Doctoral Dissertation

# Control and Fault Diagnosis of Railway Signaling Systems: A Discrete Event Systems Approach

Mustafa Seckin Durmus

December 2014

Graduate School of Engineering,
Osaka University

Doctoral Dissertation

# Control and Fault Diagnosis of Railway Signaling Systems: A Discrete Event Systems Approach

Mustafa Seckin Durmus

December 2014

Graduate School of Engineering,
Osaka University

**Summary**

The use of railway transportation among different alternatives (e.g. road and air transportation) brings many profits such as less carbon dioxide emission and energy consumption. Although the infrastructure and the signaling costs of railways are high, they provide more environmental friendly and affordable solutions. Railway signaling systems are divided into two main categories named as fixed-block (conventional) and moving-block signaling systems. Independent of the signaling category, the vital component of railway systems which provides safe travel and transportation is the signaling system, namely, the interlocking software. Since railway signaling systems are classified as safety-critical systems due to the high risk value, the design and development steps of railway signaling systems are defined by international committees such as European Committee for Electrotechnical Standardization (CENELEC), International Union of Railways (UIC), The European Rail Industry (UNIFE), Union Industry of Signaling (UNISIG), and European Railway Agency (ERA). In addition to the railway related safety standards, the designers should consider the requirements and safety rules of the country where the signaling system is to be applied.

After the determination of the software requirements (both world-wide and country-based safety rules), the designer should choose appropriate modeling methods, combination of software architectures, and test procedures to achieve the required Safety Integrity Level (SIL). SIL is a discrete level for specifying the safety integrity requirements of the safety functions allocated to the Electrical, Electronic, or Programmable Electronic (E/E/PE) safety-related systems.

In this thesis, railway signaling systems are studied from the discrete event systems (DESs) point of view since railway signaling systems can be regarded as DESs because of having features like non-determinism, asynchronism, event-driven, and simultaneity. The main reason for using the DES modeling tools such as automata and Petri nets in railway signaling systems is to model the specifications of the system and to evaluate the operational requirements by analysis and re-design.

First, fault diagnosis in fixed-block railway signaling systems is studied. Detecting a fault is a critical and stringent task in railway signaling systems. The signaling system components are modeled by Petri nets and a diagnoser is designed to show diagnosability of the system.

Next, to satisfy the safety requirements of the railway related functional safety standards, a signaling system architecture which consists of two controllers and a coordinator for a fixed-block railway signaling system is studied. Based on the Petri net models of railway field components, decision making strategies including fault diagnosis are developed.

Instead of fixed-block signaling systems, moving-block signaling systems are in use to increase the transport capacity by reducing headways on railway lines. As a final study, speed control of two consecutive trains as moving-block is realized in two levels: the modeling level and the control level. The aim of this final study is to provide safe travel of trains in moving-block signaling systems. The generalized batches Petri nets approach is used for modeling the system to cope with both discrete and continuous behavior of the moving-block signaling systems and a fuzzy logic control method is proposed at the control level.

**Acknowledgements**

I wish to thank the following people because without their help and support this Ph.D. thesis would not have been possible.

Firstly, I wish to express my sincere gratitude and my sincere regards to my supervisor Professor Shigemasa Takai for his suggestions, encouragements, support and guidance in approaching to different challenges during the progress in this thesis. The discussions and cooperative studies with Professor Takai guided me to acquire valuable insight and perspective for solving the problems during my thesis studies. I wish also to thank Professor Takai for his kind hospitality, generosity and helpfulness during the times I spent in Osaka.

I wish to thank Associate Professor Toshiyuki Miyamoto for his kind hospitality, helpfulness and generosity during the times I spent in Osaka.

I present my kind regards to Professor Tetsuzo Tanino in the Division of Electrical, Electronic and Information Engineering at Osaka University, for his attentive review of my thesis and his many valuable comments.

I wish to express my appreciation to Professor Toshifumi Ise and Professor Tsuyoshi Funaki in the Division of Electrical, Electronic and Information Engineering, and to Professor Hiroyuki Shiraga in Institute of Laser Engineering at Osaka University, for serving as members of my thesis dissertation committee.

I wish to thank Assistant Professor Naoki Hayashi for his kind hospitality and generosity during the times I spent in Osaka. Especially, I wish to thank Assistant Professor Hayashi for his helpfulness and his patience in the face of my questions.

I am also grateful to all the members of the Takai Laboratory for their friendship and help during the times I spent in Osaka. Specially, I am thankful to Ms. Kiyo Nakano for her guidance in the completion of academic procedures during my research in Osaka.

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

**Contents**

**Page**

# 1. Introduction

## 1.1 Background

The use of railway transportation among different alternatives (e.g. road and air transportation) brings many profits such as less carbon dioxide emission and energy consumption. Although the infrastructure and the signaling costs of railways are rather high, they provide more environmental friendly and affordable solutions.

Railway systems can be grouped as fixed-block railway systems and moving-block railway systems from the structural point of view. In fixed-block railway systems, railway lines are divided into blocks with fixed-length and trains are moving according to the route reservation procedure whereas in moving-block railway systems, each train is regarded as a moving-block and more than one train occupancy is allowed in the same railway block.

Although there are many infrastructure and superstructure components in railways, the main component that provides safe travel and transportation is the signaling system, in other words, the interlocking system. As the speed and the density of railways are increasing day by day, the need of reliable and safe signaling systems in railways is much more today.

To provide safety in railways and fulfill the railway related functional safety requirements, railway people and committees formed international standards. For example, for fixed-block railway signaling systems, the EN 50126 standard describes the functional safety requirements related with all kinds of railway applications where Reliability, Availability, Maintenance and Safety analysis (RAMS) is determined. The EN 50128 (similar to the EN 61508-3) determines methodologies for building software for railway control applications and the EN 50129 (similar to the EN 61508-2) defines requirements for hardware of electric, electronic, and programmable devices used in railways. In addition to these European standards for fixed-block railway signaling systems, EIRENE (European Integrated Railway Radio Enhanced Network) and GSM-R (GSM for Railways) specifications are formed by

the UIC (International Union of Railways) and the ERA (European Railway Agency). ERTMS (European Rail Traffic Management System) is the combination of European Train Control System (ETCS) and GSM-R. ERTMS is a unified standard that combines different European standards for both fixed and moving-block railway systems.

In addition to the requirements and recommendations of railway related safety standards, signaling system engineers should take fault diagnosis into account while developing the signaling system software, namely, interlocking software. From the safety-related standards point of view, fault diagnosis is regarded as the activity of checking whether a system is in a faulty state, and it should be performed at the smallest subsystem level to prohibit the effect of incorrect results [1]. Especially for large and complex systems, diagnosis of faults becomes a critical and stringent task. Diagnosability analysis for fixed-block railway signaling systems can be considered as an intermediate step between modeling the system and testing the developed software. This intermediate step can be seen as a time-consuming and stringent task for signaling system software developers but it determines whether the developed system models are diagnosable or not before testing the signaling system software. Another benefit of this intermediate step is to combine the theoretical background and the practical background of signaling system engineers.

Although there are various design methods, safety precautions, and recommendations of the railway-related safety standards, sometimes the occurrence of accidents cannot be prevented. Safety Integrity Level (SIL) is a discrete level for specifying the safety integrity requirements of the safety functions allocated to the Electrical, Electronic, or Programmable Electronic (E/E/PE) safety-related systems. According to the software design steps mentioned in the V-model in [2], designers should choose approved combinations of software architectures such as defensive programming, diverse programming, and failure assertion programming architectures from table A.3 of [3] to achieve the required SIL which should be at least at 3 for railway applications [4]. The purpose of defensive programming is to consider the worst case from all input and response to it in a predetermined and plausible way. Input variables and the effect of output variables should be checked, the coding standards should be used, and the code should be as simple as possible [5]. The main purpose of failure assertion programming is to detect software design faults while

executing a program and continue the operation for high reliability [1]. The main aim of diverse programming is to develop *N* different program versions for the given input-output specifications. These different versions should be developed by different workgroups so that they do not fail at the same time because of the same reason. These different versions are combined together under a coordinator (namely, a voter) where their responses are subjected to a voting operation. Diverse programming does not overcome possible software design faults but this method enables us to handle unpredictable and unknown design faults, prevents the system, and provides the continuity of the system operation in a safe way [1].

Detailed definitions of different voting strategies can be found in [6]-[10]. If the system is not fail-safe (where the safe-state of the system is not predetermined), then generalized voting strategies can be used and generally *N* is chosen as 3 [11]-[13]. By contrast, as mentioned in section B.17 of [3], if the system has a safe-state (or the system is fail-safe), then it is feasible to demand complete agreement, in other words, complete agreement of the program versions can be sought before getting into an unsafe state. In this case, typically *N* is chosen as 2 [3], [10], [14], [15]. In [10], according to the recommendations of the railway-related safety standards, an interlocking system architecture which consists of two controllers (called modules in [10]) and a coordinator (called a voter in [10]) was proposed for a fixed-block railway signaling system, and certain synchronization problems between controllers were addressed.

Instead of dividing railway lines into blocks with fixed length, trains are regarded as moving blocks in railway lines in moving-block signaling systems [16]. A moving-block is considered as the sum of the length of the train and the safe following distance between trains. Moving-block signaling systems provide more efficient use of railway lines by enabling multiple train movements on the same block, especially on metro and urban lines. Moreover, moving-block signaling systems increase the transport capacity and reduce headways.

From the infrastructure point of view, renovation of old railway lines in Turkey has an increasing trend in past few years with the government investments for railways. From the signaling system point of view, developing signaling systems for unsignaled railway lines and implementation of ERTMS on new high speed

railways by Turkish State Railways (TCDD), and research and development activities of private companies such as Istanbul Ulasim A.S. continue today.

### 1.2 Discrete Event Systems Approach

Railway systems are regarded as discrete event systems (DESs) because of having features like non-determinism, asynchronism, event-driven, and simultaneity [17]. Representation of such a system with a model is necessary as in the conventional control theory. Modeling tools for DESs must be suitable to cover all their different features. Several methods were introduced as DES modeling tools like Grafcet [18], automata [19], and Petri nets [20]. For more details about DES theory, the reader is referred to [17]. The main aim to use the DES modeling tools in railway signaling systems is to model the specifications of the system and to evaluate the operational requirements by analysis and re-design [21]. In fact, the use of these DES modeling tools is highly recommended in table B.5 of [2] and in table A.16 of [3] as a modeling technique to design a SIL3 or SIL4 safety-critical software.

For instance, the use of Petri nets as a modeling tool in control of several transportation alternatives such as urban traffic [22] and railway systems [23] can be found in the literature. Additionally, the use of colored Petri nets with object-oriented programming [24] and a Petri net modeling technique with a supervisory control scheme [25] can be also found in the literature. However, in these studies, faulty conditions are not included in the Petri net models and a fault diagnosis approach is not considered.

Events in DESs can be classified as observable and unobservable events. A DES is said to be diagnosable if it is possible to detect, with a finite delay, occurrences of certain unobservable events which are referred to failure events [26]. The diagnoser is built from the system model itself and performs diagnostics when it observes online the behavior of the system. States of the diagnoser carry failure information, and occurrences of failures can be detected with a finite delay by inspecting these states [27].

The pioneer study on failure diagnosis of DESs is the work of Sampath et al. [26], [27] which is an automata-based approach. They gave the definition of diagnosability and presented a necessary and sufficient condition for the system to be

diagnosable. They also proposed a diagnoser design method for an experimental simple HVAC system. In [28], they developed a diagnoser design procedure for active diagnosis of DESs that presents an integrated approach to control and diagnosis. As an alternative to automata-based modeling, Ushio et al. proposed a diagnoser for a system modeled by a Petri net where only the marking of some of the places, called observable places, is observable [21]. Chung [29] extended the work of [21] by assuming that some of the transitions are also observable in addition to observable places. In [30], an approach to test diagnosability by checking the structure property of the diagnoser was proposed based on the method of [21]. More topics and approaches on diagnosis of Petri nets can be found in [31]-[34].

### 1.3 Contributions and Structure of the Thesis

In this thesis, both fixed-block and moving-block railway systems are studied using their DES models.

The first objective of this thesis is to examine the fault diagnosis scheme by using the DESs approach and apply to fixed-block railway signaling systems. To perform fault diagnosis for fixed-block railway signaling systems, the operational behavior of the railway field components are modeled by Petri nets. Then, a diagnoser is designed to show diagnosability of the system. In [35], a time Petri net modeling technique with an online monitoring approach to estimate and monitor the train movement in a small model railway layout was examined, and certain sufficient conditions for diagnosability which take temporal information into account were obtained. As possible faults, point machine faults were considered there. In this thesis, in addition to point machine faults, the route reservation procedure and faulty conditions in wayside signals are considered, and the diagnosability property is verified based on the necessary and sufficient condition of [26] in the untimed setting.

Next, according to the recommendations of the railway-related safety standards, an interlocking system architecture [10] which consists of two controllers and a coordinator for a fixed-block railway signaling system is studied. The use of Petri nets and the combination of defensive programming, diverse programming, and failure assertion programming architectures from table A.3 of [3] is chosen as recommended to develop SIL3 software. Based on the Petri net models of railway

field components, decision making strategies of the controllers and the coordinator including fault diagnosis are developed.

Moreover, for speed control of two consecutive trains in moving-block railway systems, a two level control scheme is proposed. In the first level, a hybrid technique using a generalized batches Petri nets approach with controllable batch speed [36] is slightly modified and used for modeling the system. In the second level, two fuzzy PD controllers are designed to control velocity and acceleration of the following train. The first level can be considered as the traffic control center where the train movements are monitored. The second level can be considered as the train on-board computer. Simulation results are shown in order to demonstrate the accuracy of the proposed approach.

The structure of this thesis is as follows: Basic definitions of fixed-block signaling systems and moving-block signaling systems are explained in Chapter 2. In Chapter 3, a fault diagnosis approach based on Petri net models is explained and a case study is given. A control architecture including two controllers and a coordinator is explained, and their decision making strategies are developed in Chapter 4. In Chapter 5, modeling and speed control of moving-block signaling systems are studied. The thesis ends with a conclusion in Chapter 6.

The works of the thesis are published as journal publications in [37], [38] and [39].

## 2. Railway Signaling Systems

Railway signaling systems are mainly divided into two groups on a railway block basis. In fixed-block signaling systems, namely, conventional railway signaling systems, railway lines are partitioned into blocks with fixed length, and in moving-block signaling systems, the sum of the length of train and its braking distance is considered as a moving block. In this chapter, basic definitions of fixed-block signaling systems and moving-block signaling systems are explained.

### 2.1 Fixed-Block Signaling Systems

In fixed-block railway signaling systems, railway lines are divided into fixed-length subsections, named as railway blocks. The length of a railway block is determined according to different variables such as the permitted line speed and the gradient of the railway line. Each block has entrance and exit signals with different types depending on the location of the signal [40]. Dispatchers (responsible officers) request routes for incoming and outgoing trains in the region of their responsibility.

These requests are evaluated by the interlocking system, and are accepted if all safety conditions are satisfied or rejected if at least one safety condition is not met [41]. In order to prohibit collisions, only one train is allowed in each railway block at a time. Since the occupation of the next block is indicated by the wayside signals, train drivers have to pay attention to the signals on their way of movement. Even though conventional railway systems have several drawbacks such as the reduction in railway line capacity and the same safe braking distances for all kinds of trains, they are in use since the mid 1800's. First railway systems did not need any signaling system due to low traffic and density. 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 of the obstruction in front of their way [42]. Later, by the increment of railway traffic, many accidents occurred because of either railway guards, train drivers, or component malfunctions. In order to overcome all

these problems, the first interlocking system installation was built in UK in 1843 [40].

In addition to mechanical interlocking systems where the railway traffic operations were realized by signalboxes manually and semaphores which are the earliest forms of mechanical signals [43], electronic interlocking systems such as SMILE [44], STERNOL [45], ELEKTRA [46], and other microprocessor-based systems [47] were used. An early North American railway signaling system named as Centralized Traffic Control (CTC) system was first installed in 1937 in Colorado [48]. The Drucktasten-Relaisstellwerk Siemens (DRS), namely, the pushbutton relay system can be regarded as an early version of the CTC systems, which was in use since mid-1950 by the Turkish State Railways (TCDD) [49]. Even though the name of a signaling system varies from country to country, the basic principles are almost the same. For instance, the basic principles of the British Absolute Block Signaling (ABS) [50], the basic principles the North American CTC, and the DRS are very similar to each other.

Today, the need for reliable and safe signaling systems is much more than in the past because of high train speeds and traffic density. Ensuring the system safety at all times is the most important issue in railway systems where small failures may result in a large number of casualties and property loss.

### 2.1.1 Components of the Fixed-Block Signaling Systems

Similar to [40], [51], a brief description of the components of fixed-block signaling systems is given in this chapter.

#### 2.1.1.1 Traffic Control Center

The Traffic Control Center (TCC) is responsible for the whole train traffic in its region. The components of the railway field can be controlled and monitored by the TCC. The dispatchers manage all operations including train movements.

#### 2.1.1.2 Signaling System Control Software (Interlocking System)

The signaling system control software, namely, the interlocking system (IS) evaluates the requests of the dispatchers and sends proper commands to the railway

field equipment, if necessary. The most important task of the signaling system control software is to provide the system safety at all times.

