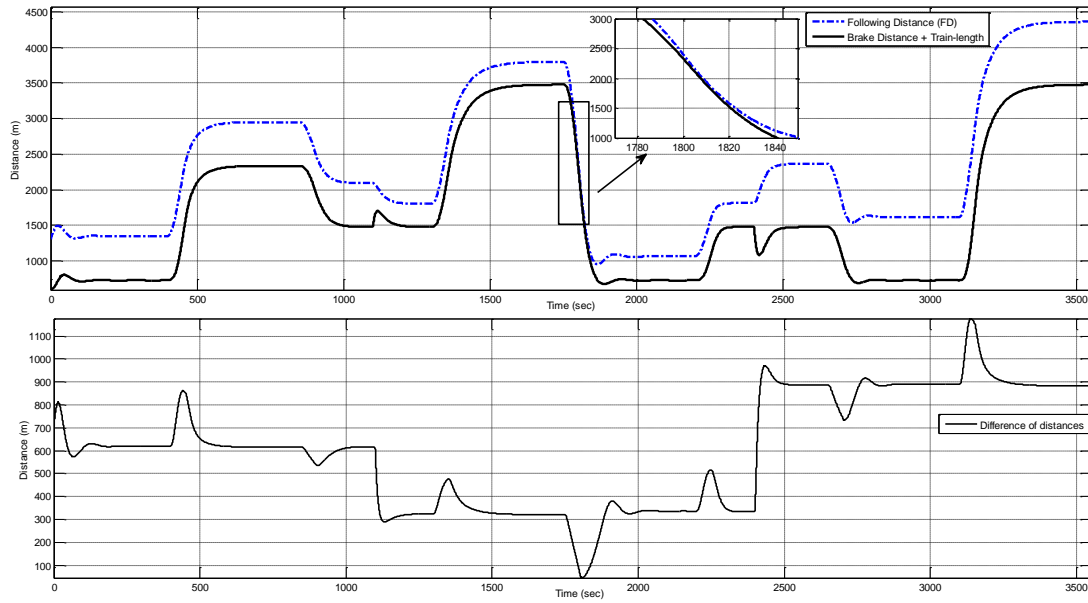


**Figure 3.13 :** The change of velocity ( $V_L$ : Velocity of the leading train,  $V_{Rm}$ : Measured velocity of the rear train,  $V_{Rr}$ : Output velocity of the rear train) and acceleration ( $a_{Rr}$ ) with respect to position.

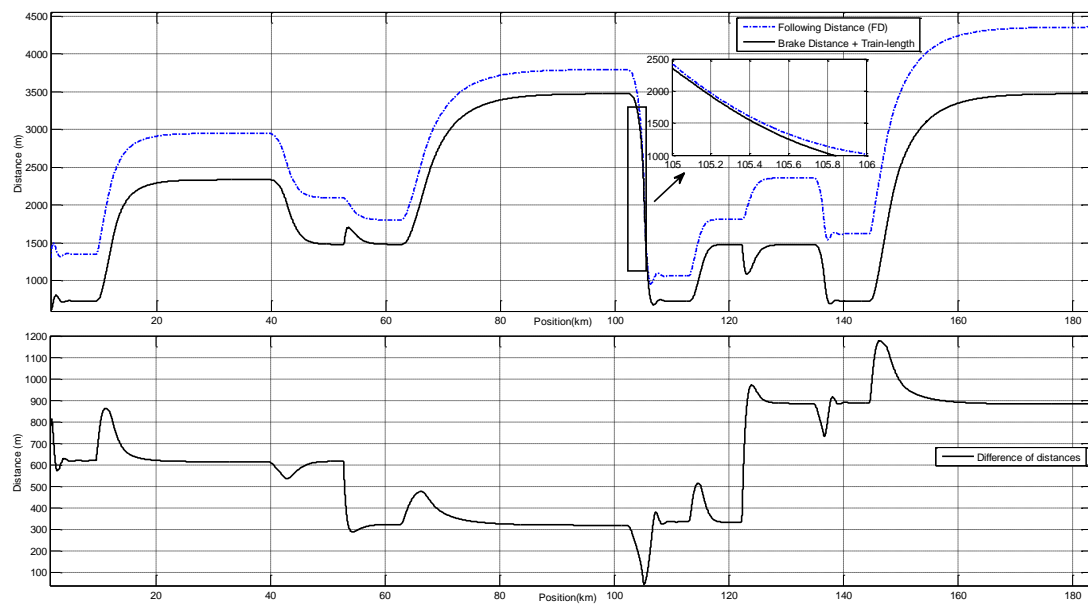


**Figure 3.14 :** The change of velocity ( $V_L$ : Velocity of the leading train,  $V_{Rm}$ : Measured velocity of the rear train,  $V_{Rr}$ : Output velocity of the rear train) and acceleration ( $a_{Rr}$ ) with respect to time.

As can be seen from the simulation results in figure 3.12, figure 3.13 and figure 3.14, the designed controller also provides a good tracking performance and the disturbances are also rejected in an acceptable short time. The acceleration of the rear train is also within the limits for a comfortable journey.



**Figure 3.15 :** The change of FD, BD and the difference between FD and BD with respect to time.



**Figure 3.16 :** The change of FD, BD and the difference between FD and BD with respect to position.

The safe following distance between trains is also kept for the noisy case. Since the noise is added to the measured velocity of the rear train, the effect of the noise is also decreased while calculating the braking distance of the rear train (please see figure 3.6).



### 3.5 Results and Discussion

European Rail Traffic Management System (ERTMS) is the combination of European Train Control System (ETCS) and GSM-R. ERTMS is a unique standard which is developed to combine different national standards in Europe. ERTMS also became a world-wide standard. The application levels of ETCS and the differences between the fixed-block signaling systems and the moving-block signaling systems are explained in this section. Calculation of train braking distance (or safe stopping distance) and its effect on safe movement of trains as moving-blocks are also mentioned in this section.

Additionally, an online adaptive PD controller design is realized for speed control of two consecutive trains. Simulation results by considering the braking distance and the dynamic train equations are also given. The robustness/success of the proposed online controller is also tested by adding measurement noise and input disturbance. The simulation results show that the proposed online controller is successful in disturbance rejection and provides robustness against the measurement noises. A disadvantage of the designed controller is the complexity of the controller structure. For instance due to the need of the inverse transform of the Jacobian matrix in the calculation of PD parameters, selecting a larger value of parameter  $K$  (the prediction horizon) leads to increase the computational complexity. This type of controller is not appropriate systems with short time constants.

A theoretical framework for proving the satisfactory performance of the proposed controller as well as designing controllers for multiple trains in the same railway line where train characteristics such as mass and gradient are assumed as uncertain parameters and the stability analysis of the proposed controller can be considered as future works.



#### **4. CONCLUSION**

Since the railway systems are regarded as safety-critical systems and its signaling system ensures the safety of travel and transportation, the design steps have to be carried out very carefully. The recommendations of the railway-related safety standards, the general safety aspects accepted by all over the world and regional safety criteria must be taken into account while designing a signaling software or namely the interlocking software. By considering the above definitions and the need for the development of railway signaling systems in Turkey, the pioneer project named as the Turkish National Railway Signaling Project in cooperation with the Turkish State Railways and TUBITAK is realized.

In this thesis, the basic concepts of the railway signaling systems that can be classified into two main topics known as fixed-block and moving-block railway signaling systems are discussed. The interlocking software development of the TRNSP and the problems encountered during this process is mentioned. A new voting strategy is proposed to solve these problems. Moreover, to provide the safe follow-up procedure in moving-block systems, an existing online adaptive controller design method is applied to control the speed of the trains.

To summarize, the investments on railways in Turkey have increased much more than in previous periods. International agreements such as deployment of European Rail Traffic Management System (ERTMS), Turkish State Railways and so development of railway systems in Turkey accelerated. These investments also lead private construction companies to work on railway signaling systems. Finally, by completion of the Marmaray tunnel project, Europe and Asia were connected together. Within this perspective, developing original, reliable and fail-safe interlocking systems, which constitute one of the most vital parts of the railway signaling systems, is of great importance for Turkey. Construction of the railways compatible with ERTMS still continues and the Turkish Government has already planned new investments for future extensions.



## REFERENCES

- [1] **URL-1** <[www.ecopassenger.org](http://www.ecopassenger.org)>, date retrieved 14.04.2014.
- [2] **IEC 61508-3**. (2010). Functional Safety of Electrical/Electronic/Programmable electronic safety-related systems, Part 3: Software requirements, *European Committee for Electrotechnical Standardization*, Brussels.
- [3] **Clark, S.** (2010). A history of railway signalling: From the bobby to the balise. In RSCS 2010. *Proceedings of the IET Professional Development Course on Railway Signalling and Control Systems*, 7-11 June.
- [4] **Hall, S.** (2001). *Modern Signalling Handbook*, Ian Allan Publishing, England.
- [5] **Kuepper, G. J.** (1999). 150 years of train-disasters - practical approaches for emergency responders. *9-1-1 Magazine*, issue. *September/October*, pp. 30-33.
- [6] **Akita, K., Watanabe, T., Nakamura, H. and Okumura, I.** (1985). Computerized Interlocking System for Railway Signalling Control: SMILE. *IEEE Transactions on Industry Applications*, IA-21-4, 826-834.
- [7] **Petersen, J. L.** (1998). Automatic Verification of Railway Interlocking Systems: A Case Study. In FMSP'98. *Proceedings of the 2nd Workshop on Formal Methods in Software Practice*, Clearwater Beach, FL, USA, 04-05 March.
- [8] **Kantz, H. and Koza C.** (1995). The ELEKTRA Railway Signalling-System: Field Experience with an Actively Replicated System with Diversity. In FTCS'25. *Proceedings of the 25th International Symposium on Fault-Tolerant Computing*, Pasadena, CA, USA, 27-30 June.
- [9] **Rao, V.P. and Venkatachalam, P.A.** (1987). Microprocessor-Based Railway Interlocking Control with Low Accident Probability. *IEEE Transactions on Vehicular Technology*, VT-353, 141-147.
- [10] **Hartonas-Garmhausen, V., Campos, S., Cimatti, A., Clarke, E. and Giunchiglia, F.** (2000). Verification of a Safety-Critical Railway Interlocking System with Real-time Constraints. *Science of Computer Programming*, 36, 53-64.
- [11] **Nakamatsu, K., Kiuchi, Y., Chen, W.Y. and Chung, S.L.** (2004). Intelligent Railway Interlocking Safety Verification Based on Annotated Logic Program and its Simulator. *Proceedings of the IEEE International Conference on Networking, Sensing&Control*, 21-23 March.
- [12] **Dipoppa, G., D'Alessandro, G., Semprini, R. and Tronci, E.** (2001). Integrating Automatic Verification of Safety Requirements in Railway Interlocking System Design. In HASE 2001. *Proceedings of the 6th*

*IEEE International Symposium on High Assurance Systems Engineering*, Boco Raton, FL, USA, 22-24 October.