#### 2.1.1.3 Signals

Since every country has its own signaling principles and safety standards, the use of colors of railway signals and their combinations may vary from country to country. Railway signals inform train drivers of the occupation of the next block. The train drivers have to pay attention to the signals on the right side with respect to their direction of movement. For example, in the TCDD, the meaning of the red color is that the next block is occupied, whereas the yellow color means that the next block is free but not after the next block. The yellow color also permits a train to proceed with reduced speed. The green color indicates that the next two blocks are free and the train can proceed. The Japanese Railways uses red, yellow, and green signals with their combinations and the North American Railways uses purple and amber signals. Signals are generally located at the entrance and exit of railway blocks.

#### 2.1.1.4 Point Machines (Points, Railway Switches)

Point machines (PMs) enable the railway vehicles to change the track one to another. They are established in certain locations where track change is needed. They have two location indicators known as normal and reverse. The positions of the PM can be controlled by the TCC either manually or automatically. PMs can also be controlled by the officers in the railway field by using a metal bar (lever).

#### 2.1.1.5 Railway Blocks

The occupation of a train in a railway block is detected by the help of track circuits or axle counters [52]. Depending on the length of the block, one or more track circuits are used. Track circuits operate according to the short-circuit principle. By the entrance of a train into a railway block, the track circuit is short-circuited by the axles of the train. In this situation, the interlocking system considers that the related block is occupied. TCDD uses three different types of track circuits, namely, DC-type, AC-type, and Jointless-type track circuits [40]. On the other hand, 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 count the

train axles. The railway block is assumed to be occupied until the total number of the incoming axles becomes equal to the total number of the outgoing axles.

A general block diagram of the whole system is given in Figure 2.1.

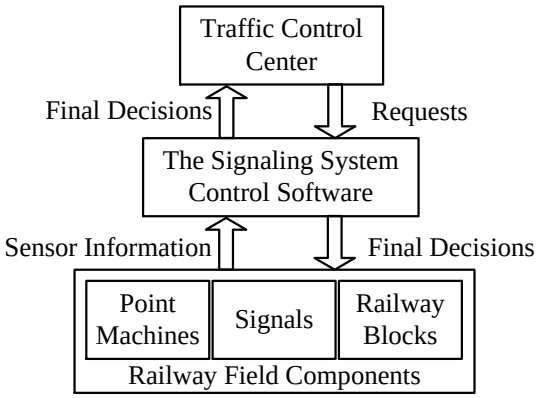


**Figure 2.1 :** General block diagram of the fixed-block signaling system.

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

Failure is defined as termination of the ability of a functional unit to provide a required function or operation of a functional unit in any way other than as required, whereas fault is defined as abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function [53]. It is important to note that the definitions of fault and failure are slightly different in the safety standards but in this thesis both of them are used to mean that the system does not work as desired. A safety-critical system is defined as a system where human safety is dependent upon the correct operation of the system. Also, a system is said to be safety-critical in [54] if the failure of a system could lead to results that are determined to be undesirable. Based on these definitions, air traffic control systems, nuclear power reactor control systems, and railway signaling systems can be classified as safety-critical systems because sometimes possible failures may lead to death of many people [55].

In this context, development of railway signaling systems has been guided by the railway-related safety standards. The umbrella standard IEC 61508 defines the functional safety requirements of all kinds of Electrical/Electronic/Programmable Electronic (E/E/PE) devices. Moreover, EN 50126 standard describes the functional safety requirements related with all kinds of railway applications where Reliability,

Availability, Maintenance and Safety analysis (RAMS) is determined. EN 50128 standard (similar to EN 61508-3) determines methodologies for building software for railway control applications and EN 50129 standard (similar to EN 61508-2) defines requirements for hardware of E/E/PE devices [4].

The designers should consider the appropriate methods and techniques from the railway related safety standards according to the correct SIL. The term SIL is a discrete level for specifying the safety integrity requirements of the safety functions allocated to the E/E/PE safety-related systems [53]. The SIL definition is made for two categories as Software SIL and System SIL. Software SIL is a classification number that determines the techniques that have to be applied to reduce software faults to an appropriate level and System SIL is a classification number that determines the required rate of confidence [3]. For instance, for a SIL 3 system in high demand mode of operation or continuous mode of operation [53], the average frequency of a dangerous failure of the safety function per hour (failure rate - $\lambda$) is between $10^{-8}$ and $10^{-7}$ [56]. 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 and 10000 years without falling into a hazardous state. The required SIL for a given system can be determined by using figure E.1, figure E.2, and table E.1 in EN 61508-5 [57].

The software development lifecycle (the V-model) is defined in [2] for the guidance in the software development process for safety-critical systems. The V-model is given in Figure 2.2.

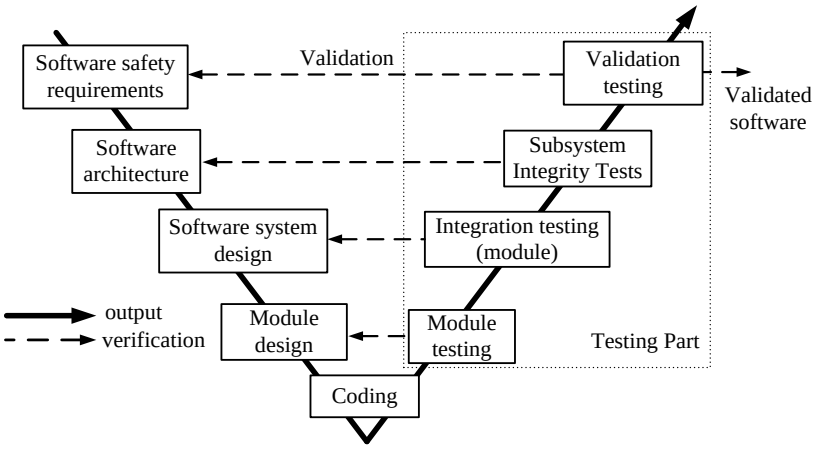


**Figure 2.2 :** The V-model.

It is obvious from the initial step of the V-model that the safety requirements of the software and the required SIL of the software have to be determined. These requirements are determined by the combination of the international safety requirements such as only one train is allowed in a railway block and the safety requirements determined by the competent authorities. In the second step, combination of several software architectures recommended in table A.3 in [3] should be chosen in order to provide the determined SIL in the first step. Later, related system sub models (modules) have to be obtained. Suitable methods for system modeling according to the determined SIL can be found in table A.16 in [3]. After modeling the required components of the safety-critical system, the obtained models have to be transformed into code snippets (or software function blocks). At last, the developed software have to be tested for verification, validation, and commissioning. From the engineering point of view, to cope with the requirements of the railway related safety standards and to achieve the desired SIL, railway signaling engineers have to pay more attention to both signaling software development and signaling software testing when designing the signaling software for fixed-block railways [2], [3], [58], [59].

### 2.2 Moving-Block Signaling Systems

As mentioned previously, trains are moving according to a route reservation procedure in fixed-block signaling systems. Trains cannot enter the same railway line in opposite directions and must leave at least one block while moving on the same railway line in the same direction. Briefly, for each block, at most one train is allowed to move. Since trains need a long stopping distance that depends on different variables such as mass of train, brake reaction time, or type of brakes etc., the length of the blocks have to be determined carefully. As it is obvious from the above, it is not possible to use the overall capacity of the railway lines efficiently [40].

Moving-block signaling systems [16] provide more efficient use of the railway lines by enabling multiple train movements on the same block, especially on metro and urban lines. Moreover, the moving-block signaling system increases the transport capacity and reduces headways. A moving block is defined as the sum of the length of the train and the safe braking distance. ERTMS application level 3 [60] and Communication Based Train Control (CBTC) [61] systems are examples of

moving-block signaling systems and already in use, in different regions in world-wide. Unlike fixed-block signaling systems, track circuits and wayside signals are removed from the railway lines. As a result of this, the total maintenance costs of railway lines are significantly decreased.

ERTMS can be considered as a standard for safety signaling and communication systems 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 [62].

European Train Control System (ETCS) has mainly three application levels from 1 to 3. The application levels 1 and 2 can be regarded as fixed-block signaling systems with ATS (Automatic Train Stop) and ATP (Automatic Train Protection) features, respectively [42] whereas the application level 3 is considered as moving-block signaling systems [63]. Detailed explanations can be found in the following subchapters.

### 2.2.1 European Train Control System

The basic of ETCS was defined 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 levels are defined below in detail.

#### 2.2.1.1 Application Level 0

In this application level, train drivers should obey the national rules and requirements. It is assumes as level 0 when an ETCS equipped vehicle is used on a route without ETCS equipment.

#### 2.2.1.2 Application Level 1

In this application level, wayside signals and track circuits are used to inform train drivers of the occupation of the track in front of them. The communication between the train and the railway block (railway track) is realized over balises

(Eurobalise®) or beacons [64]. The on-board train computer named Eurocab® receives the 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 Driver Machine Interface (DMI) [65]. Track circuits are used to detect the occupation in railway blocks. Trains cannot pass the balise as long as the next signal is red. If the train passes the related balise while the related signal is red, then it will stop automatically by the Eurocab®, or if the driver does not react in time for a signal change then the train will slow down by its own.

#### 2.2.1.3 Application Level 2

In this application level, MA is sent to the on-board train computer directly from Radio Block Center (RBC) via GSM-R instead of balises. There is no need for wayside signals and Eurocab® is always up to date over GSM-R. Balises are used as position markers and send fixed messages such as location and gradient.

#### 2.2.1.4 Application Level 3

In this application level, all necessary information from the control center to a train is sent directly to on-board train computers over GSM-R and vice versa whereas CBTC uses the bidirectional radio frequency [61]. The location of a train is detected by the help of balises placed on proper positions on the railway line. Balises provide information to a train to check the actual train location and to calibrate its odometer. It is mentioned in [66] that the proper balise position also reduces train headways and corrects speed errors.

For this application level, while moving on a railway line, depending on the conditions, End of Authority (EOA) messages could be received by the train from the control center and new MA will be uploaded to train on-board computers via GSM-R. The control center and the interlocking system communicate with the GSM-R network by using the nearest RBC. As mentioned before, more than one train can share the same block while moving on the same railway line in the same direction but trains have to leave a sufficient gap between them to prevent from collision. This gap is calculated by considering the braking distances and the safety distance which

can be chosen as the length of the train. The movement of trains is illustrated in Figure 2.3.

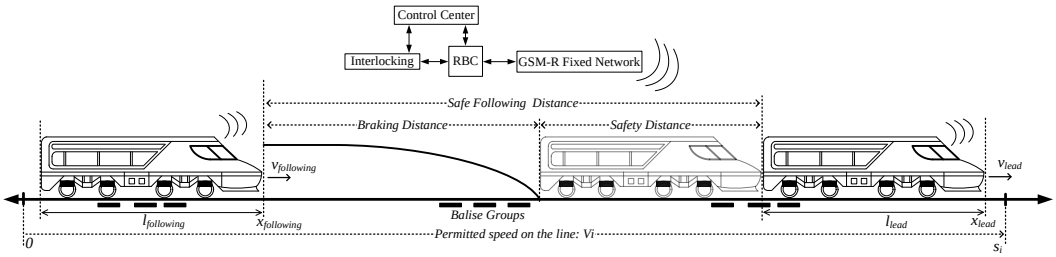


**Figure 2.3 :** Movement of trains in a railway line.

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

GSM-R [67] standard combines all past experiences and key functions from systems that were used previously in Europe. GSM-R enables communication between RBC and trains without any data loss up to very high speed (500km/h). GSM-R is mainly based on European Integrated Railway Radio Enhanced Network (EIRENE) and Mobile Radio for Railway Networks in Europe (MORANE) specifications determined by UIC [68]. GSM-R network and the communication architecture are given in Figure 2.4 [69].

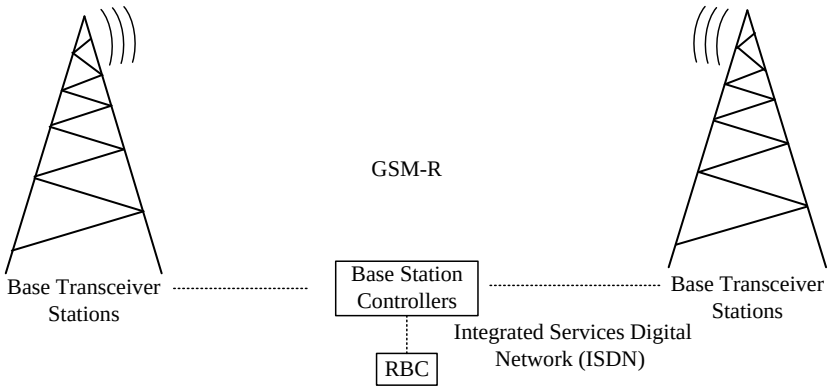


**Figure 2.4 :** GSM-R communication.

### 2.2.3 Train Braking Distance Calculation

In order to avoid train collisions in moving-block systems, trains have to leave enough distance (namely, safe stopping distance or safe braking distance) while following each other. In fixed-block signaling systems, the length of a railway block is fixed and the same braking distance is used for all kinds of trains. While calculating the braking distance in moving-block systems, the factors including the speed of the train when brakes are applied, the brake delay time, the railway track

gradient, the mass distribution of the train etc. have to be considered [70]. An example of train braking distance calculation is shown in [71] for a German high-speed train (ICE), which is 410m long with the 300km/h maximum speed, and it is found as 4000m. For high-speed trains (HST), the braking distance is calculated as 7179m in [72]. So, for a 320m long HST with the maximum speed 300km/h, the safe following distance is calculated as the sum of the train length and the braking distance which is approximately 7500m.

The braking curves are also updated depending on the train speed and MAs. MA is first uploaded to Eurocab© 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. The Eurocab© always keeps the maximum allowed speed limit by communicating with the lineside equipment, interlocking system etc. [69].

Every railway line has a permitted speed limit because of operational or environmental conditions. In case of violation of the permitted speed limit, the on-board computer activates service or emergency brakes to keep the speed of the train below the permitted speed limit [70]. 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 the train's speed is decreased to the permitted speed limit. If the 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 decreased to 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 decreased to the permitted speed limit. This prevention is used when the service brake is not available or the train passes the EOA [73]. In this situation, the train has to remain at a standstill until a new MA is available. Many studies on braking distance calculations can be found in the literature [74]-[77].

## 3. Fault Diagnosis in Fixed-Block Signaling Systems

In this chapter, fault diagnosis in fixed-block railway signaling systems is studied from the DESs point of view. First, the signaling system components are modeled by Petri nets and next a diagnoser is designed to show diagnosability of the system. Briefly, the main aim to use the DES modeling tools such as Petri nets in fixed-block railway signaling systems is to model the specifications of the system and to evaluate the operational requirements by analysis and re-design [1].

### 3.1 Petri Nets

A Petri net [20] is defined as

$$PN = (P, T, F, W, M_0), \tag{3.1}$$

where

- $P = \{p_1, p_2, ..., p_k\}$ is the finite set of places,
- $T = \{t_1, t_2, ..., t_z\}$ is the finite set of transitions,
- $F \subseteq (P \times T) \cup (T \times P)$ is the set of arcs,
- $W : F \to \{1, 2, 3, ...\}$ is the weight function,
- $M_0 : P \to \{0, 1, 2, 3, ...\}$ is the initial marking,
- $P \cap T = \varnothing$ and $P \cup T \neq \varnothing$.

We use $I(t_j)$ and $O(t_j)$ to represent the sets of input places and output places of transition $t_j$, respectively, as

$$I(t_j) = \{p_i \in P : (p_i, t_j) \in F\}, \tag{3.2}$$

$$O(t_j) = \{p_i \in P : (t_j, p_i) \in F\}. \tag{3.3}$$

For a marking $M : P \to \{0,1,\ 2,\ 3,\ ...\}$, $M(p_i) = n$ means that the *i*th place has *n* tokens [20]. A marking *M* can also be represented by a vector with *k* elements where *k* is the total number of places.

**Definition 3.1 [17]:** A transition $t_j$ is said to be enabled at a marking *M* if each input place $p_i$ of $t_j$ has at least $W(p_i, t_j)$ tokens, where $W(p_i, t_j)$ is the weight of the arc from place $p_i$ to transition $t_j$, that is, $M(p_i) \geq W(p_i, t_j)$ for all $p_i \in I(t_j)$.

Note that if $I(t_j) = \varnothing$, transition $t_j$ is always enabled. An enabled transition may or may not fire (depending on whether or not the event actually takes place). The firing of an enabled transition $t_j$ removes $W(p_i, t_j)$ tokens from each $p_i \in I(t_j)$ and adds $W(t_j, p_i)$ tokens to each $p_i \in O(t_j)$, where $W(t_j, p_i)$ is the weight of the arc from $t_j$ to $p_i$. That is,

$$M'(p_i) = M(p_i) - W(p_i, t_j) + W(t_j, p_i), \qquad \textbf{(3.4)}$$

where $M'(p_i)$ is the number of tokens in the *i*th place after the firing of transition $t_j$, and we let $W(p_i, t_j) = 0$ if $(p_i, t_j) \notin F$ and $W(p_i, t_j) = 0$ if $(t_j, p_i) \notin F$. The notation $M[t_j >$ denotes that a transition $t_j$ is enabled at a marking *M*. Also, $M[t_j > M'$ denotes that after the firing of $t_j$ at *M*, the resulting marking is $M'$. These notations can be extended to a sequence of transitions.

**Definition 3.2 [20]:** A Petri net *PN* is said to be *pure* if it has no self-loops and said to be *ordinary* if all of its arc weights are 1.

**Definition 3.3 [20]:** A marking $M_n$ is reachable from the initial marking $M_0$ in a Petri net *PN* if there exists a sequence of transitions $t_1 t_2 \ldots t_n$ such that $M_0[t_1 > M_1[t_2 > \ldots M_{n-1}[t_n > M_n$ and $R(M_0)$ denotes the set of all reachable markings from $M_0$.

**Definition 3.4 [20]:** 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, $\forall M_k \in R(M_0),\ \forall p_i \in P : M_k(p_i) \le m$. Additionally, *PN* is *safe* if it is 1-bounded.

**Definition 3.5 [20], [78]:** A Petri net *PN* is said to be *deadlock-free* (complete absence of deadlocks) if at least one transition is enabled at every reachable marking $M_k \in R(M_0)$.

The set *P* of places is partitioned into the set $P_o$ of observable places and the set $P_{uo}$ of unobservable places [21]. Similarly, the set *T* of transitions is partitioned into the set $T_o$ of observable transitions and the set $T_{uo}$ of unobservable transitions. That is,

$$P = P_o \cup P_{uo} \text{ and } P_o \cap P_{uo} = \varnothing, \tag{3.5}$$

$$T = T_o \cup T_{uo} \text{ and } T_o \cap T_{uo} = \varnothing. \tag{3.6}$$

Also, a subset $T_F$ of $T_{uo}$ represents the set of faulty transitions. It is assumed that there are *n* different failure types and $\Delta_F = \{F_1, F_2, \ldots, F_n\}$ is the set of failure types. That is,

$$T_F = T_{F_1} \cup T_{F_2} \cup \cdots \cup T_{F_n}, \tag{3.7}$$

where $T_{F_i} \cap T_{F_j} = \varnothing$ if $i \neq j$. The label set is defined as $\Delta = \{N\} \bigcup 2^{\Delta_F}$ where *N* denotes the label "normal" which indicates that no faulty transition has fired, and $2^{\Delta_F}$ denotes the power set of $\Delta_F$, that is, $2^{\Delta_F}$ is the set of all subsets of $\Delta_F$. In the rest of the thesis, unobservable places and unobservable transitions are represented by striped places and striped transitions as shown in Figure 3.1.

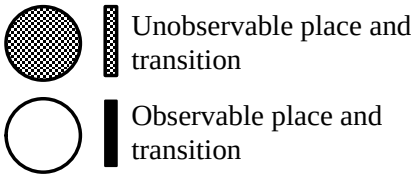


**Figure 3.1 :** Representation of places and transitions.

### 3.2 Fault Diagnosis Based on Petri Net Models

Due to the existence of unobservable places, some markings cannot be distinguished. We denote $M_1 \equiv M_2$ if $M_1(p_i) = M_2(p_i)$ for any $p_i \in P_o$, in other words, the observations of markings $M_1$ and $M_2$ are the same. It is useful to define the quotient set $\hat{R}(M_0)$ as in [30] with respect to the equivalence relation $(\equiv)$; $\hat{R}(M_0) := {}^{R(M_0)}\!/_{\equiv} := \{\hat{M}_0, ..., \hat{M}_n, ...\}$ where $M_0 \in \hat{M}_0$. An element of $\hat{R}(M_0)$ is referred to the observation of a marking or an observable marking.

For simplicity, we impose the following two assumptions in this thesis.

**Assumption 1 [21], [26]:** A Petri net system *PN* defined by (3.1) is *bounded and deadlock-free.*

**Assumption 2 [21], [26]:** There does not exist a sequence of unobservable transitions whose firing generates a cycle of markings which have the same observation, that is, for any $M_i \in R(M_0)$ and $t_i \in T_{uo}$, $i = 1, 2, ..., n$, $M_1[t_1 > M_2[t_2 > \ldots M_n[t_n > M_1 \Rightarrow \exists i, j \in \{1, 2, \ldots, n\}: M_i \neq M_j$.

We define a diagnoser [21], [26], [29] for a Petri net system *PN*. A state $q_d$ of the diagnoser is of the form $q_d = \{(M_1, l_1), (M_2, l_2), \ldots, (M_n, l_n)\}$, which consists of pairs of a marking $M_i \in R(M_0)$ and a label $l_i \in \Delta$. The notation $Q = 2^{R(M_0)\times\Delta}$ denotes the power set of $R(M_0)\times\Delta$, that is, each element of $Q$ is a subset of $R(M_0)\times\Delta$ and is of the form $\{(M_1, l_1), (M_2, l_2), \ldots, (M_n, l_n)\}$. The diagnoser is an automaton given by

$$G_d = (Q_d, \Sigma_o, \delta_d, q_0), \tag{3.8}$$

where $Q_d \subseteq Q$ is the set of states, $\Sigma_o = \hat{R}(M_0) \cup T_o$ is the set of events, $\delta_d : Q_d \times \Sigma_o \to Q_d$ is the partial state transition function, and $q_0 = \{(M_0, N)\}$ is the initial state. The state set $Q_d$ is the set of states in $Q$ which are reachable from the initial state $q_0$ under the state transition function $\delta_d$. Each observed event $\sigma_o \in \Sigma_o$

represents the observation of a marking in $\hat{R}(M_0)$ or an observable transition in $T_o$. The transition function $\delta_d$ is defined by using the label propagation function and the range function. The label propagation function $LP: R(M_0)\times\Delta\times T^* \to \Delta$ propagates the label (normal or faulty) over a sequence $s \in T^*$ of transitions, where $T^*$ is the set of all finite sequences of elements of $T$, as follows [21], [26]-[28]:

$$LP(M,l,s) = \begin{cases} N, & \text{if } (l=N)\wedge(\forall F_i \in \Delta_F : T_{F_i} \notin s) \\ \{F_i : F_i \in l \vee T_{F_i} \in s\}, & \text{otherwise,} \end{cases} \quad \textbf{(3.9)}$$

where $T_{F_i} \in s$ (respectively, $T_{F_i} \notin s$) indicates that a sequence $s \in T^*$ of transitions contains (respectively, does not contain) a faulty transition with failure type $F_i$. Briefly, if the sequence of transitions does not include any faulty transition, then the label attached to the resulting marking is normal ($N$). If the sequence of transitions includes a faulty transition, then the label includes the corresponding failure type. Then, the range function $LR: Q\times\Sigma_o \to Q$ is obtained by modifying its definition of [29] as follows:

$$LR(q,\sigma_o) = \bigcup_{(M,l)\in q} \bigcup_{s\in T^*(M,\sigma_o)} \{(M', LP(M,l,s))\} \quad \textbf{(3.10)}$$

where $M[s>M'$, and $T^*(M,\sigma_o) \subseteq T^*$ is defined in the following two cases:

**1.** If $\sigma_o \in \hat{R}(M_0)$,

$$T^*(M,\sigma_o) = \begin{cases} \varnothing, & \text{if } M \in \sigma_o \\ \{s \in T_{uo}^* : (M_s \in \sigma_o)\wedge(\forall s'(\neq s) \in \overline{s} : M_{s'} \equiv M)\}, & \text{otherwise,} \end{cases} \quad \textbf{(3.11)}$$

where $M[s>M_s$, $M[s'>M_{s'}$, and $\overline{s}$ denotes the set of all prefixes of $s$. In (3.11), the case of $M \notin \sigma_o$ corresponds to a change of the observable marking. In this case, $T^*(M,\sigma_o)$ is the set of sequences $s \in T_{uo}^*$ of unobservable transitions such that,

during the firing of *s*, all of the interval markings except the last one in $\sigma_o$ have the same observation.

**2.** If $\sigma_o \in T_o$,

$$T^*(M,\sigma_o)=\left\{s\sigma_o : \left(s\in T_{uo}^*\right)\wedge\left(M\left[s\sigma_o>\right)\wedge\left(\forall s'\in \bar{s} : M_{s'}\equiv M\right)\right\}, \qquad \textbf{(3.12)}$$

where $M\left[s'>M_{s'}\right.$. When the firing of an observable transition $\sigma_o \in T_o$ is observed, $T^*(M,\sigma_o)$ is the set of sequences of unobservable transitions followed by $\sigma_o$ such that all of the interval observable markings except the last one are the same.

That is, $T^*(M,\sigma_o)$ is the set of possible transition sequences from *M* which are consistent with the observed event $\sigma_o$.

**Remark 3.1:** In this thesis, we modify the definition of $T^*(M,\sigma_o)$ of [29] as follows:

- When $M\in\sigma_o\in\hat{R}(M_0)$, we let $T^*(M,\sigma_o)=\varnothing$, instead of $T^*(M,\sigma_o)=\{\varepsilon\}$ [29], to avoid the self-loop labeled by the current observable marking in $G_d$.
- When $\sigma_o\in T_o$, $M\in\sigma_o$ is impossible, so this case is not considered.

Finally, the transition function $\delta_d : Q_d\times\Sigma_o\to Q_d$ is defined as follows [21], [26]-[28]:

$$\delta_d(q,\sigma_o)=\begin{cases}LR(q,\sigma_o), & \text{if } LR(q,\sigma_o)\neq\varnothing\\ \text{undefined}, & \text{otherwise.}\end{cases} \qquad \textbf{(3.13)}$$

### 3.3 Diagnosability Analysis

A Petri net system *PN* is said to be diagnosable [21] if the type of the fault is always detected within a uniformly bounded number of firings of transitions after the occurrence of the fault. It is possible to classify states in $Q_d$ as follows:

**1.** A state $q\in Q_d$ is said to be $F_i$-certain if $F_i\in l$ for any $(M,l)\in q$.

**2.** A state $q \in Q_d$ is said to be $F_i$-uncertain if there exist $(M, l)$ and $(M', l') \in q$ such that $F_i \in l$ and $F_i \notin l'$. (With a slight abuse of notation, we let $F_i \notin N$).

If the system is diagnosable then, after the occurrence of a faulty transition (e.g. $t \in T_{F_i}$), the state of the diagnoser reaches an $F_i$-certain state within a finite number of firings of transitions [21]. A set $\{q_1, q_2, ..., q_n\} \subseteq Q_d$ of $F_i$-uncertain states is named an $F_i$-indeterminate cycle [21], [26], if the following conditions hold:

**1.** The states $q_1, q_2, ..., q_n \in Q_d$ constitute a cycle in $G_d$, that is, there exist $\sigma_1, \sigma_2, ..., \sigma_n \in \Sigma_o$ such that, $\delta_d(q_j, \sigma_j) = q_{j+1}$ for each $j = 1, 2, \ldots, n-1$ and $\delta_d(q_n, \sigma_n) = q_1$.

**2.** For each $j = 1, 2, \ldots, n$, $k = 1, 2, \ldots, m$, and $r = 1, 2, \ldots, m'$, there exist $(M_j^k, l_j^k), (\tilde{M}_j^r, \tilde{l}_j^r) \in q_j$ which satisfy the following two conditions. In the second condition (b), $M_j^k$ $(j = 1, 2, \ldots, n,\ k = 1, 2, \ldots, m)$ constitute a cycle involving $nm$ markings that carry $F_i$ in their labels, whereas the markings $\tilde{M}_j^r$ $(j = 1, 2, \ldots, n,\ r = 1, 2, \ldots, m')$ constitute a cycle involving $nm'$ markings that do not carry $F_i$ in their labels. We use $m$ and $m'$ to indicate the number of times the cycle $q_1$, $q_2$, ..., $q_n \in Q_d$ is completed in $G_d$ before the cycles of $M_j^k$ and $\tilde{M}_j^r$ are completed, respectively [26]. Thus, $k$ (respectively, $r$) is used to denote that the $k$th (respectively, $r$th) cycle $q_1$, $q_2$, ..., $q_n \in Q_d$ is executed in $G_d$.

(a) For any $j = 1, 2, \ldots, n$, $k = 1, 2, \ldots, m$, and $r = 1, 2, \ldots, m'$, $F_i \in l_j^k$ and $F_i \notin \tilde{l}_j^r$.

(b) Markings $M_j^k (j = 1, 2, \ldots, n,\ k = 1, 2, \ldots, m)$ and $\tilde{M}_j^r (j = 1, 2, \ldots, n,\ r = 1, 2, \ldots, m')$ satisfy the following conditions:

$$
\begin{aligned}
&\exists s_j^k \in T^*\left(M_j^k,\sigma_j\right): M_j^k\left[s_j^k > M_{j+1}^k\left(j=1,2,\ldots,n-1,k=1,2,\ldots,m\right),\right.\\
&\exists s_n^k \in T^*\left(M_n^k,\sigma_n\right): M_n^k\left[s_n^k > M_1^{k+1}\left(k=1,2,\ldots,m-1\right),\right.\\
&\exists s_n^m \in T^*\left(M_n^m,\sigma_n\right): M_n^m\left[s_n^m > M_1^1,\right.\\
&\exists \tilde{s}_j^r \in T^*\left(\tilde{M}_j^r,\sigma_j\right): \tilde{M}_j^r\left[\tilde{s}_j^r > \tilde{M}_{j+1}^r\left(j=1,2,\ldots,n-1,r=1,2,\ldots,m'\right),\right.\\
&\exists \tilde{s}_n^r \in T^*\left(\tilde{M}_n^r,\sigma_n\right): \tilde{M}_n^r\left[\tilde{s}_n^r > \tilde{M}_1^{r+1}\left(r=1,2,\ldots,m'-1\right),\right.\\
&\exists \tilde{s}_n^{m'} \in T^*\left(\tilde{M}_n^{m'},\sigma_n\right): \tilde{M}_n^{m'}\left[\tilde{s}_n^{m'} > \tilde{M}_1^1.\right.
\end{aligned}
\qquad \textbf{(3.14)}
$$

**Remark 3.2:** The above definition of an $F_i$-indeterminate cycle is obtained by slightly modifying the definition of [21] by taking the existence of observable transitions into account.

**Theorem 3.1 [21], [26]:** A Petri net system *PN* is diagnosable if and only if the diagnoser given by (3.8) does not contain an $F_i$-indeterminate cycle for any failure type $F_i$.

The proof of this theorem can be found in [26].

### 3.4 Case Study: Modeling of the Railway Field Components and Fault Diagnosis

The train movement in a fixed-block railway system mainly relies on restrictions and prohibitions. The specifications will be explained by the help of an example railway field shown in Figure 3.2 which includes four railway blocks T001, T002, PMT01, and T003.

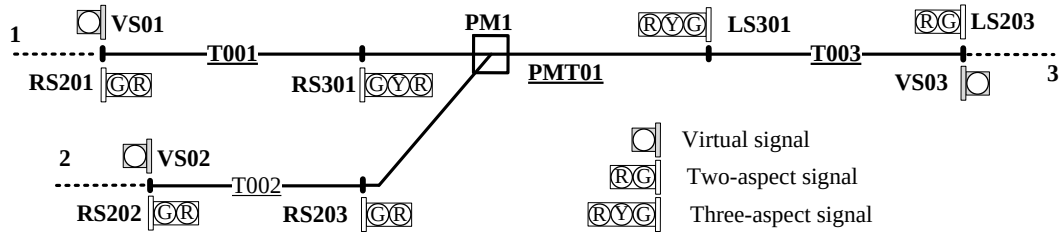


**Figure 3.2 :** An example railway field.

The given railway field layout in Figure 3.2 is general enough in the sense that it includes the main railway field components such as a PM, track circuits, and railway wayside signals. For example, regardless of the scale of a railway field, the operational logic of a signal is the same. For example, the red signal always means that a train must stop and the green signal is used to indicate that the next two

railway blocks are free. The signaling system control software always checks the incoming requests and the condition of the railway field equipment. When there is no reserved route, all signals must be red. If any signal in the field becomes any other color than red, the signaling system control software sends a warning message to the TCC. Furthermore, the given railway field including these equipment can be easily extended to check diagnosability of large railway fields.

First of all, a route must be reserved for a single train to allow its movement and prevent it from colliding with others. Several conditions have to be checked before and after the route reservation. As an initial condition, to accept a route request which is made by the TCC, there must not be any intersecting pre-reserved route. For instance, if the route 1 to 3 is requested by the TCC, the route 3 to 1, 2 to 3, or 3 to 2 must not be pre-reserved before. All these restrictions can be summarized in a table known as the interlocking table [51].

After the acceptance of a route request, the signaling control software sends proper commands to adjust the position of the PM. In Figure 3.2, if the route request from 2 to 3 or 3 to 2 is accepted, then the PM1 has to be adjusted to the reverse position. Later, when the PM has the proper position related with the requested route, the signaling software sends appropriate information to the entrance signal of the route. The entrance signals of the routes 1 to 3, 3 to 1, 2 to 3, and 3 to 2 are RS301, LS301, RS203, and LS301, respectively. If all components have proper conditions (the PM is in the proper position and the entrance signal shows the right color), then the signaling control software reserves the route and electronically locks the components of the route. This electronic lock of the components can be regarded as an additional prevention of the components. For example, in order to prevent a train from derailment, the PM must keep its position while its railway block is occupied by a train.