- [13] **Roanes-Lozano, E., Roanes-Macias, E. and Laita, L.M.** (2000). Railway Interlocking Systems and Gröbner bases. *Mathematics and Computers in Simulation*, 51, 473-481.
- [14] **She, X., Sha, Y., Chen, Q. and Yang, J.** (2007). The Application of Graph Theory on Railway Yard Interlocking Control System. *Proceedings of the IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, 13-15 June.
- [15] **Banci, M., Fantechi, A. and Ginesi, S.** (2004). The Role of Formal Methods in Developing a Distributed Railway Interlocking System. *Proceedings of the 5th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems*.
- [16] **Banci, M., Fantechi, A. and Ginesi, S.** (2005). Some Experiences on Formal Specification of Railway Interlocking Systems using Statecharts. *Train International Workshop at SEFM2005*.
- [17] **Bohn, J., Damm, W., Klose, J., Moik, A. and Wittke, H.** (2002). Modeling and Validating Train System Applications Using Statemate and Live Sequence Charts. In IDPT 2002. *Proceedings of the Integrated Design and Process Technology Conference*, 2002.
- [18] **Hei, X., Takahashi, S. and Nakamura, H.** (2006). Distributed Interlocking System and Its Safety Verification. *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Dalian, China, 21-23 June.
- [19] **Hei, X., Takahashi, S. and Nakamura, H.** (2008). Toward Developing a Decentralized Railway Signalling System Using Petri nets. *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics*, Chengdu, China, 21-24 September.
- [20] **URL-2** <<https://www.thalesgroup.com/en/content/seltracr-cbtc-communications-based-train-control-urban-rail>>, date retrieved 14.04.2014.
- [21] **URL-3** <<http://www.railsigns.co.uk/info/ablock1/ablock1.html>>, date retrieved 14.04.2014.
- [22] **URL-4** <<http://mysite.du.edu/~jcalvert/railway/ctc.htm>>, date retrieved 14.04.2014.
- [23] **IEC 61508-4.** (2010). Functional Safety of Electrical/Electronic/Programmable electronic safety-related systems, Part 4: Definitions and abbreviations, *European Committee for Electrotechnical Standardization*, Brussels.
- [24] **Leveson, N. G. and Stolzy J. L.** (1987). Safety Analysis Using Petri Nets. *IEEE Transactions on Software Engineering*, SE-13, 386-397.
- [25] **Knight, J.C.** (2002). Safety critical systems: challenges and directions," Software Engineering. In ICSE 2002. *Proceedings of the 24rd International Conference on Software Engineering*, Orlando, FL, USA 23-25 May.

- [26] **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*, 19, 216-223.
- [27] **EN 50128.** (2001). Railway Applications, Communications, signalling and processing systems, Software for railway control and protection systems, *European Committee for Electrotechnical Standardization*, Brussels.
- [28] **IEC 61508-1.** (2010). Functional Safety of Electrical/Electronic/Programmable electronic safety-related systems, Part 1: General requirements, *European Committee for Electrotechnical Standardization*, Brussels.
- [29] **Söylemez, M.T., Durmuş, M.S. and Yıldırım, U.** (2011). Functional Safety Application on Railway Systems: Turkish National Railway Signalization Project. In COMADEM'11. *Proceedings of the 24th International Congress on Condition Monitoring and Diagnostics Engineering Management*, Stavanger, Norway, 30 May-01 June.
- [30] **IEC 61508-5.** (2010). Functional Safety of Electrical/Electronic/Programmable electronic safety-related systems, Part 5: Examples of Methods for the Determination of Safety Integrity Levels, *European Committee for Electrotechnical Standardization*, Brussels.
- [31] **Yıldırım, U., Durmuş, M. S. and Söylemez, M. T.** (in press). Automatic Generation of the Railway Interlocking Tables by Using Computer Algebra Toolbox, *Journal on Automation and Control Engineering*.
- [32] **Mutlu, İ., Yıldırım, U., Durmuş, M. S. and Söylemez, M. T.** (2013). Automatic Interlocking Table Generation for Non-Ideal Railway Yards. In ACATTA 2013. *Proceedings of the IFAC Workshop on Advances in Control and Automation Theory for Transportation Applications*, Istanbul Technical University, İstanbul, Turkey, 16-17 September.
- [33] **Latif-Shabgahi G., Bass, J. M. and Bennett, S. A.** (2004). Taxonomy for Software Voting Algorithms Used in Safety-Critical Systems. *IEEE Transactions on Reliability*, 53, 319-328.
- [34] **Lorczak, P. R., Çağlayan, A. K. and Eckhardt, D. E.** (1989). A Theoretical Investigation of Generalized Voters. In FTCS'19. *Proceedings of the 19th International Symposium on Fault-Tolerant Computing*, Chicago, IL, USA, 21-23 June.
- [35] **Gersting, J. L., Nist, R. L., Roberts, D. B. and Van Valkenburg, R. L.** (1991). A Comparison of Voting Algorithms for N-version Programming. *Proceedings of the 24th Annual Hawaii International Conference on System Sciences*, Kauai, Hawaii, 8-11 January.
- [36] **Parhami, B.** (1994). Voting Algorithms. *IEEE Transactions on Reliability*, 43, 617-629.
- [37] **Mitra, S. and McCluskey, E. J.** (2000). Word-Voter: A New Voter Design for Triple Modular Redundant Systems. *Proceedings of the 18th IEEE VLSI Test Symposium*, Montreal, Canada, 30 April-04 May.

- [38] **Latif-Shabgahi, G., Bennett, S. A. and Bass, J. M.** (2003). Smoothing Voter: A Novel Voting Algorithm for Handling Multiple Errors in Fault-Tolerant Control Systems. *Microprocessors and Microsystems*, 27, 303-313.
- [39] **Latif-Shabgahi G.** (2004). A Novel Algorithm for Weighted Average Voting Used in Fault Tolerant Computing Systems. *Microprocessors and Microsystems*, 28, 357-361.
- [40] **Latif-Shabgahi, G. and Hirst, A. J.** (2005). A Fuzzy Voting Scheme for hardware and Software Fault Tolerant Systems. *Fuzzy Sets and Systems*, 150, 579-598.
- [41] **Singamsetty, P. K. and Panchumarthy, S. R.** (2011). A Novel History Based Weighted Voting Algorithm for Safety Critical Systems. *Journal of Advances in Information Technology*, 2, 139-145.
- [42] **Von Neumann J.** (1956). *Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components*, Automata Studies, Annual of Mathematic Studies, Princeton University Press.
- [43] **Lyons, R. E. and Vanderkulk, W.** (1962). The Use of Triple-Modular Redundancy to Improve Computer Reliability. *IBM Journal of Research and Development*, 6, 200-209.
- [44] **Cooper, A. E. and Chow, W. T.** (1976). Development of On-board Space Computer Systems. *IBM Journal of Research and Development*, 20, 5-19.
- [45] **Cassandras, C. G. and Lafortune, S.** (2008). *Introduction to discrete event systems*, 2nd ed., Springer.
- [46] **Durmuş, M. S., Yıldırım, U., Eriş, O. and Söylemez, M. T.** (2011). Synchronizing Automata and Petri Net based Controllers. In ELECO'11. *Proceedings of the 7th International Conference on Electrical and Electronics Engineering*, Bursa, Turkey, 1-4 December.
- [47] **Ramadge, P. J. and Wonham, W. M.** (1989). The Control of Discrete Event Systems. *Proceedings of IEEE*, 77, 81-98.
- [48] **Murata, T.** (1989). Petri nets: Properties, Analysis and Applications. *Proceedings of IEEE*, 77, 541-580.
- [49] **Uzam, M. and Jones, A. H.** (1998). Discrete event control system design using automation Petri nets and their ladder diagram implementation. *The International Journal of Advanced Manufacturing Technology*, 14, 716-728.
- [50] **Thapa, D., Dangol, S. and Wang, G. N.** (2005). Transformation from Petri nets model to programmable logic controller using one-to-one mapping technique. In CIMCA-IAWTIC'05. *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, Vienna, Austria, 28-30 November.
- [51] **Frey, G.** (2000). Automatic implementation of Petri net based control algorithms on PLC. In ACC'00. *Proceedings of the 2000 American Control Conference*, Chicago, IL, USA, 28-30 June.