Another important issue is to check the condition of the field components periodically such as the color information shown by the signals, the positions of the PMs etc. As a world-wide operational condition, a PM has to change its position in pre-defined time which is 7 seconds in the Turkish State Railways and a PM must not remain in the middle of the movement when it starts to change its position. On the other hand, malfunctions such as wrong signal color indication and no signal

color indication can occur at signals. All these unwanted situations have to be taken into account by the signaling control software and the safety of the system has to be provided immediately. For more explanations about the interlocking principles, the reader is referred to [41].

At last, the signals VS01, VS02, and VS03 in Figure 3.2 indicate the virtual signals which do not exist in the railway field and are used for the route reservations from signal to signal by the dispatchers.

### 3.4.1 Modeling by Petri Nets

In this chapter, compact Petri net models of the railway field components are explained. The Petri net models given in this chapter are *ordinary* and *safe*. Since there are four possible route reservations intersecting with each other for the given railway field in Figure 3.2, the Petri net model related with the route reservations can be represented by five places (see Figure 3.3). The meanings of the places and transitions for all Petri net models are given in Table 3.1 in this chapter.

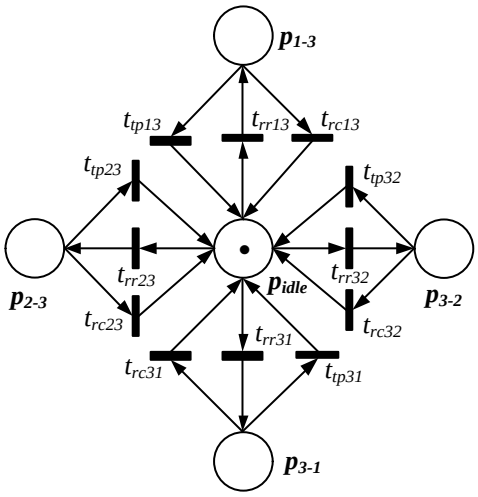


**Figure 3.3 :** Petri net model of the route reservations.

As can be seen from Figure 3.3, only one route reservation is permitted to prevent the trains from collisions.

The Petri net model of the PM is given in Figure 3.4. As represented in Figure 3.4, some places and some transitions are assumed as unobservable. The position of the PM can be detected by the help of the position sensors. So the movement of the PM from one position to another is assumed as unobservable (the

PM is in the middle of movement). As an operational restriction, the PM has to change its position in 7 seconds. When the PM begins to change its position, the token in $p_2$ (or $p_1$) moves to the unobservable place $p_3$ (or $p_4$) over the transition $t_3$ (or $t_1$). If the PM does not reach its new position in 7 seconds, then the faulty transition $t_{f2}$ (or $t_{f1}$) fires and the token in the unobservable place $p_3$ (or $p_4$) moves to $p_{f2}$ (or $p_{f1}$) over the transition $t_{f2}$ (or $t_{f1}$). On the other hand, if the PM reaches its new position, the token in $p_3$ (or $p_4$) moves to $p_1$ (or $p_2$) over the transition $t_4$ (or $t_2$).

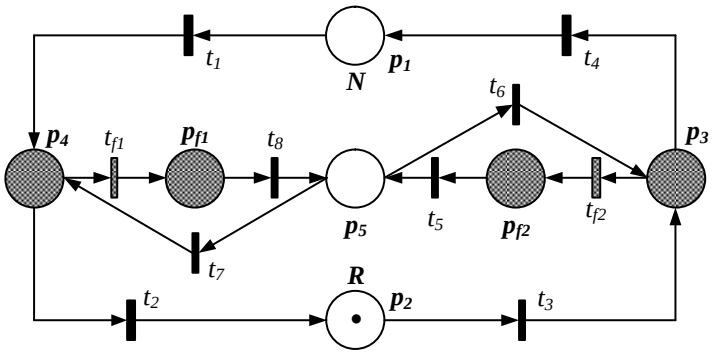


**Figure 3.4 :** Petri net model of the PM.

The Petri net models of the signals are given in Figure 3.5 (a: two-aspect signal, b: three-aspect signal). Initially, all signals in the railway field are red. Signals can show proper color indications other than red, according to route reservations. Assume that it is desired to reserve the route 1 to 3. When the route 1 to 3 is reserved, a train occupies the blocks T001, PMT01 and T003, respectively.

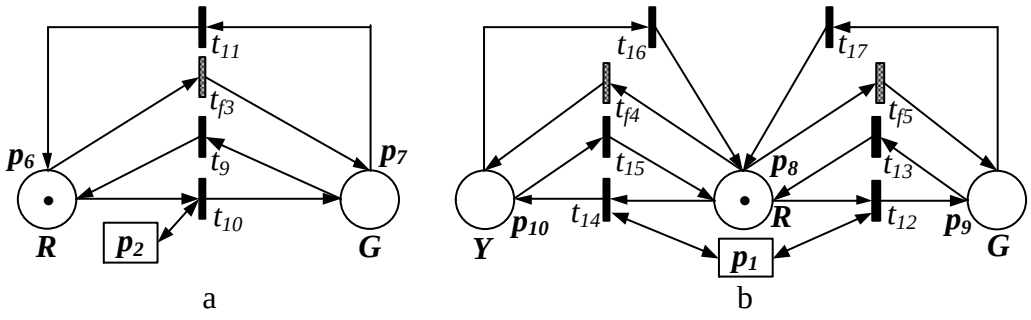


**Figure 3.5 :** Petri net models of the signals.

The Petri net models given in Figure 3.5 represent the signals RS203 and RS301. *R, Y* and *G* indicate Red, Yellow, and Green colors, respectively. After the acceptance of the related route request, it is expected that the PM is in the normal position. If it is not in the proper position, then the signaling control software sends an appropriate command to the PM. When the PM is in the proper position, the token

in the place $p_8$ moves to the place $p_9$ by the firing of the transition $t_{12}$. The places $p_1$ and $p_2$ denoted by rectangles in Figure 3.5 represent such conditions. The signal RS203 can be green ($p_7$) if the PM is in the reverse position ($p_2$). Similarly, the signal RS301 can be green ($p_9$) or yellow ($p_{10}$) if the PM is in the normal position ($p_1$). The signal RS301 remains to be green until a train enters the railway block PMT01 or a route cancellation request from the TCC is received.

The unobservable transitions $t_{f3}$, $t_{f4}$, and $t_{f5}$ represent the faulty transitions which can be encountered in signal malfunctions. In some cases, signals can show wrong color indications due to electromagnetic fields and cable short circuits. When such a fault occurs, it has to be detected immediately by the signaling control software, and depending on the condition of the other components, the system moves to safe-state (e.g. all route requests are rejected and all signals related to the faulty signal become red).

Finally, the train movement is monitored and detected by the help of the track circuits and each railway block is modeled by a single place with entrance and exit transitions which represent the direction of the movement (see Figure 3.6). The train is expected to enter and exit the railway blocks in order. When a train enters the related railway block, a token is put in its related place (e.g. for T001, a token is added to the place $p_{11}$ by the firing of $t_{18}$ or $t_{21}$) depending on the direction of movement. The token in the place $p_{11}$ is removed by the firing of $t_{20}$ (or $t_{19}$) when the train exits the track T001. The places $p_{11\_1}$, $p_{12\_1}$, $p_{13\_1}$ and $p_{14\_1}$ are used to restrict the train entrance into the railway blocks while the corresponding block is occupied by another train. Additionally, the entrance of each railway block is enabled by the signal colors. The entrance of a train to T001 and T003 are enabled by the signals RS201 and LS203, respectively, as shown in Figure 3.6. Similarly, for the routes 1 to 3 or 2 to 3, the entrance of a train to PMT01 is enabled by the signals RS301 or RS203. After the route 1 to 3 is reserved, a train in T001 can enter PMT01 when the signal RS301 is yellow or green. The places $p_7$, $p_9$ and $p_{10}$ denoted by rectangles in Figure 3.6 represent the restriction of the entrance of the train to PMT01 by the signals RS301 ($p_9$, $p_{10}$) or RS203 ($p_7$). Depending on the condition of the railway field, the entrance into PMT01 is enabled by the proper signal colors. Similar

restrictions have to be added to each railway block Petri net model but not shown here for simplification.

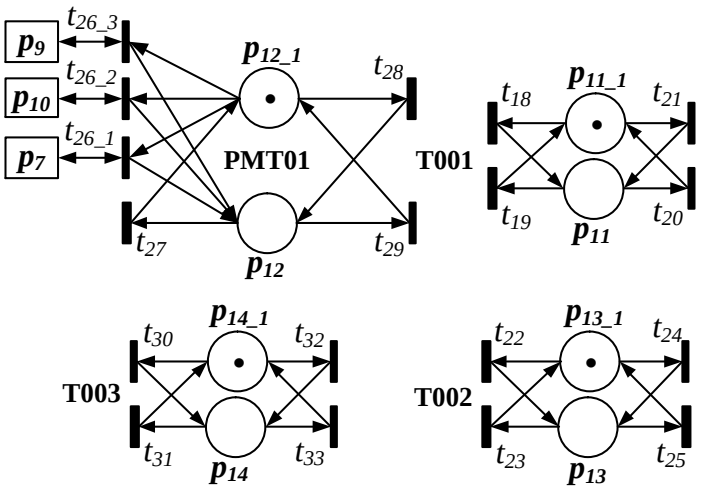


**Figure 3.6 :** Petri net models of the railway blocks.

Note that the Petri net models of the railway field components given in this thesis are compact models which just represent the main operational behavior of the components. These models can then be converted to software blocks by using different approaches [79]-[81]. For the railway field shown in Figure 3.2, the software contains six signals, one PM, and four railway blocks including their relations.

### 3.4.2 Some Possible Faults

In railway signaling systems, in order to provide a safe and effective transportation, failures can be classified according to their importance. Some failures can be overcome by just informing the TCC and the train drivers of their occurrences, whereas some failures cannot. In the latter case the whole system has to be stopped. For instance, when a PM malfunction occurs, the signaling system informs the TCC of the occurrence and do not permit the train movement in the PM area. Furthermore, trains can be guided to alternative ways. Likewise, if a track circuit (or an axle counter) malfunction occurs such as the disappearance of the train on the railway line, the signaling system has to stop trains immediately and also informs the TCC. The former case for which raising an alarm is enough is referred to an alarm condition, whereas the latter case for which trains have to stop is referred to

a safe-state condition. For the given railway field of Figure 3.2, some possible failures and their results can be summarized as follows:

**Table 3.1 :** Meanings of places and transitions in the model given in Figure 3.3 - Figure 3.6.

| Place | Meaning | Transition | Meaning |
|---|---|---|---|
| $p_{idle}$ | There is no route reservation | $t_1$ | PM left normal position |
| $p_{1\text{-}3}$ | The route from 1 to 3 is reserved | $t_2$ | PM reached reverse position |
| $p_{2\text{-}3}$ | The route from 2 to 3 is reserved | $t_3$ | PM left reverse position |
| $p_{3\text{-}1}$ | The route from 3 to 1 is reserved | $t_4$ | PM reached normal position |
| $p_{3\text{-}2}$ | The route from 3 to 2 is reserved | $t_5$ | 7 seconds passed |
| $p_1$ | PM is in normal position | $t_6$ | Move PM to normal position |
| $p_2$ | PM is in reverse position | $t_7$ | Move PM to reverse position |
| $p_3$ | PM is moving from reverse position to normal position | $t_8$ | 7 seconds passed |
| $p_4$ | PM is moving from normal position to reverse position | $t_9$ | Turn signal to red |
| $p_5$ | PM does not reach desired position | $t_{10}$ | Turn signal to green |
| $p_6$ | Two-aspect signal is red | $t_{11}$ | Signal fault acknowledged |
| $p_7$ | Two-aspect signal is green | $t_{12}$ | Turn signal to green |
| $p_8$ | Three-aspect signal is red | $t_{13}$ | Turn signal to red |
| $p_9$ | Three -aspect signal is green | $t_{14}$ | Turn signal to yellow |
| $p_{10}$ | Three -aspect signal is yellow | $t_{15}$ | Turn signal to red |
| $p_{11}$ | Railway block T001 is occupied | $t_{16}$ | Signal fault acknowledged |
| $p_{12}$ | Railway block PMT01 is occupied | $t_{17}$ | Signal fault acknowledged |
| $p_{13}$ | Railway block T002 is occupied | $t_{18}$ ($t_{21}$) | Occupy railway block T001 |
| $p_{14}$ | Railway block T003 is occupied | $t_{19}$ ($t_{20}$) | Vacate railway block T001 |
| $p_{f1}$ | PM did not reach reverse position | $t_{22}$ ($t_{24}$) | Occupy railway block T002 |
| $p_{f2}$ | PM did not reach normal position | $t_{23}$ ($t_{25}$) | Vacate railway block T002 |
| $p_{11\_1}$ | Restriction of T001 | $t_{26\_1}$ . $t_{26\_3}$ ($t_{28}$) | Occupy railway block PMT01 |
| $p_{12\_1}$ | Restriction of PMT01 | $t_{27}$ ($t_{29}$) | Vacate railway block PMT01 |
| $p_{13\_1}$ | Restriction of T002 | $t_{30}$ ($t_{32}$) | Occupy railway block T003 |
| $p_{14\_1}$ | Restriction of T003 | $t_{31}$ ($t_{33}$) | Vacate railway block T003 |
| **Transitions** | **Meaning** | $t_{f1}$ | Fault occurs while PM is moving from normal position to reverse position |
| $t_{tp13}$ | Train has passed the route 1 to 3 | $t_{f2}$ | Fault occurs while PM is moving from reverse position to normal position |
| $t_{rr13}$ | Reserve the route 1 to 3 | $t_{f3}$ | Faulty green color in the signal |
| $t_{rc13}$ | Cancel the route 1 to 3 | $t_{f4}$ | Faulty yellow color in the signal |
| $t_{tp23}$ | Train has passed the route 2 to 3 | $t_{f5}$ | Faulty green color in the signal |
| $t_{rr23}$ | Reserve the route 2 to 3 | $t_{tp32}$ | Train has passed the route 3 to 2 |
| $t_{rc23}$ | Cancel the route 2 to 3 | $t_{rr32}$ | Reserve the route 3 to 2 |
| $t_{tp31}$ | Train has passed the route 3 to 1 | $t_{rc32}$ | Cancel the route 3 to 2 |
| $t_{rr31}$ | Reserve the route 3 to 1 | $t_{rc31}$ | Cancel the route 3 to 1 |

**1. Alarm conditions:**

- A signal which is in the opposite direction for the route shows any color other than red after the route is reserved. Assume that the route 1 to 3 is reserved and the entrance signal RS301 is green. Then the signals LS301, LS203 and RS203 must be red. For instance, assume that the signal LS301 becomes green instead of red after the route 1 to 3 is reserved. In this situation, the route 1 to 3 have to be cancelled immediately and the entrance signal of the route (RS301) must be red if the train has not entered yet. On the other hand, if the train has entered the route when the fault occurs at the signal LS301, the TCC must inform all other incoming trains of the occurrence of the fault because the route cannot be cancelled while a train is moving. A similar scenario can be explained by the help of the example railway yard given in Figure 3.7. If a fault occurs at the signal RS203 when the route 1 to 3 is not reserved and the block T002 is free, raising an alarm is enough to provide system safety.
- The position of the PM is corrupted after a route is reserved. Assume that the route 3 to 2 is reserved, the PM is in the reverse position, and the entrance signal LS301 is yellow. The position of the PM must be locked in reverse until the route is cancelled or a train has passed. When the position of the PM is corrupted, the route has to be cancelled if there is no train in the route. If a train is moving in the route when the fault occurs in the PM, the train driver must be informed by the TCC about the occurrence of the fault. In both cases, an alarm signal is also produced to inform the dispatchers.

**2. Safe-state conditions:**

- A signal which is in the opposite direction for the route shows any color other than red after the route is reserved and the railway block is occupied. Assume that the route 1 to 3 is reserved and the train occupies the blocks T001 and PMT01. In this situation, the entrance signal RS301 is red because the train has entered PMT01. Assume also that the railway block T002 is also occupied by another train. If the color of the signal RS203 becomes green by the occurrence of a fault (while a train is moving in the route 1 to 3), then the whole system have to move to the safe-state where all

signals must be red and all trains have to be stopped immediately in order to prevent collisions. This scenario is illustrated in Figure 3.7.

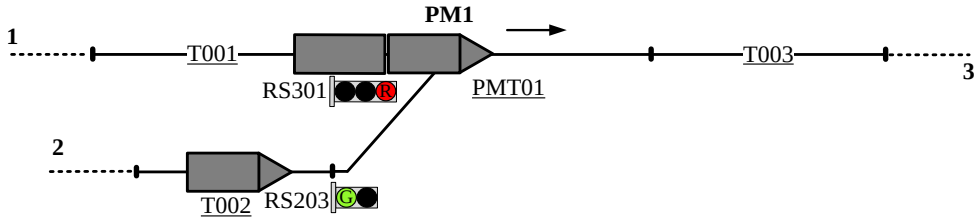


**Figure 3.7 :** Catastrophic fault.

### 3.4.3 Diagnoser Design

The diagnoser is designed by using the procedure given in [21], [26], [29], [30]. Although there are four possible route reservations for the railway field given in Figure 3.2, only a part of the diagnoser for the reservation of the route 1 to 3 is given in Figure 3.8. The diagnoser in Figure 3.8 partially represents the route reservation process after the route 1 to 3 is requested from the TCC and accepted by the signaling control software. The other route reservations are not mentioned here but they can be dealt with in a similar manner.

Each state represented by a rectangle consists of a pair of a marking of places $p_1$ to $p_{14}$, $p_{f1}$, and $p_{f2}$ and a label $N$ or $F_i$ $\left(i \in \{1,2,3,4,5\}\right)$. That is, in the part of the diagnoser, a marking just after an observed event is uniquely determined. Also, multiple failures are not dealt with for simplification. Each failure type $F_i$ is related with the faulty transition $t_{fi}$. Each state transition is labeled by the observation of a marking or a pair of the observation of a marking and an observable transition with a slight abuse of notation. (According to the definition of (3.8), each state transition is labeled by the observation of a marking or an observable transition. In Figure 3.8, for ease of understanding, the observation of a marking is also attached in the latter case.) The former corresponds to the case where only the change of the observation of a marking is observed, whereas the latter corresponds to the case where the firing of an observable transition is observed with the observation of a marking. For instance, at the initial state of the diagnoser, the event label $\hat{M}1\text{-}t_3$ indicates that the firing of an observable transition $t_3$ is observed with the observation $\hat{M}1$ of the resulting marking. Also, at the state $\left\{\left((1,0,\underline{0},\underline{0},0,1,0,1,0,0,0,0,1,0,\underline{0},\underline{0}), N\right)\right\}$ with the

observable marking $\hat{M}20$, the event label $\hat{M}23$ represents that the observable marking $\hat{M}20$ changes to $\hat{M}23$ by the firing of an unobservable transition $t_{f5}$ which is not indicated in the event label. Note that in any marking, underlines are used to indicate unobservable places.

According to Theorem 3.1 given in Subchapter 3.2 and the part of the diagnoser given in Figure 3.8, there is no $F_i$-indeterminate cycle for any failure type $F_i$ and the system is diagnosable in the considered situation. (Note that we can verify that the system is entirely diagnosable.) Due to safety reasons, it should be an expected result. In Figure 3.8, the thick rectangles indicate the safe-state conditions whereas the dashed rectangles indicate alarm conditions.

While reserving the route 1 to 3, if a fault occurs in the PM movement (the faulty transition $t_{f2}$ in Figure 3.4 fires), an alarm is raised and the route reservation is cancelled. This situation is represented by the state $\{((0,0,\underline{0},\underline{0},1,1,0,1,0,0,0,0,0,0,\underline{0},\underline{0}), F_2)\}$ with the observable marking $\hat{M}6$ which means that the PM did not reach the desired position in 7 seconds. At this state, the dispatcher in the TCC can request to move the PM either to the reverse or to the normal position. When the PM moves to the proper position for the route 1 to 3 and is locked, if a fault occurs at the entrance signal of the route (the signal RS301) or at the signal RS203 (the faulty transitions $t_{f3}$, $t_{f4}$, or $t_{f5}$ fires), the route reservation is also cancelled. This situation is represented by the state $\{((0,0,\underline{1},\underline{0},0,0,1,1,0,0,0,0,0,0,\underline{0},\underline{0}), F_3)\}$ with the observable marking $\hat{M}7$, $\{((0,0,\underline{1},\underline{0},0,1,0,0,0,1,0,0,0,0,\underline{0},\underline{0}), F_4)\}$ with the observable marking $\hat{M}8$ or $\{((0,0,\underline{1},\underline{0},0,1,0,0,1,0,0,0,0,0,\underline{0},\underline{0}), F_5)\}$ with the observable marking $\hat{M}9$. The safe-state condition may occur if, after the route 1 to 3 is reserved, the railway block T002 is occupied and a train is moving in the route. These are indicated as thick rectangles in Figure 3.8. In such a situation, the trains have to stop by the help of the Automatic Train Protection (ATP) and Automatic Train Stop (ATS) equipment which are out of the scope of this thesis.

A benefit of such a diagnoser design is that it enables us to check the adequacy of the constructed Petri net models. While developing signaling control software (see Figure 2.1), the designers have to preform worst-case analysis which can also be considered as a deal between the system safety and ease of operations. Since the Petri net models have to be simple for easy error tracking of the software and also reliable for safe transportation, the designers have to cope with both simplicity and reliability specifications. For example, if the signaling control software is developed by just considering the safety issues at first sight, then the whole system may fall into the safe-state condition in case of any simple failure such as the stoppage of the whole train traffic with a simple signal malfunction. This will result in long headways in railway operations and waste of time. On the other hand, if safety issues are not considered enough, then accidents may occur, which is also an unwanted situation. The design of a diagnoser using the DESs approach helps to decide what should be done in case of each failure. For some failures, just raising an alarm could be enough to provide the system safety (e.g. $\{((1,0,\underline{0},\underline{0},0,1,0,0,0,1,0,0,1,0,\underline{0},\underline{0}), F_4)\}$ with the observable marking $\hat{M}22$ and $\{((1,0,\underline{0},\underline{0},0,1,0,0,1,0,0,0,1,0,\underline{0},\underline{0}), F_5)\}$ with the observable marking $\hat{M}23$ in Figure 3.8) whereas in some cases the whole system has to move into the safe-state where all signals are red, all PMs are locked, and all trains have to be stopped immediately (e.g. $\{((0,0,\underline{1},\underline{0},0,0,1,1,0,0,0,0,1,0,\underline{0},\underline{0}), F_3)\}$ with the observable marking $\hat{M}17$ and $\{((0,0,\underline{1},\underline{0},0,0,1,1,0,0,1,0,1,0,\underline{0},\underline{0}), F_3)\}$ with the observable marking $\hat{M}18$ in Figure 3.8). Briefly, designing a diagnoser can be considered as an additional safety procedure which allows designers to verify the accurateness of the related Petri net models and so the developed software.

### 3.5 Concluding Remarks

The structure of the fixed-block railway signaling systems enables signaling software designers to study these systems as DESs. Since the railway systems are in the class of safety-critical systems where human life is in question, the detection of the occurrence of the fault as soon as possible and preventing the system from possible faulty conditions is the most important issue. The safety of the entire system

has to be guaranteed at all times. In this chapter, a sample railway field is modeled by using a well-known modeling tool Petri nets and the DESs based fault diagnosis approach is applied to design a diagnoser. Designing a diagnoser can be time-consuming but enables signaling software designers to verify their models before proceeding to the test phase of the developed signaling system software. Even for a small-scaled railway field as given in Figure 3.2, the testing phase of all possibilities of the developed signaling system software can take one week. After testing the developed signaling system software, the errors should be reported and later, the developers should go back to the design phase and fix the reported errors. After that, all the tests have to be performed from the very beginning [4].

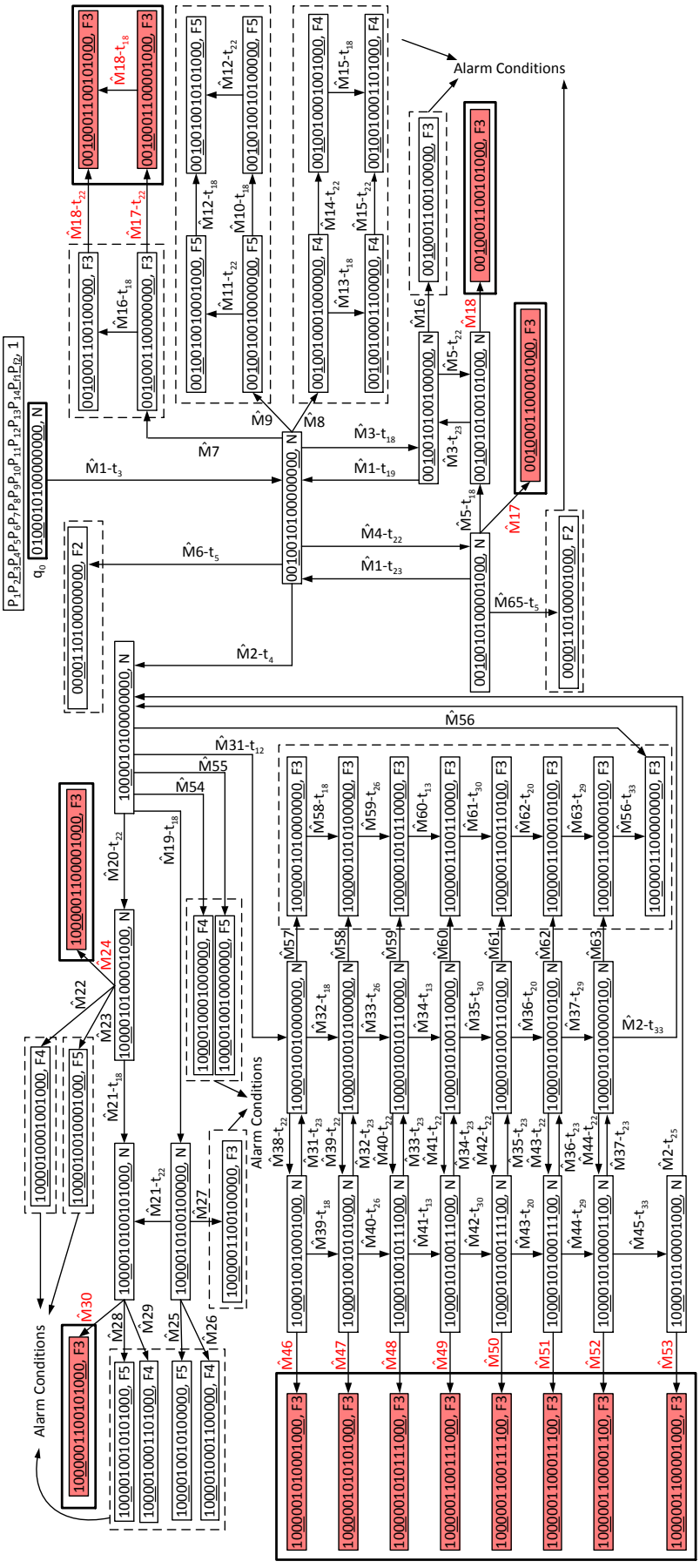


**Figure 3.8 :** A part of the diagnoser for the route 1 to 3.

## 4. Decision Making Strategies in Fixed-Block Signaling Systems

In this chapter, according to the recommendations of the railway-related safety standards, decision making strategies including fault diagnosis are developed based on the Petri net models of railway field components in the interlocking system architecture of [10] which consists of two controllers and a coordinator.

### 4.1 Control Architecture

The control architecture studied in this chapter is given in Figure 4.1 [10]. We use combination of failure assertion programming, defensive programming, and *N*-version programming in this control architecture to satisfy the requirements of the safety standards for developing SIL3 software. Since railway signaling systems can be classified as safety-critical, the signaling system has to be fail-safe and *N* is chosen as 2. It is assumed that there is a reliable communication (e.g. safe-ethernet) between the coordinator, the control center, and the controllers.

In Figure 4.1, the system block represents the railway field. The components in the railway field are desired to be controlled by using two parallel running controllers (e.g. programmable logic controllers) which are assumed to be fail-safe. These controllers are designed by two independent workgroups according to diverse programming, and they do not communicate with each other.

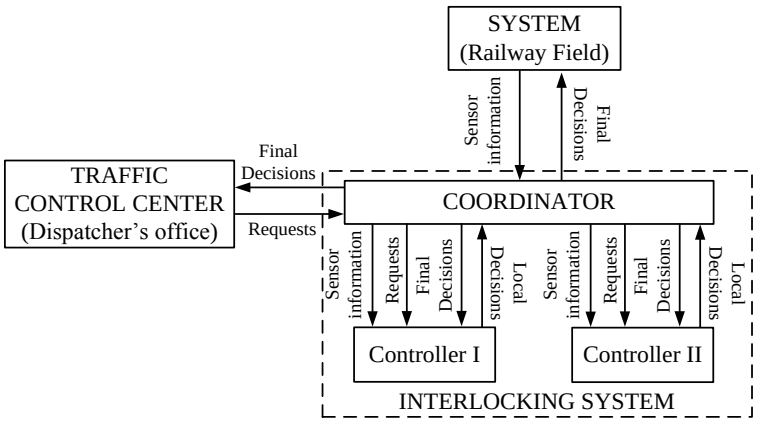


**Figure 4.1 :** The control architecture.

The structure of a controller is shown in Figure 4.2. Each block in the controller is the software block for each railway field component and each route in the railway topology. For every railway field component, a single Petri net model is constructed and added to the corresponding software block. The software block also includes the diagnoser and failure recovery module for the component (e.g. a PM block contains the Petri net model, diagnoser, and failure recovery module for the PM).

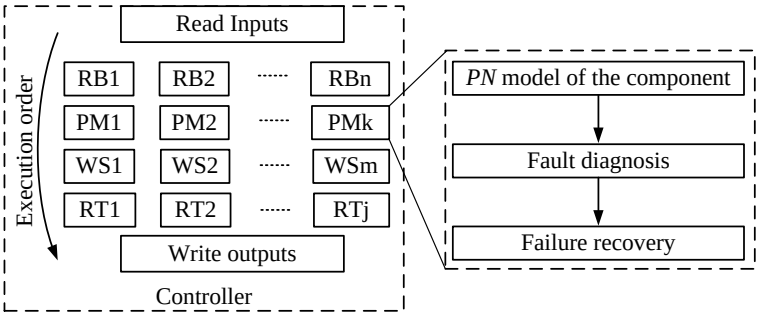


**Figure 4.2 :** Structure of a controller.

As previously mentioned in the introduction section, the use of defensive and failure assertion programming techniques should be implemented in the software block while constructing the Petri net models. For instance, as a general rule that has been accepted all over the world, movement of a PM must be rejected whenever its railway block is occupied. This rule should be taken into account while constructing the Petri net models by the designers as an application of the defensive programming technique. Moreover, as an application of the failure assertion programming technique, position information of a PM and color information of a signal should be checked before and after an incoming request from the control center to ensure it is functioning correctly.

For the purpose of failure diagnosis, we assume that the Petri net model of each railway field component satisfies Assumptions 1 and 2 imposed in Subchapter 3.2. Since we focus on the failure diagnosis, we do not address the issue of failure recovery in this thesis.

The coordinator receives the requests from the control center and the sensor information from the railway field. The requests of the control center are sent to the controllers immediately whereas the sensor information are sent to the controllers periodically (e.g. 2 sec.). Each controller evaluates the requests according to the common

knowledge (e.g. a table where the safety precautions are determined), produces its decision, and then sends it back to the coordinator. The coordinator gives final decisions by comparing controllers' decisions using simple logical operations. It is important to note that the controllers have different cycle times and evaluation strategies due to diverse design [11]. Due to this asynchronous nature, they do not receive the requests from the coordinator simultaneously. Similarly, the coordinator does not receive the decisions of the controllers at the same time.

If one controller detects a fault in a railway field component but the other controller does not, the former reports this fault information to the coordinator immediately and the coordinator sends the fault information to both the traffic control center and the other controller which did not detect the fault. Then, the controller that did not detect the fault acts as there is faulty condition in the related railway field component (e.g. the controller denies the incoming requests related with the faulty field component).

### 4.2 Decision Making Strategies

Since high level of risk is in question, railway signaling systems must have strict rules. As previously mentioned the interlocking system warns the traffic control center or stops the whole system depending on the failure. If the stuck of a PM occurs during a route reservation procedure, the route request should be rejected and information about the failure should be sent to the control center. It is not necessary to stop the whole system in case of such a failure.

In Turkey, most of the railways are rather old as compared with the other European countries. Especially, most of the equipment used in the old fixed-block railway lines are old and faults can occur easily in bad weather conditions such as cold and rain. For example, in a cold day, the blades of these old PMs can stick to each other. Similarly, when there is heavy rain, the blades of the PMs can be stopped in the middle of the movement due to electrical malfunction. Therefore, while developing interlocking software for old railway lines, one of the main requirements of the TCDD is to detect these kinds of faults.

It is supposed that while a train is moving in the pre-reserved route for itself, due to a faulty decision of the controller(s), the position of the PM is changed even if

the railway block of the point machine is occupied. As a result, the cars of the train will be derailed. This situation is illustrated in Figure 4.3. The interlocking software must be designed so that this situation can be avoided.

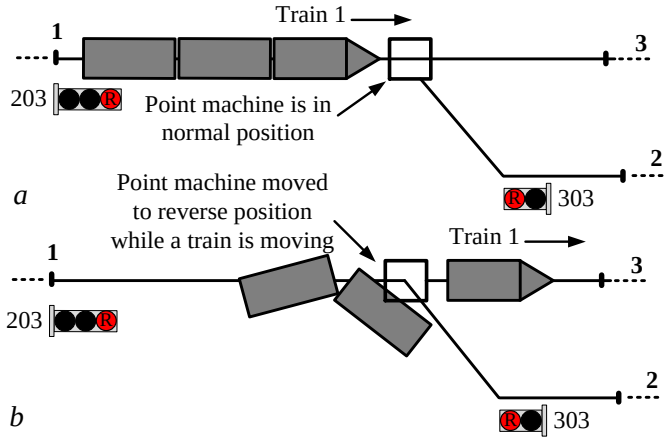


**Figure 4.3 :** Derailment of a train.

### 4.2.1 Petri Net Models and Diagnosers

For each railway field component, its Petri net models are constructed by two different workgroups. Since two controllers obtain the same sensor information on the railway field, we assume that the constructed Petri net models have the common observable places and common observable transitions. Two different Petri net models of a PM are shown in Figure 4.4. Note that, unobservable places and transitions are indicated in Figure 4.4 as striped places and transitions.