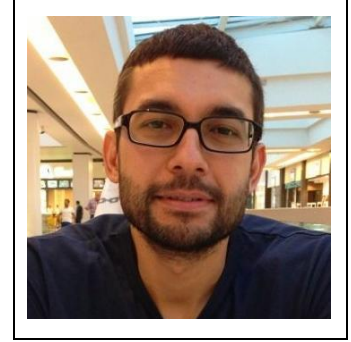
- [52] **Durmuş, M. S., Eriş, O., Yıldırım, U. and Söylemez, M. T.** (2011). A New Voting Strategy in Diverse Programming for Railway Interlocking Systems. In TMEE'11. *Proceedings of the International Conference on Transportation and Mechanical & Electrical Engineering*, Changchun, China, 16-18 December.
- [53] **Durmuş, M. S., Yıldırım, U., Eriş, O. and Söylemez, M. T.** (2012). Signalization System Design for Fixed-Block Railway Systems: The Case of Turkish National Railway Signalization Project. In CTS'12. *Proceedings of the 13th IFAC Symposium on Control in Transportation Systems*, Sofia, Bulgaria, 12-14 September.
- [54] **Mutlu, İ., Ovatman, T., Söylemez, M. T. and Gören-Sümer, L.** (2011). A New Test Environment for PLC Based Interlocking Systems. In TMEE'11. *Proceedings of the International Conference on Transportation and Mechanical & Electrical Engineering*, Changchun, China, 16-18 December.
- [55] **URL-5** <[http://www.ertms.net/?page\\_id=55](http://www.ertms.net/?page_id=55)>, date retrieved 14.04.2014.
- [56] **Lockyear, M. J.** (1996). Changing track: Moving-block railway signaling. *IEE Reviews*, 42, 21-25.
- [57] **ERA-ERTMS-015560.** (2009). ERTMS/ETCS Driver Machine Interface, Informative Specification Document, ver.2.3, *European Railway Agency*.
- [58] **URL-6** <<http://www.uic.org/spip.php?article631>>, date retrieved 14.04.2014.
- [59] **URL-7** <[http://www.uic.org/IMG/pdf/gsm-r\\_guide.pdf](http://www.uic.org/IMG/pdf/gsm-r_guide.pdf)>, date retrieved 14.04.2014.
- [60] **Barney, D., Haley, D. and Nikandros, G.** (2001). Calculating Train Braking Distance. In SCS'01. *Proceedings of the 6th Australian Workshop on Safety Critical Systems and Software*, Brisbane, Australia.
- [61] **Zimmermann, A. and Hommel, G.** (2003). A Train Control System Case Study in Model-Based Real Time System Design. In IPDPS'03. *Proceedings of the International Parallel and Distributed Processing Symposium*, 22-26 April.
- [62] **Vincze, B. and Tarnai, G.** (2006). Development and Analysis of Train Brake Curve Calculation Methods with Complex Simulation. In ELECTR'06. *Proceedings of International Exhibition of Electrical Equipment for Power Engineering, Electrical Engineering, Electronics, Energy and Resource-saving Technologies, Household Electric Appliances*, Zilina, Slovika, 23 - 24 May.
- [63] **Abed, S. K.** (2010). European Rail Traffic Management System - An Overview. *Proceedings of the 1st International Conference on Energy, Power and Control*, Basrah, Iraq, 30 November - 2 December.
- [64] **ERA-ERTMS-040026.** (2012). Introduction to ETCS Braking Curves, Information Document, ver.1.2, *European Railway Agency*.
- [65] **Patra, S.** (2007). Worst-Case Software Safety Level for Braking Distance Algorithm of a Train. *Proceedings of the 2nd International Conference on System Safety*, London, England, 22-24 October.

- [66] **Wei, S., Bai-gen, C., Jing-jing, W. and Jian, W.** (2010). Research and Analysis of ETCS Controlling Curves Model. In *ICACTE. Proceedings of the 3rd International Conference on Advances Computer Theory and Engineering*, Chengdu, China, 20-22 August.
- [67] **Booth, P. D.** (2010). Intermittent and Continuous Automatic Train Protection. In *RSCS 2012. Proceedings of the IET Professional Development Course on Railway Signalling and Control Systems*, London, England, 21-24 May.
- [68] **IEEE 1698-2009.** (2009). IEEE Guide for the Calculation of Braking Distance for Rail Transit Vehicles. *IEEE Vehicular Technology Society*.
- [69] **Pearson, L.V.** (1973). Moving Block Railway Signalling. *PhD thesis*, Loughborough University of Technology, UK.
- [70] **Ping, L. K., You, G. Z. and Hua, M. B.** (2007). Energy-Optimal Control Model for Train Movements. *Chinese Physics*, 16, 359-364.
- [71] **İplikçi, S.** (2010). A comparative study on a novel model-based PID tuning and control mechanism for nonlinear systems. *International Journal of Robust and Nonlinear Control*, 20, 1483-1501.
- [72] **Ke, B. R., Lin, C. L. and Lai, C. W.** (2011). Optimization of train-speed trajectory and control of mass rapid transit systems. *Control Engineering Practice*, 19, 675-687.
- [73] **Smola, A. J. and Scholkopf, B.** (2004). A tutorial on support vector regression. *Statistics and Computing*, Vol. 14, pp. 199-222.
- [74] **Cristianini, N. and Shawe-Taylor, J.** (2000). *An Introduction to Support Vector Machines and other Kernel based Learning Methods*, Cambridge University Press, England.
- [75] **Takao, K., Yamamoto, T. and Hinamoto, T.** (2006). A design of PID controllers with a switching structure by a support vector machine. In *IJCNN'06. Proceedings of the International Joint Conference on Neural Network*, Vancouver, Canada.
- [76] **İplikçi S.** (2009). Controlling the Experimental Three-Tank System via Support Vector Machines. *Lecture Notes in Computer Science*, 5495, 391-400.
- [77] **Zhao, X. H., Wang, G., Zhao, K. K. and Tan, D. J.** (2009). On-line least squares support vector machine algorithm in gas prediction. *Mining Science and Technology*, 19, 194-198.
- [78] **Zhu, Y. F. and Mao, Z. Y.** (2004). Online Optimal Modeling of LS-SVM based on Time Window. In *ICIT'04, Proceedings of the IEEE International Conference on Industrial Technology*, 08-10 December.
- [79] **Bobál, V., Böhm, J., Fessl, J. and Macháček, J.** (2005). *Digital Self-tuning controller: algorithms, implementation and applications*. Advanced Textbooks in Control and Signal Processing, Springer-Verlag, London Limited.
- [80] **Luenberger, D. G. and Ye, Y.** (2008). *Linear and Nonlinear Programming*. 3rd ed., Springer Science+Business Media, LLC.

- [81] **Ravindran, A., Ragsdell, K. M. and Reklaitis, G. V.** (2006). *Engineering Optimization Methods and Applications*, 2nd ed., John Wiley & Sons.