Representations of the Petri net models shown in Figure 4.4a and Figure 4.4b are as follows, respectively:

$$P_{c1\_o} = \{p_{c1\_1}, p_{c1\_2}, p_{c1\_5}, p_{c1\_6}\},$$

$$P_{c1\_uo} = \{p_{c1\_3}, p_{c1\_4}, p_{c1\_7}, p_{c1\_8}, p_{c1\_f1}, p_{c1\_f2}\},$$

$$T_{c1\_o} = \{t_{c1\_1}, t_{c1\_2}, t_{c1\_3}, t_{c1\_4}, t_{c1\_5}, t_{c1\_6}, t_{c1\_7}, t_{c1\_8}, t_{c1\_9}, t_{c1\_10}, t_{c1\_11}, t_{c1\_12}\},$$

$$T_{c1\_uo} = \{t_{c1\_f1}, t_{c1\_f2}, t_{c1\_f3}, t_{c1\_f4}, t_{c1\_f5}, t_{c1\_f6}, t_{c1\_f7}, t_{c1\_f8}\},$$

$$M_{c1\_0} = \left(M_{c1\_0}(p_{c1\_1}), M_{c1\_0}(p_{c1\_2}), M_{c1\_0}(p_{c1\_3}), M_{c1\_0}(p_{c1\_4}), M_{c1\_0}(p_{c1\_5}),\right.$$
$$\left. M_{c1\_0}(p_{c1\_6}), M_{c1\_0}(p_{c1\_7}), M_{c1\_0}(p_{c1\_8}), M_{c1\_0}(p_{c1\_f1}),\ M_{c1\_0}(p_{c1\_f2})\right)$$
$$= (0,1,\underline{0},\underline{0},0,0,\underline{0},\underline{0},\underline{0},\underline{0}).$$

$$P_{c2\_o} = \{p_{c2\_1}, p_{c2\_2}, p_{c2\_5}, p_{c2\_6}\},$$

$$P_{c2\_uo} = \{p_{c2\_3}, p_{c2\_4}, p_{c2\_7}, p_{c2\_f1}, p_{c2\_f2}\},$$

$$T_{c2\_o} = \{t_{c2\_1}, t_{c2\_2}, t_{c2\_3}, t_{c2\_4}, t_{c2\_5}, t_{c2\_6}, t_{c2\_7}, t_{c2\_8}, t_{c2\_9}, t_{c2\_10}\},$$

$$T_{c2\_uo} = \{t_{c2\_f1}, t_{c2\_f2}, t_{c2\_f3}\},$$

$$M_{c2\_0} = \left(M_{c2\_0}(p_{c2\_1}), M_{c2\_0}(p_{c2\_2}), M_{c2\_0}(p_{c2\_3}), M_{c2\_0}(p_{c2\_4}), M_{c2\_0}(p_{c2\_5})\right.$$
$$\left. M_{c2\_0}(p_{c2\_6}), M_{c2\_0}(p_{c2\_7}), M_{c2\_0}(p_{c2\_f1}), M_{c2\_0}(p_{c2\_f2})\right)$$
$$= (0,1,\underline{0},\underline{0},0,0,\underline{1},\underline{0},\underline{0}).$$

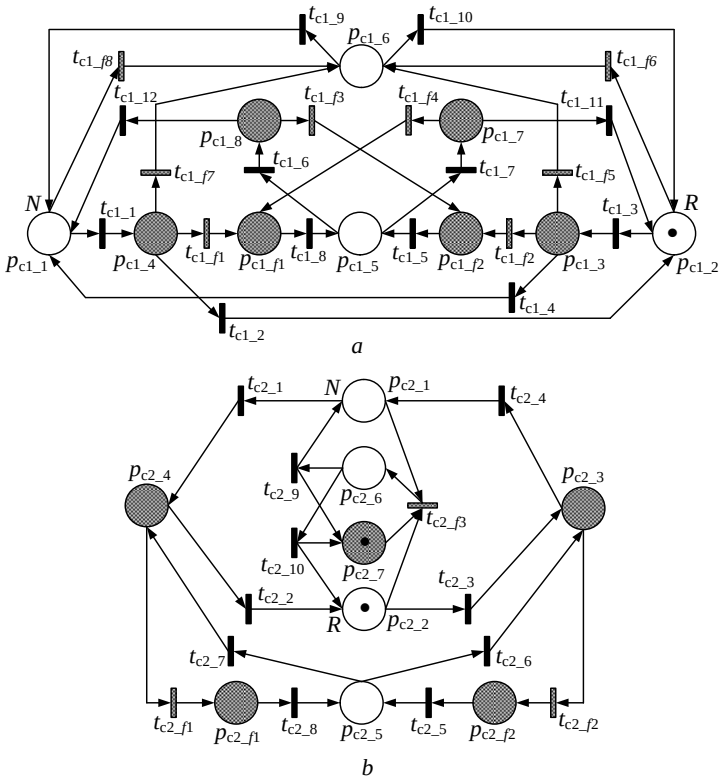


**Figure 4.4 :** Different Petri Net models of a PM.

The meanings of the places and the transitions in these models are given in Table 4.1. Note that in any marking, underlines are used to indicate unobservable places. It is assumed that there are three different failure types $\Delta_F = \{F_1, F_2, F_3\}$, where $T_{F_1} = \{t_{c1\_f1}, t_{c1\_f4}\}$, $T_{F_2} = \{t_{c1\_f2}, t_{c1\_f3}\}$, and $T_{F_3} = \{t_{c1\_f5}, t_{c1\_f6}, t_{c1\_f7}, t_{c1\_f8}\}$ for the Petri net in Figure 4.4a and $T_{F_1} = \{t_{c2\_f1}\}$, $T_{F_2} = \{t_{c2\_f2}\}$, and $T_{F_3} = \{t_{c2\_f3}\}$ for the Petri net in Figure 4.4b. $F_1$ and $F_2$ mean that a PM does not reach its desired position in 7 sec while moving to reverse and normal position, respectively. $F_3$ means the stuck of the blades of the PM. If the actual position (possibly the initial

position) of a PM does not change in a predefined time instance after an incoming PM position request, it is assumed as the stuck of the blades of the PM by the interlocking software.

The Petri net models in Figure 4.4 are slightly different from each other due to the difference between the design strategies of two independent workgroups. Assume that the PM is in the reverse position (the tokens are in the places $p_{c1\_2}$ and $p_{c2\_2}$, respectively) and by an incoming position request from the coordinator, the tokens move to the places $p_{c1\_3}$ and $p_{c2\_3}$ by the firing of $t_{c1\_3}$ and $t_{c2\_3}$, respectively. In the Petri net model given in Figure 4.4a, the token can move to the place $p_{c1\_f2}$ by the faulty transition $t_{c1\_f2}$ or the place $p_{c1\_1}$ by the observable transition $t_{c1\_4}$. Although it is likely to occur, the token in the place $p_{c1\_3}$ can move to the place $p_{c1\_6}$ by the faulty transition $t_{c1\_f5}$. By contrast, this failure situation is not considered by the other workgroup. If the last mentioned situation occurs, only the first controller can detect the occurrence of the fault ($t_{c1\_f5}$). As mentioned before, the main purpose of the diverse programming is to detect the design faults and prevent the system.

**Table 4.1 :** Meanings of places and transitions in the models shown in Figure 4.4.

| Place | Meaning | Transition | Meaning |
|---|---|---|---|
| $p_{c1\_1}$ ($p_{c2\_1}$) | PM is in normal position | $t_{c1\_1}$ ($t_{c2\_1}$) | PM left normal position |
| $p_{c1\_2}$ ($p_{c2\_2}$) | PM is in reverse position | $t_{c1\_2}$, $t_{c1\_11}$, ($t_{c2\_2}$) | PM reached reverse position |
| $p_{c1\_3}$ ($p_{c2\_3}$) | PM is moving from reverse position to normal position | $t_{c1\_3}$ ($t_{c2\_3}$) | PM left reverse position |
| $p_{c1\_4}$ ($p_{c2\_4}$) | PM is moving from normal position to reverse position | $t_{c1\_4}$, $t_{c1\_12}$, ($t_{c2\_4}$) | PM reached normal position |
| $p_{c1\_5}$ ($p_{c2\_5}$) | PM did not reach desired position | $t_{c1\_5}$, $t_{c1\_8}$ ($t_{c2\_5}$, $t_{c2\_8}$) | 7 seconds passed |
| $p_{c1\_6}$ ($p_{c2\_6}$) | PM stuck fault has occurred | $t_{c1\_6}$ ($t_{c2\_6}$) | Move PM to normal position |
| $p_{c1\_7}$ | PM is moving to reverse position after fault | $t_{c1\_7}$ ($t_{c2\_7}$) | Move PM to reverse position |
| $p_{c1\_8}$ | PM is moving to normal position after fault | $t_{c1\_9}$ ($t_{c2\_9}$) | PM stuck fault acknowledge to normal position |
| ($p_{c2\_7}$) | Stuck fault restriction of PM | $t_{c1\_10}$ ($t_{c2\_10}$) | PM stuck fault acknowledge to reverse position |
| $p_{c1\_f1}$ ($p_{c2\_f1}$) | PM position indication fault while moving to reverse position | $t_{c1\_f1}$, $t_{c1\_f4}$ ($t_{c2\_f1}$) | Indication fault occurs while PM is moving to reverse position |
| $p_{c1\_f2}$ ($p_{c2\_f2}$) | PM position indication fault while moving to normal position | $t_{c1\_f2}$, $t_{c1\_f3}$ ($t_{c2\_f2}$) | Indication fault occurs while PM is moving to normal position |
| | | $t_{c1\_f5}$, $t_{c1\_f6}$, $t_{c1\_f7}$, $t_{c1\_f8}$ ($t_{c2\_f3}$) | Stuck of PM blades is detected |

Parts of the diagnosers, defined in Subchapter 3.2, for the Petri net models given in Figure 4.4 are shown in Figure 4.5. Each state represented by a rectangle includes a pair of marking of places and a label $N$ or $F_i$ $\left(i \in \{1,2,3\}\right)$. That is, in the parts of the diagnosers, a marking just after an observed event is uniquely determined. Besides, multiple failures are not dealt with in the thesis for simplification.

As in the diagnoser shown in Figure 3.8, each state transition of the diagnoser is labeled by the observation of a marking or a pair of the observation of a marking and an observable transition with a slight abuse of notation. (According to the definition of (3.8), each state transition is labeled by the observation of a marking or an observable transition.)

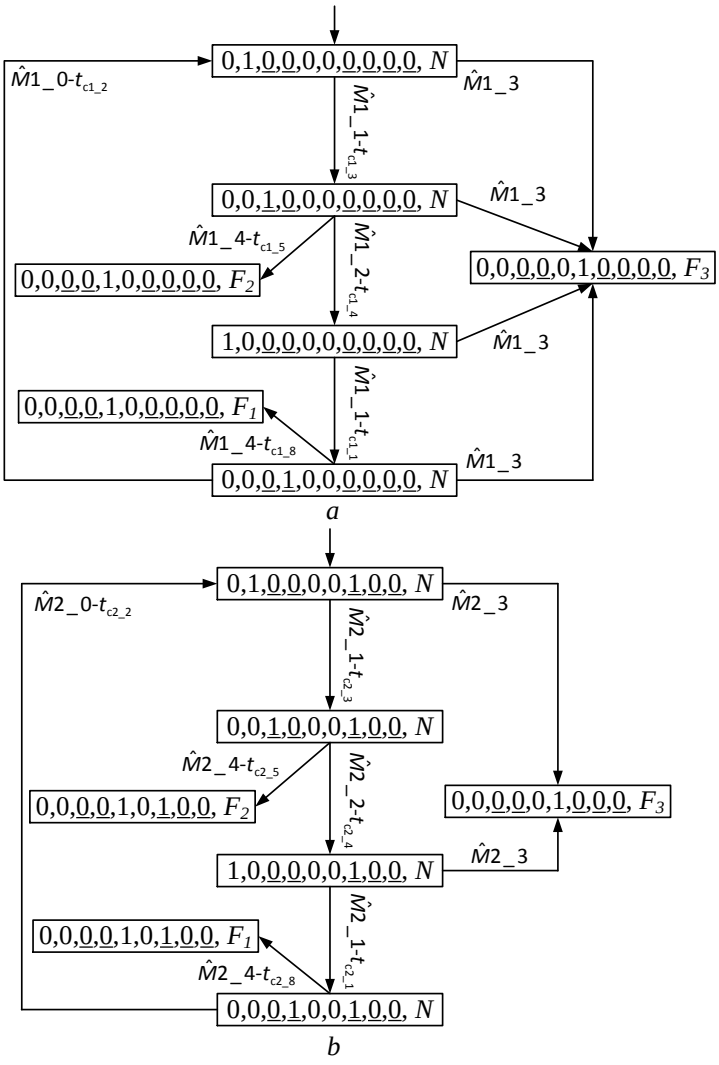


**Figure 4.5 :** Diagnosers of the Petri net models given in Figure 4.4.

For instance, at the initial state of the diagnoser in Figure 4.5a, the event label $\hat{M}1\_3$ represents that the observable marking $\hat{M}1\_3$ is observed by the firing of the unobservable transition $t_{c1\_f6}$. Similarly, the state of the diagnoser shown in Figure 4.5b changes from $\{((0,0,\underline{1},\underline{0},0,0,\underline{1},\underline{0},\underline{0}),N)\}$ to $\{((1,0,\underline{0},\underline{0},0,0,\underline{1},\underline{0},\underline{0}),N)\}$ by the firing of the observable transition $t_{c2\_4}$ with the observation $\hat{M}2\_2$ of the resulting marking. According to Theorem 3.1 given in Subchapter 3.2, since there is no $F_i$-indeterminate cycle in the diagnosers, both Petri net models are diagnosable. As discusses in Subchapter 3.4, the diagnosability concept is a must for railway interlocking systems, and checking the diagnosability of the constructed models allows designers to prevent design inadequacy before testing the interlocking software.

### 4.2.2 Decision Rules of the Controllers and the Coordinator

The following four requests from the traffic control center are considered:

$r_1$ : Reserve route,
$r_2$ : Cancel route,
$r_3$ : Move PM to normal,
$r_4$ : Move PM to reverse.

Let $R$ be the set of these requests. When the coordinator receives a request $r_j \in R$ from the traffic control center, it sends the request $r_j$ to the controllers. Then, the controllers make local decisions $C_{di}(r_j) \in \{1,\ 0\},\ (i=1,\ 2,\ j=1,\ 2,\ 3,\ 4)$, where $C_{di}(r_j)=1$ (respectively, 0) means that the controller $i$ decides to accept (respectively, reject) the request $r_j$. For example, when a PM position request ($r_3$ or $r_4$) is received from the coordinator, each controller checks if the railway block of the PM is occupied, if the PM is moving from one position to another, and if there is any faulty condition or not. The controller rejects the incoming request unless all these criteria are met. That is, the local decision is made as follows:

$$C_{di}(r_j)=\begin{cases}1, & \text{if all safety criteria related with the request are satisfied}\\ 0, & \text{otherwise.}\end{cases} \qquad \textbf{(4.1)}$$

whether all safety criteria are satisfied is verified using the Petri net models in the controller.

The coordinator compares the local decisions of the controllers and gives its final decision to prevent the whole system from falling into a dangerous situation [10]. The final decision is sent from the coordinator to the both controllers, the traffic control center, and the railway field (if necessary). For some of the requests, the coordinator does not demand full agreement of the controllers whereas for some of the requests, the coordinator demands full agreement of the controllers. We first consider a route request ($r_1$) for a non-reserved route. When a route request is received, the coordinator demands full agreement of the controllers to reserve the route. That is, the final decision is made as follows:

$$C_d(r_1)=\begin{cases}1, & \text{if } C_{d1}(r_1)=1 \wedge C_{d2}(r_1)=1,\\ 0, & \text{otherwise.}\end{cases} \quad \textbf{(4.2)}$$

By contrast, for a route cancelation request ($r_2$) after the route is reserved (the route is electronically locked, the color of the entrance signal of the route is not red, and the train has not entered the route yet), the coordinator does not demand full agreement of the controllers because keeping a signal red is safer than any other color indication, and the electronic lock of the route will be released after the cancellation procedure. That is, the final decision is given as

$$C_d(r_2)=\begin{cases}1, & \text{if } C_{d1}(r_2)=1 \vee C_{d2}(r_2)=1,\\ 0, & \text{otherwise.}\end{cases} \quad \textbf{(4.3)}$$

If the cancellation request ($r_2$) is received before the route is reserved (the route is not locked electronically and the color of the entrance signal of the route is still red), then the coordinator demands full agreement like in (4.2).

For a PM position request ($r_3$ or $r_4$), the coordinator checks additional requirements. For example, if the railway block of a PM is occupied, the coordinator does not move the PM even if both controllers agree. The occupation of the railway block is an additional safety check to move a PM to a desired position, and it is represented by a Boolean variable $Oc_{\mathrm{RB}} \in \{1,0\}$ defined as

$$Oc_{\mathrm{RB}}=\begin{cases}1, & \text{if unoccupied,}\\ 0, & \text{if occupied.}\end{cases} \quad \textbf{(4.4)}$$

The final decision for a PM position request ($r_3$ or $r_4$) is made as follows:

$$C_d\left(r_j\right)=\begin{cases}1, & \text{if } C_{d1}\left(r_j\right)=1\wedge C_{d2}\left(r_j\right)=1\wedge Oc_{\mathrm{RB}}=1,\\ 0, & \text{otherwise}\end{cases}\left(j=3,4\right). \tag{4.5}$$

For instance, when the request $r_3$ is received from the traffic control center, the coordinator will accept this request if both controllers accept it and the related railway block is unoccupied. Using the additional check of $Oc_{\mathrm{RB}}$, a kind of derailment shown in Figure 4.3 can be prevented. In general, the coordinator is designed as simple as possible to decrease the possible design faults. We include this additional safety check in the operations of the coordinator since it involves only the AND operation.

The controllers not only make decisions for requests but also perform failure diagnosis. When failure information is received from a controller, the coordinator immediately informs the traffic control center without demanding full agreement and provides railway field safety (e.g. all related signals in the railway field become red and all incoming requests will be rejected).

The local diagnosis decision rule $C_{fi}\left(i=1,2\right)$ of the controllers is defined as a map $C_{fi}:Q_{di}\to\left\{N,U\right\}\bigcup 2^{\Delta_F}$, where for each state $q_{di}\in Q_{di}$ of the diagnoser[38],

$$C_{fi}\left(q_{di}\right)=\begin{cases}\left\{F_j\in\Delta_F:q_{di}\text{ is }F_j\text{-certain}\right\}, & \text{if } q_{di}\text{ is }F_j\text{-certain for some }F_j\in\Delta_F\\ N, & \text{if } l^{ci}=N\text{ for any }\left(M^{ci},l^{ci}\right)\in q_{di}\\ U, & \text{otherwise.}\end{cases} \tag{4.6}$$

Note that *U* is used to indicate that the local diagnosis is unsure whether a fault has occurred or not. Each controller outputs the local diagnosis decision $C_{fi}\left(q_{di}\right)$ when its diagnoser's state is $q_{di}$. The decision fusion rule $C_f$ of the coordinator is also defined as a map $C_f:\left(\left\{N,U\right\}\bigcup 2^{\Delta_F}\right)\times\left(\left\{N,U\right\}\bigcup 2^{\Delta_F}\right)\to\left\{N,U\right\}\bigcup 2^{\Delta_F}$, where

$$C_f\left(D_{f1},D_{f2}\right)=\begin{cases}D_{f1}\bigcup D_{f2}, & \text{if } D_{f1}\in 2^{\Delta_F}\text{ and }D_{f2}\in 2^{\Delta_F}\\ D_{f1}, & \text{if } D_{f1}\in 2^{\Delta_F}\text{ and }D_{f2}\in\left\{N,U\right\}\\ D_{f2}, & \text{if } D_{f1}\in\left\{N,U\right\}\text{ and }D_{f2}\in 2^{\Delta_F}\\ N, & \text{if } D_{f1}=D_{f2}=N\\ U, & \text{otherwise}\end{cases} \tag{4.7}$$

for any local decisions denoted by $D_{f1}$ and $D_{f2} \in \{N, U\} \bigcup 2^{\Delta_F}$. For the detection of a fault, the coordinator does not demand full agreement of the controllers whereas for the decision "normal" (*N*), the coordinator demands full agreement of the controllers for the safety of the system.

As an example, we consider the diagnosers shown in Figure 4.5. The faults $F_1$ and $F_2$ can be detected by the both diagnosers. By contrast, in some case, the fault $F_3$ can be detected by only the diagnoser shown in Figure 4.5a. For example, the diagnoser in Figure 4.5a reaches the state $\{((0,0,\underline{1},\underline{0},0,0,\underline{0},\underline{0},\underline{0},\underline{0}), N)\}$ from its initial state $\{((0,\underline{1},\underline{0},\underline{0},0,0,\underline{0},\underline{0},\underline{0},\underline{0}), N)\}$ by the firing of the observable transition $t_{c1\_3}$ with the observation $\hat{M}1\_1$ of the resulting marking. Similarly, the diagnoser in Figure 4.5b reaches the state $\{((0,0,\underline{1},\underline{0},0,0,\underline{1},\underline{0},\underline{0}), N)\}$ from its initial state $\{((0,\underline{1},\underline{0},\underline{0},0,0,\underline{1},\underline{0},\underline{0}), N)\}$ by the firing of the observable transition $t_{c2\_3}$ with the observation $\hat{M}2\_1$ of the resulting marking. At this state if the diagnoser in Figure 4.5a observes $\hat{M}1\_3$ as the resulting observable marking, the state of the diagnoser becomes $\{((0,0,\underline{0},\underline{0},0,1,\underline{0},\underline{0},\underline{0},\underline{0}), \{F_3\})\}$ whereas the state of the diagnoser in Figure 4.5b remains at $\{((0,0,\underline{1},\underline{0},0,0,\underline{1},\underline{0},\underline{0}), N)\}$. By (4.6), the local decisions of the diagnosers are

$$C_{f1}\left(\{((0,0,\underline{0},\underline{0},0,1,\underline{0},\underline{0},\underline{0},\underline{0}), \{F_3\})\}\right) = \{F_3\}$$

and

$$C_{f2}\left(\{((0,0,\underline{1},\underline{0},0,0,\underline{1},\underline{0},\underline{0}), N)\}\right) = N .$$

Furthermore, by (4.7), the final decision of the coordinator is given as

$$C_f\left(\{F_3\}, N\right) = \{F_3\} .$$

Consequently, the coordinator decides that the fault of the type $F_3$ has occurred.

The failure diagnosis scheme looks similar to Protocol 3 of [82]. However, there are certain differences. Since the controllers are designed by different workgroups, the diagnosers are constructed based on different models of the system to be diagnosed. Furthermore, they receive the same global sensor information from the coordinator. By contrast, in [82], the diagnosers are constructed based on the same system model and they receive different local sensor information.

An example data sequence diagram that illustrates the diagnosis of fault by a single controller and the behavior of the whole is shown in Figure 4.6. According to the data sequence diagram, when stuck of the blades is received from PM position sensors, only the controller 2 detects the PM stuck fault and informs the coordinator of the occurrence of the failure. Then, the coordinator sends this information to the controllers and the traffic control center. If a request is received related with the faulty PM, both controllers reject it.

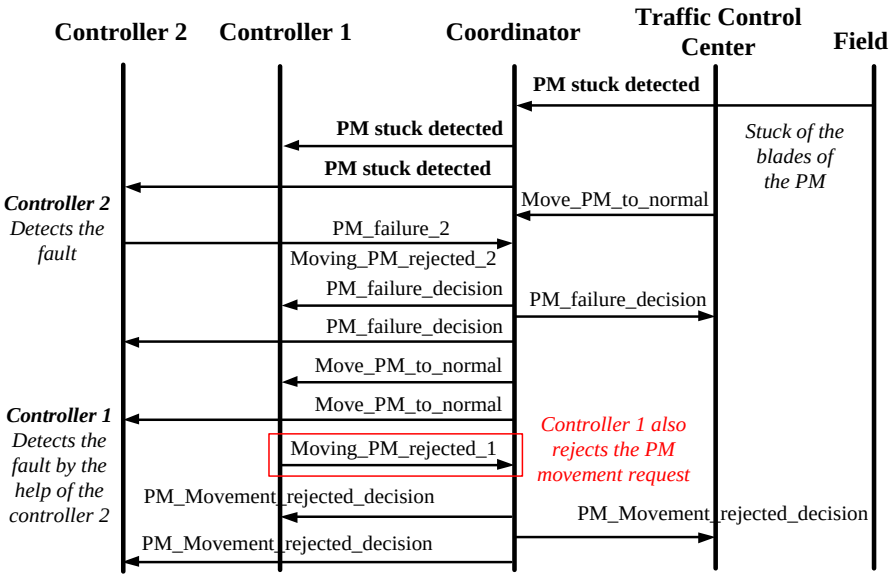


**Figure 4.6 :** Data sequence diagram for the PM stuck fault.

The coordinator logs the useful information such as the request time, the decision of each controller, the situation of the railway field components etc. for retrospective reporting. These reports will then be used by the inspectors in case of any accident.

An advantage of the control architecture and the handshaking process (data flow between the controllers and the coordinator) is to provide the continuity of the

system operation, prevent the system from possible design faults, and provide the safety of the system. However, the reduction in system availability due to synchronization problems [10] is a main disadvantage.

### 4.3 Concluding Remarks

Satisfying the recommended requirements of the railway-related safety standards is a must for railway interlocking system designers. Since the DES modeling methods are highly recommended by the railway-related safety standards, a fixed-block railway signaling system is studied from the DESs point of view. Construction of railway field component models and their diagnosers allows engineers to represent the whole system in a formal way. Although it may seem as a complex and time-consuming task, this process enables us to check the appropriateness of the models before testing the developed software. In this chapter, to meet the requirements of the railway-related safety standards, a control architecture which consists of a coordinator and two controllers designed by different workgroups [10] is studied according to diverse programming. The contribution of the chapter is developing decision making strategies including fault diagnosis for fixed-block railway signaling systems.

## 5. Modeling and Speed Control of Moving-Block Signaling Systems

In this chapter, speed control of two consecutive trains as moving-block is realized in two levels: the modeling level and the control level. To cope with both discrete and continuous behavior of the moving-block signaling system, a Generalized Batches Petri Nets (GBPNs) approach is used for modeling the system whereas a fuzzy logic control method is proposed at the control level.

The term batch in this chapter is used to indicate a group (set) or collection of things (components, vehicles, etc.) of the same kind, which has three characteristics: length, density and position.

### 5.1 Generalized Batches Petri Nets with Controllable Batch Speed

A Generalized Batches Petri Net (GBPN) is defined by [83] as follows:

$$B = \left(PN,\ f, c, Tempo, M_0\right) \tag{5.1}$$

where

- *PN* is a Petri net defined by $PN = \left(P, T, F, W, M_0\right)$ as in Subchapter 3.1. Note that to distinguish the notations of the time and a transition, we use the small letter *t* to denote the time and the capital letter $T_j$ to denote a transition in this chapter. We also denote a place by the capital letter $P_i$.
- $f : P \cup T \to \{D, C, B\}$, called the "batch function", indicates for every node if it is a discrete (*D*), continuous (*C*), or batch (*B*) node.
- $c : \{P_i \in P : f(P_i) = B\} \to \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+$ , called the "characterized batch function", associates three continuous characteristics $(V_i, d_{\max i}, s_i)$, (speed, maximum density, and length) to every batch place $P_i$.
- *Tempo* is a function that associates a rational positive or null number to every transition $T_j$:

- If $f(T_j) = D$, then $Tempo(T_j) = d_j$ is the delay associated with the discrete transition $T_j$, expressed in time unit.
- If $f(T_j) = C$ or $B$, then $Tempo(T_j) = \Phi(T_j) = \Phi_j$ is the maximum firing flow associated with the transition $T_j$, expressed in entities/time unit. To every continuous or batch transition $T_j$, an instantaneous firing flow, denoted by $\varphi_j(t)$, representing the quantity of markings by time unit that fires the transition $T_j$ is also associated.

- $M_0 = M(t_0) = (m_1^0, m_2^0, ..., m_k^0)$ is an initial marking.

Also note that, every batch place must have a continuous place which limits the capacity of the batch place. This capacity is calculated as $C_{\max i} = s_i \cdot d_{\max i}$ [84]. Representation of places and transitions of GBPNs are given in Figure 5.1.

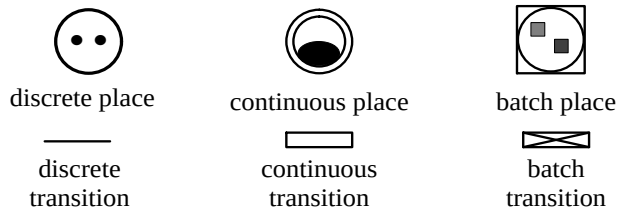


**Figure 5.1 :** Places and transitions of GBPNs.

A marking $M(t)$ at time *t* is denoted as $(m_1(t), m_1(t), ..., m_k(t))$, where

- if $f(P_i) = D$, then $m_i(t) \in \mathbb{Z}^+$: the marking of a discrete place is a nonnegative integer,
- if $f(P_i) = C$, then $m_i(t) \in \mathbb{R}^+$: the marking of a continuous place is a nonnegative real,
- if $f(P_i) = B$, then $m_i(t) = \{\text{ICB1}_i, ..., \text{ICB}k_i, ..., \text{ICB}n_i\}$: the marking of a batch place is a series of $n_i$ batches, where $n_i$ is the number of batches in the place, with $\text{ICB}k(t) = (l_k(t), d_k(t), x_k(t)) \in \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+$, where $l_k$ denotes the length, $d_k$ denotes the density, and $x_k$ denotes the head position, respectively.

Weights of arcs also depend on the types of places. For every place $P_i$ and transition $T_j$,

- if $f(P_i) = D$, then $W(P_i, T_j)$ and $W(T_j, P_i)$ are nonnegative integers, where $W(P_i, T_j)$ is the weight of the arc from place $P_i$ to transition $T_j$, and $W(T_j, P_i)$ is the weight of the arc from transition $T_j$ to place $P_i$, respectively.
- if $f(P_i) = C$ or $f(P_i) = B$, then $W(P_i, T_j)$ and $W(T_j, P_i)$ are nonnegative real numbers.

In addition to these properties defined above, some structural conditions must be added on the structure of a GBPN [83]. Note that

- as given in (3.2), $I(T_j)$ is the set of input places of transition $T_j$,
- as given in (3.3), $O(T_j)$ is the set of output places of transition $T_j$,
- similarly, $I(P_i)$ is the set of input transitions of place $P_i$,
- $O(P_i)$ is the set of output transitions of place $P_i$.

- To satisfy that the marking of a discrete place is an integer independently of the evolution of a GBPN, a loop must exist between a discrete place and a continuous or a batch transition:
  - if $f(P_i) = D$, $f(T_j) \in \{B, C\}$, and $P_i \in I(T_j)$, then $P_i \in O(T_j)$ and $W(P_i, T_j) = W(T_j, P_i)$.
- To avoid conflict structures on batch places, input and output transitions of a batch place are only composed of a single batch transition:
  - if $f(P_i) = B$ then $I(P_i) = \{T_j\}$ and $O(P_i) = \{T_{j'}\}$ for some $T_j$, $T_{j'} \in T$ where $f(T_j) = f(T_{j'}) = B$.

A controllable batch of a batch place $P_i$ at time $t$ which allows controlling the speed of batches moving at different speeds is defined by [36] as follows:

$$CtB_{ki}(t) = (l_k(t), d_k(t), x_k(t), v_k(t)) \in \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \quad \textbf{(5.2)}$$

where $l_k$ is a length, $d_k$ a density, $x_k$ a head position, and $v_k$ a driving speed. The *instantaneous flow* of a controllable batch is defined as $\varphi_k(t) = v_k(t) \cdot d_k(t)$ [36]. Movement of two controllable batches can be seen in Figure 5.2.

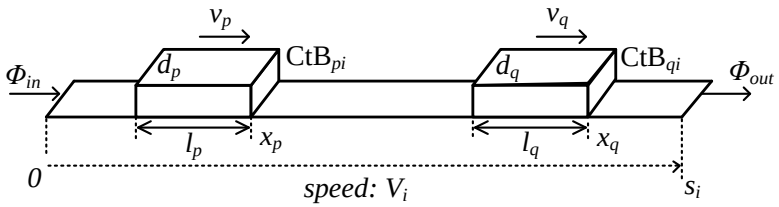


**Figure 5.2 :** Movement of controllable batches.

On the other hand, dynamics of controllable batches is divided into two categories. It is assumed that a controllable batch is in *a free type behavior* if its elements move freely at the driving speed $v_k(t)$ or in *an accumulation behavior* if its elements are not transferred at the driving speed but move according to an output flow which has a lower value than its instantaneous flow.

A batch place has an accumulation at the exit if there exists an accumulated output batch or the output batch is in an accumulated behavior. The evolution rules of a GBPN and explanation of transition firings are given in [83]. More definitions and explanations on basics of Batches PNs can be found in [85]-[87].