## CURRICULUM VITAE



**Name Surname:** Mustafa Seçkin Durmuş  
**Place and Date of Birth:** Antalya 27.08.1980  
**E-Mail:** msd.itu@gmail.com, msdurmus@pau.edu.tr  
**B.Sc.:** Electrical and Electronics Engineering,  
Pamukkale University  
**M.Sc.:** Institute of Science, Pamukkale University,

### Professional Experience and Rewards:

- Turkish National Railway Signalization Project, Scholar, TUBITAK Project No: 108G186 (June 2009 - March 2012).
- Ayrık Olay Sistemlerde Hata Teşhisi ve Toparlanma, ITU, Ph.D. Thesis Project, Project Number: 34549 (May 2011 - December 2013).
- Japan Society for the Promotion of Science (JSPS) Ronpaku fellowship (April 2012 - March 2015).
- Development of Signaling System for Esenler Depot, Consultant, Istanbul Transportation Company, ARII, Technocity (March 2012 - ).

### List of Publications:

#### National Conference Proceedings (In Turkish):

1. **Durmuş, M. S.**, Takai S and Söylemez, M. T. (2013). Raylı Sistem Sinyalizasyon Tasarımında Ayrık Olay Sistem Yaklaşımı ile Arıza Teşhisi. In TOK'13. *Proceedings of the Turkish National Meeting on Automatic Control*, Inonu University, Malatya, Turkey, 26-28 September.
2. **Durmuş, M. S.**, Yıldırım, U. and Söylemez, M. T. (2012). Hareketli-Blok Raylı Ulaşım Sistemlerinde Tren Hız Kontrolü. In TOK'12. *Proceedings of the Turkish National Meeting on Automatic Control*, Niğde University, Niğde, Turkey, 11-13 October.
3. **Durmuş, M. S.**, Yıldırım, U., Üstoğlu, İ., Kaymakçı, O. and Akçil, L. (2012). Kent İçi Raylı Ulaşımında Sabit-Blok Sinyalizasyon Sistemi Tasarımı: Esenler

- Depo Sahası. In TOK'12. *Proceedings of the Turkish National Meeting on Automatic Control*, Niğde University, Niğde, Turkey, 11-13 October.
4. **Durmuş, M. S.**, Yıldırım, U. and Söylemez, M. T. (2011). Demiryolu Sinyalizasyonunda Farklı Programlama Tekniği ile Otomat ve Petri Ağı Tabanlı Kontrolörlerin Senkronizasyonu. In TOK'11. *Proceedings of the Turkish National Meeting on Automatic Control*, Dokuz Eylül University, İzmir, Turkey, 14-16 September.
  5. **Durmuş, M. S.**, Yıldırım, U. and Söylemez, M. T. (2011). Demiryolu Sinyalizasyon Tasarımında Fonksiyonel Güvenlik ve Ayrık Olay Sistem Yaklaşımı. In EUSİS 2011. *Proceedings of the Electrical Transportation Systems Symposium*, Bursa-Eskişehir, Turkey, 7-9 April.
  6. Yıldırım, U., **Durmuş, M. S.** and Söylemez, M. T. (2011). Demiryolu Sinyalizasyon Sistemleri için Otomatik Anlaşman Tablosu Oluşturulması. In EUSİS 2011. *Proceedings of the Electrical Transportation Systems Symposium*, Bursa-Eskişehir, Turkey, 7-9 April.
  7. Yıldırım, U., **Durmuş, M. S.** and Söylemez, M. T. (2010). Anlaşman Tasarımı için Petri Ağı Denetçilerinin Otomatik Oluşturulması. In ELECO 2010. *Proceedings of the Conference on Electrical and Electronics Engineering*, Bursa, Turkey, 2-5 December.
  8. Yıldırım, U., **Durmuş, M. S.**, Kurşun, A. and Söylemez, M. T. (2010). Demiryolu Hemzemin Geçitleri için Hatada-Güvenli Sinyalizasyon ve Anlaşman Tasarımı. In TOK'10. *Proceedings of the Turkish National Meeting on Automatic Control*, Gebze Yüksek Teknoloji Enstitüsü, Gebze, Turkey, 21-23 September.
  9. Akın, K., **Durmuş, M. S.** and Söylemez, M. T. (2010). Demiryolu Sinyalizasyon Sistemi Bileşenlerinin Otomasyon Petri Ağları ile Modellenmesi ve PLC ile Gerçeklenmesi. In TOK'10. *Proceedings of the Turkish National Meeting on Automatic Control*, Gebze Institute of Technology, Gebze, Turkey, 21-23 September.
  10. Saygın, S., Yakın, İ., **Durmuş, M. S.** and Söylemez, M. T. (2009). Petri Ağları ile Demiryolu Makas Bölgelerinin Anlaşman ve Sinyalizasyonu Tasarımı. In TOK'09. *Proceedings of the Turkish National Meeting on Automatic Control*, Yıldız Technical University, İstanbul, Turkey, 13-16 October.
  11. **Durmuş, M. S.** and Söylemez, M. T. (2008). Petri Ağları ile Demiryolu Anlaşman ve Sinyalizasyon Tasarımı. In ELECO 2008. *Proceedings of the Conference on Electrical and Electronics Engineering*, Bursa, Turkey, 26-30 November.
  12. **Durmuş, M. S.**, Kumbasar, T., Yeşil, E., Eksin, İ. and Söylemez, M. T. (2008). İntegratör Etkili Ölü Zamanlı Süreçlerin PI-PD Kontrolü İçin Bir Bulanık Ayar Mekanizması. In TOK'08. *Proceedings of the Turkish National Meeting on*

*Automatic Control*, Istanbul Technical University, İstanbul, Turkey, 13-15 November.

13. **Durmuş, M. S.** and İplikçi, S. (2007). Veri Kümeleme Algoritmalarının Performansları Üzerine Karşılaştırmalı Bir Çalışma. In AB'07. *Proceedings of the Academic Informatics Conference*, Dumlupınar University, Kütahya, Turkey, 31 January-2 February.

#### **International Conference Proceedings:**

1. Mutlu, İ., Yıldırım, U., **Durmuş, M. S.** and Söylemez, M. T. (2013). Automatic Interlocking Table Generation for Non-Ideal Railway Yards. In ACATTA 2013. *Proceedings of the IFAC Workshop on Advances in Control and Automation Theory for Transportation Applications*, Istanbul Technical University, İstanbul, Turkey, 16-17 September.
2. **Durmuş, M. S.**, Uçak, K., Öke, G. and Söylemez, M. T. (2013). Train Speed Control in Moving-Block Railway Systems: An Online Adaptive PD Controller Design. In ACATTA 2013. *Proceedings of the IFAC Workshop on Advances in Control and Automation Theory for Transportation Applications*, Istanbul Technical University, İstanbul, Turkey, 16-17 September.
3. Okan, M. R, **Durmuş, M. S.**, Özmal, K., Akçil, L., Üstoğlu, İ. and Kaymakçı, Ö. T. (2013). Signaling System Solution for Urban Railways: Esenler Railway Depot. In ACATTA 2013. *Proceedings of the IFAC Workshop on Advances in Control and Automation Theory for Transportation Applications*, Istanbul Technical University, İstanbul, Turkey, 16-17 September.
4. Üstoğlu, Kaymakçı, Ö. T., **Durmuş, M. S.**, Yıldırım, U. and Akçil, L. (2012). Signaling System Design for Urban Transportation: The Case of İstanbul Esenler Depot. In FORMS/FORMAT 2012. *Proceedings of the 9th Symposium on Formal Methods*, Braunschweig Technical University, Braunschweig, Germany, 12-13 December.
5. **Durmuş, M. S.**, Yıldırım, U. and Söylemez, M. T. (2012). Automatic Generation of Petri Net Supervisors for Railway Interlocking Design. In AUCC 2012. *Proceedings of the 2nd IEEE Australian Control Conference*, Sydney, Australia, 15-16 November.
6. Yıldırım, U., **Durmuş, M. S.** and Söylemez, M. T. (2012). A Computer Algebra Toolbox for Finding All Stabilizing PID Controllers. In AUCC 2012. *Proceedings of the 2nd IEEE Australian Control Conference*, Sydney, Australia, 15-16 November.
7. **Durmuş, M. S.**, Yıldırım, U. and Söylemez, M. T. (2012). Interlocking System Design for ERTMS / ETCS: An Approach with Batches Petri Nets. In WODES'12. *Proceedings of the 11th IFAC International Workshop on Discrete Event Systems*, Guadalajara, Mexico, 3-5 October 2012.
8. **Durmuş, M. S.**, Yıldırım, U., Eriş, O. and Söylemez, M. T. (2012). Signalization System Design for Fixed-Block Railway Systems: The Case of

- Turkish National Railway Signalization Project. In CTS'12. *Proceedings of the 13th IFAC Symposium on Control in Transportation Systems*, Sofia, Bulgaria, 12-14 September.
9. Eriş, O., Yıldırım, U., **Durmuş, M. S.**, Söylemez, M. T. and Kurtulan S. (2012). N-version Programming for Railway Interlocking Systems: Synchronization and Voting Strategy. In CTS'12. *Proceedings of the 13th IFAC Symposium on Control in Transportation Systems*, Sofia, Bulgaria, 12-14 September.
  10. Yıldırım, U., **Durmuş, M. S.** and Söylemez, M. T. (2012). Automatic Interlocking Table Generation for Railway Stations using Symbolic Algebra. In CTS'12. *Proceedings of the 13th IFAC Symposium on Control in Transportation Systems*, Sofia, Bulgaria, 12-14 September.
  11. **Durmuş, M. S.**, Eriş, O., Yıldırım, U. and Söylemez, M. T. (2011). A New Voting Strategy in Diverse Programming for Railway Interlocking Systems. In TMEE'11. *Proceedings of the International Conference on Transportation and Mechanical & Electrical Engineering*, Changchun, China, 16-18 December.
  12. **Durmuş, M. S.**, Yıldırım, U., Eriş, O. and Söylemez, M. T. (2011). Synchronizing Automata and Petri Net based Controllers. In ELECO 2011. *Proceedings of the 7th International Conference on Electrical and Electronics Engineering*, Bursa, Turkey, 01-04 December.
  13. Söylemez, M. T., **Durmuş, M. S.**, Yıldırım, U., Türk, S. and Sonat, A. (2011). The Application of Automation Theory to Railway Signalization Systems: The Case of Turkish National Railway Signalization Project. *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 28 August-2 September.
  14. **Durmuş, M. S.**, Yıldırım, U. and Söylemez, M. T. (2011). Application of Functional Safety on Railways Part I: Modelling & Design. In ASCC'11. *Proceedings of the 8th IEEE Asian Control Conference*, Kaohsiung, Taiwan, 15-18 May.
  15. Yıldırım, U., **Durmuş, M. S.** and Söylemez, M. T. (2011). Application of Functional Safety on Railways Part II: Software Development. In ASCC'11. *Proceedings of the 8th IEEE Asian Control Conference*, Kaohsiung, Taiwan, 15-18 May.
  16. Söylemez, M. T., **Durmuş, M. S.** and Yıldırım, U. (2011). Functional Safety Application on Railway Systems: Turkish National Railway Signalization Project. In COMADEM'11. *Proceedings of the 24th International Congress on Condition Monitoring and Diagnostics Engineering Management*, Stavanger, Norway, 30 May-1 June.
  17. **Durmuş, M. S.**, Yıldırım, U. and Söylemez, M. T. (2010). Signalization and Interlocking Design for a Railway Yard: A Supervisory Control Approach by Enabling Arcs. In IMS'10. *Proceedings of the 7th International Symposium on Intelligent and Manufacturing Systems*, Sarajevo, Bosnia Herzegovina, 15-17 September.



18. Yıldırım, U., **Durmuş, M. S.** and Söylemez, M. T. (2010). Fail-Safe Signalization and Interlocking Design for a Railway Yard: An Automation Petri Net Approach. In *IMS'10. Proceedings of the 7th International Symposium on Intelligent and Manufacturing Systems*, Sarajevo, Bosnia Herzegovina, 15-17 September.
19. **Durmuş, M. S.**, Yıldırım, U., Kurşun, A. and Söylemez, M. T. (2010). Fail-Safe Signalization Design for a Railway Yard: A Level Crossing Case. In *WODES'10. Proceedings of the 10th IFAC International Workshop on Discrete Event Systems*, Berlin Technical University, Berlin, Germany, 30 August-1 September.
20. **Durmuş, M. S.**, Akin, K. and Söylemez, M. T. (2010). Supervisory Control Approach by Inhibitor Arcs for Signalization and Interlocking Design of a Railway Yard. In *INISTA'10. Proceedings of the International Symposium on INnovations in Intelligent SysTems and Applications*, Kayseri & Cappadocia, Turkey, 21-24 June.
21. **Durmuş, M. S.**, Söylemez, M. T. and Avşaroğulları, E. (2009) Coloured Automation Petri Nets Based Interlocking and Signalization Design. In *DECOM-IFAC'09. Proceedings of the 6th IFAC International Workshop on Knowledge and Technology Transfer in/to Developing Countries*, Ohridean Riviera, R. Macedonia, 26-29 September.
22. **Durmuş, M. S.** and Söylemez, M. T. (2009). Railway Signalization and Interlocking Design via Automation Petri Nets. In *ASCC'09. Proceedings of the 7th IEEE Asian Control Conference*, Hong Kong, 26-29 August.
23. **Durmuş, M. S.** and Söylemez, M. T. (2009). Automation Petri Net Based Railway Interlocking and Signalization Design. In *INISTA'10. Proceedings of the International Symposium on INnovations in Intelligent SysTems and Applications*, Karadeniz Technical University, Trabzon, Turkey, 29 June-01 July.

#### **International Refereed Journal Papers:**

1. **Durmuş, M. S.** and Takai, S. (2013). Modeling Moving-Block Railway Systems: A Generalized Batches Petri net Approach, *SICE Journal of Control, Measurement and System Integration*, Vol. 6, no. 6, 403-410.
2. Yıldırım, U., **Durmuş, M. S.** and Söylemez, M. T. (in press). Automatic Generation of the Railway Interlocking Tables by Using Computer Algebra Toolbox, *Journal on Automation and Control Engineering*.
3. **Durmuş, M. S.**, Eriş, O., Yıldırım, U. and Söylemez, M. T. (in press). A New Bitwise Voting Strategy for Safety-Critical Systems with Binary Decisions, *Turkish Journal of Electrical Engineering and Computer Sciences*.
4. **Durmuş, M. S.**, Takai, S. and Söylemez, M. T. (in press). Fault Diagnosis in Fixed-Block Railway Signaling Systems: A Discrete Event Systems Approach,

*IEEJ Transactions on Electrical and Electronic Engineering*, Vol. **9**, No. 5, 2014.

**National Refereed Journal Papers:**

1. **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*, Vol. **19**, issue. 5, 216-223.

**National Non-Refereed Journal Papers:**

1. **Durmuş, M. S.**, Yıldırım, U., Eriş, O. and Söylemez, M. T. (2012). Raylı Sistem Sinyalizasyon Tasarımı: Ulusal Demiryolu Sinyalizasyonu Projesi, *3e ELECTROTECH, Monthly Energy, Electric and Electronic Technologies Journal*, Vol. **214**, April 2012, issue. 4.
2. **Durmuş, M. S.**, Yıldırım, U., Eriş, O., Özdeş, O. and Söylemez, M. T. (2012). Sabit-Blok Demiryolu Sistemleri için Güvenli Anlaşman Yazılımı Tasarımı, *Automation Journal*, Vol. **238**, April 2012, issue. 4.