In accordance with our objective to make modeling more suitable for trains, the density $(d_k)$ feature in (5.2) is replaced by the acceleration $(a_k)$ feature as follows:

$$CtB_{ki}(t) = \left(l_k, \boxed{a_k(t)}, v_k(t), x_k(t)\right) \in \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \qquad \textbf{(5.3)}$$

where $l_k$ is the length of the train (assumed as a fixed value), $a_k$ is the acceleration of the train, $x_k$ is the head position of the train, $v_k$ is the actual driving speed of the train, subscript $k$ is the train label, and $i$ is the railway line where the train moves, respectively. The characteristics of each batch place $P_i$ are also modified as

$$c(P_i) = \left(V_{permitted\_i}, d_{\max\_i}, s_{length\_i}\right) \qquad \textbf{(5.4)}$$

where $V_{permitted\_i}$ is the permitted speed on the railway line, $d_{\max\_i}$ is the maximum density of trains allowed on the line, and $s_{length\_i}$ is the length of the railway line. The main idea is to provide safe movement of trains by adjusting the speed of the following train depending on the location and the speed of the leading train. Unlike

the batch movements, trains are allowed to move with desired speed values as long as the permitted speed limit is not exceeded and we assume that all trains are moving in a *free type behavior*. Moreover, it is assumed that trains enter the batch place (the main line) with a fixed speed.

### 5.2 Control Architecture and Modeling

The given railway layout is first modeled by a GBPN which is considered as an upper level task. In this level, PMs, signals, and the number of trains that enter the main line are controlled. On the other hand, a lower level control task that realizes acceleration and speed control will be explained in the next subchapter. The control objective is to ensure that the following train must move far enough from the leading train, in other words, the following train must not get closer to the leading train more than the *safe following distance*. The proposed general control architecture is given in Figure 5.3.

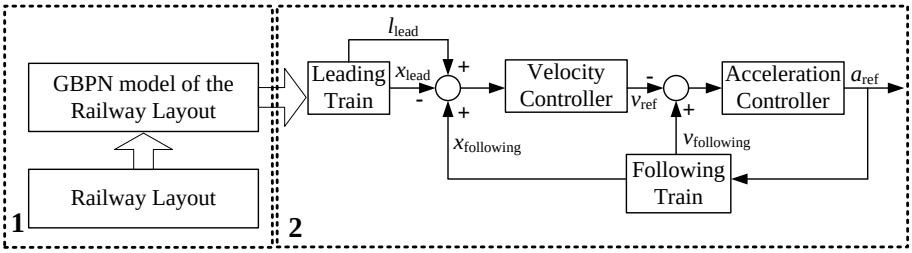


**Figure 5.3 :** The general control architecture.

In this chapter, a railway layout, given in Figure 5.4, with two stations (each station has two platforms) which are connected with a single railway line (main line) is considered. It is assumed that communication between trains and the control center is achieved without any problems such as communication delays or loss, and for simplicity only movements from station A to station B are considered.

Only one train can enter each platform in the stations, so these platforms can be modeled by discrete places. However, the PM region is modeled by a continuous place because trains are not allowed to wait in this region. After the position of the PM is adjusted and locked, trains have to pass this region immediately. Finally, the railway line between two stations is modeled as a batch place. If different MAs will be sent to the trains, more than one batch place can be used for modeling a single line. Note that different MAs mean different permitted speed values.

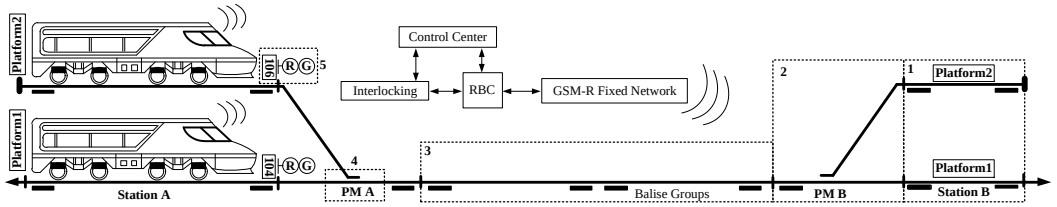


**Figure 5.4 :** Railway layout.

In addition to the railway layout model, components such as signals and PMs are modeled by simple Petri nets. Petri net models of PM A and signal 106 are given in Figure 5.5. Signals of station B are not considered here because only train movements from station A to station B are considered. The word *simple* refers to general logic of operation of components because, as mentioned in Subchapter 3.4, in order to move a PM from one position to another, several conditions such as incoming signals from balises (occupation of railway tracks) and other train movements on the same railway line have to be taken into account.

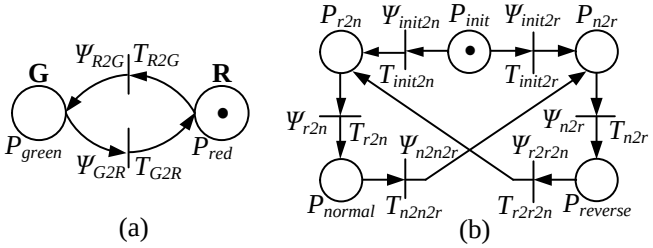


**Figure 5.5 :** Simple Petri net models of two-aspect signal (a) and PM (b).

Similarly, changing the color of a signal from red to green depends on many conditions and safety criteria. A signal is assumed to be on red aspect ($P_{red}$) and if all conditions ($\Psi_{R2G}$) are satisfied, it changes to green aspect ($P_{green}$). A PM is assumed to be in its initial position ($P_{init}$) and, depending on the route, it has to take the normal ($P_{normal}$) or the reverse position ($P_{reverse}$). The model of the railway layout shown in Figure 5.4 is given in Figure 5.6.

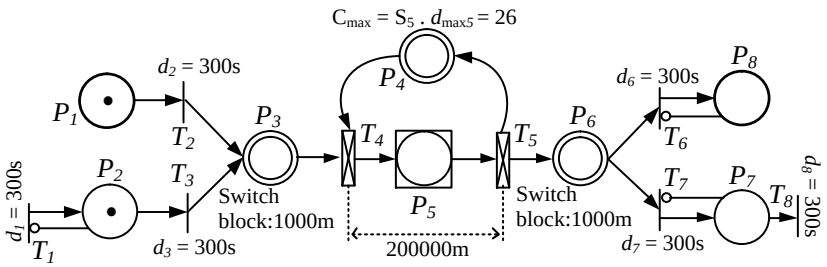


**Figure 5.6 :** BPN model of the railway layout in Figure 5.4.

The platforms of each station are modeled as discrete places $P_1$, $P_2$, $P_7$, and $P_8$. The entrance to platform 1 in station A is realized through $T_1$, whereas the exit from platform 1 in station B is realized through $T_8$. Since platform 2 in each station is a parking place, places $P_1$ and $P_8$ do not have any entrance and exit transition, respectively. The entrance of the trains from platforms to the main line is realized through discrete transitions $T_2$ and $T_3$ and the PM area is modeled as a continuous place $P_3$. For example, in order for a train to move from platform 2 of station A to the main line (from $P_1$ to $P_5$), PM A must be in the reverse position and the signal 106 must be green. After these conditions are satisfied, a train at platform 2 can enter the main line ($P_5$) over the PM area ($P_3$). Multiple trains can enter the main line ($P_5$) with respect to headways, their length, and the length of the main line ($s_{length\_i}$).

The length of the main line and the length of the PM area are assumed as 200km and 1km, respectively. The train length is chosen as 320m. Time required for the entrance of any train to the platforms (or from platforms to the main line) is assumed as 300 seconds ($d_1$, $d_2$, $d_3$, $d_6$, $d_7$, $d_8$). The safe following distance is calculated as approximately 7500m (the braking distance for a HST is determined as 7179m) by considering the definitions given in [72]. 26 trains can enter the main line. Since every batch place has a continuous place that limits the batch capacity as given in Figure 5.6 ($P_4$ limits the capacity of the batch place $P_5$), the limit of the batch place is achieved by continuous place $P_4$ where

$$
\begin{aligned}
d_{\max\_AtoB} &= \frac{C_{\max\_AtoB}}{s_{200\,\mathrm{km}\_AtoB}} = \frac{26}{200} = 0.13\,\mathrm{block/km}, \\
c(P_5) &= (V_{200\,\mathrm{km/h}\_AtoB},\ d_{\max\_AtoB},\ s_{200\,\mathrm{km}\_AtoB}) \\
&= (0.13\,\mathrm{km/h},\ 0.13\,\mathrm{block/km},\ 200\mathrm{km}).
\end{aligned}
\qquad \textbf{(5.5)}
$$

The first level of the control scheme given in Figure 5.3 is achieved using the model given in Figure 5.6. The advantage of using this kind of representation allows expressing the discrete characteristics (entrance and exit of trains to platforms), the continuous characteristics (train movement in the PM area), and the batch characteristics (multiple train movement on the main line which is not allowed in fixed-block systems) in a more formal way. The second level of the control scheme given in Figure 5.3 will be explained in the next subchapter.

### 5.3 Speed Control in Batch Place

While trains are moving from station A to station B, the entrance of the trains into the main line (inside the batch place) can be considered as a speed control problem. The train length, acceleration, speed, and location characteristics are taken into account as in (5.3). The signaling (and so the interlocking) system is responsible for train movements. In this example, the leading train sends its location and speed to the controller and the controller sends a proper acceleration value to the following train. Therefore, the following train calculates its new speed according to this acceleration value.

The main advantage of using a batch place for the main line is that different control methods can be applied for train movements (outside the station). As mentioned above, two trains are considered in this case study. As an initial condition, it is assumed that the leading train is moving with 20 m/s and the following train just entered the main line (the batch place $P_5$) with the 10 m/s initial speed. The distance between the trains is assumed as 3000m at the beginning. Firstly, the locations of the leading train and the following train are compared and a reference speed value is produced by the speed fuzzy logic controller (FLC). Then the actual speed of the following train is compared with this reference speed value and an acceleration value is produced by the acceleration FLC.

Speed and acceleration FLCs are chosen as triangular membership functions whose ranges are between 0-35 m/s and -1.3-1.1 m/s$^2$, respectively. The min-max method is used for rules with the centroid defuzzification method. The scaling factors [88] are chosen as $\alpha_v = 1$, $\beta_v = 0.05$, $\gamma_v = 1.7$, $\alpha_a = 1$, $\beta_a = 0.01$, and $\gamma_a = 1.5$. The block diagram is given in Figure 5.7.

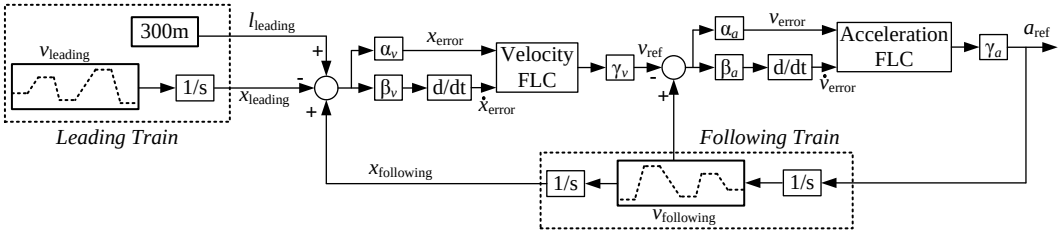


**Figure 5.7 :** Block diagram of fuzzy PD control.

Secondly, another method is applied by tuning the coefficients of the PD controller online. Unlike the previous method, in this scheme, coefficients $K_p$ and $K_d$ of the PD controller are updated by using fuzzy tuning mechanism (FTM). Similarly, speed and acceleration fuzzy tuners are chosen as triangular membership functions and the min-max method is used for rules with the centroid defuzzification method. For velocity tuning, $K_{pv}$ and $K_{dv}$ are adjusted between 0.001-2 and 1-4, respectively, whereas for acceleration tuning, $K_{pa}$ and $K_{da}$ are adjusted between 2-3 and 1-4, respectively. These values are chosen to restrict the settling time and the overshoot (%) of the controller responses. The scaling factors are chosen as $\alpha_v = 1$, $\beta_v = 0.1$, $\gamma_{Kdv} = 0.01$ $\gamma_{Kpv} = 0.007$ , $\alpha_a = 1$ , $\beta_a = 0.05$ , $\gamma_{Kda} = 0.02$ , and $\gamma_{Kpa} = 0.018$ . The block diagram is given in Figure 5.8. Simulation results without measurement noise and disturbances are given in Figure 5.9, Figure 5.10, and Figure 5.11.

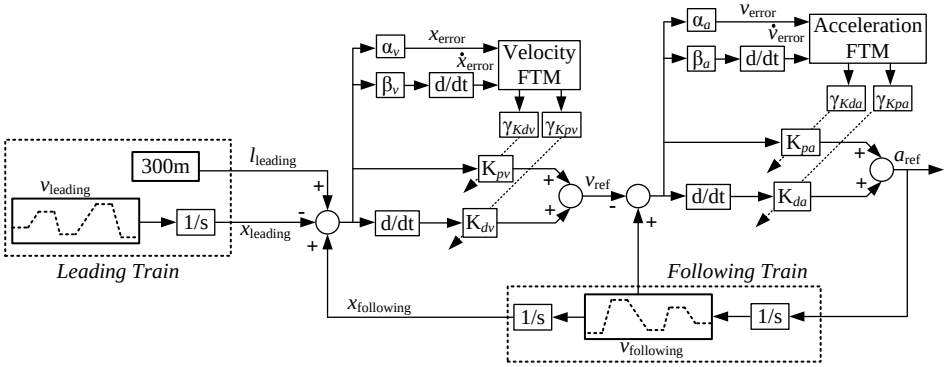


**Figure 5.8 :** Block diagram of the fuzzy PD coefficient tuning case.

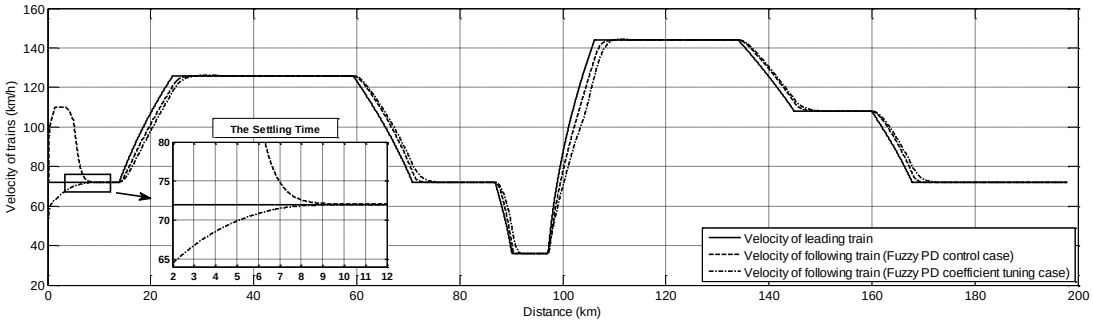


**Figure 5.9 :** Comparison of the controllers: velocity graphs.

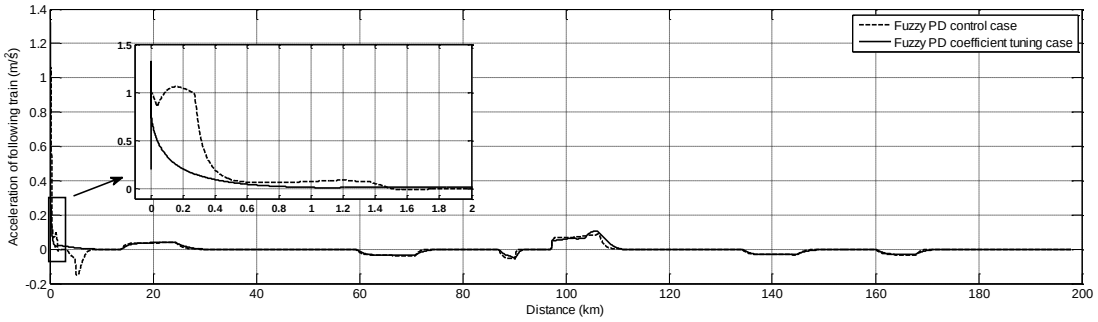


**Figure 5.10 :** Comparison of the controllers: acceleration graphs.

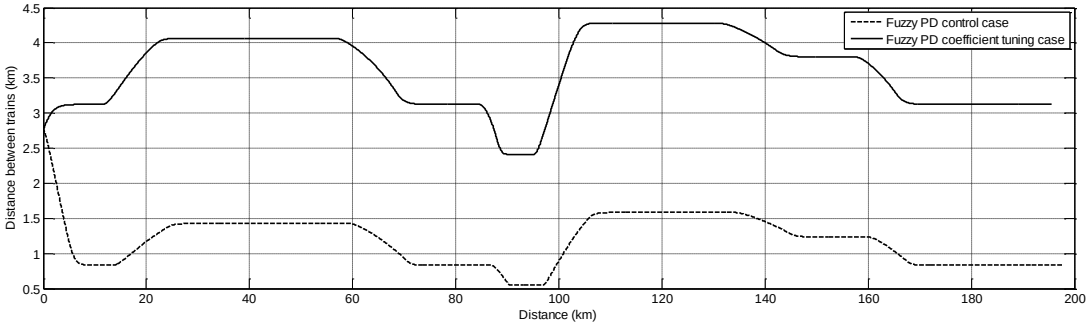


**Figure 5.11 :** Comparison of the controllers: graph of distance between trains.

In Figure 5.9, after the entrance of the following train to the main line, the following train accelerates to catch the leading train and its speed increases up to 30 m/s from 10 m/s while the distance between trains reduces approximately to 900m from 3000m. Later, the following train follows the leading train with a safe following distance. The following distance depends on the speed of the leading train, that is, the distance between the trains gets longer when the speed of the leading train increases and gets shorter when the speed of the leading train decreases. For the velocity graphs given in Figure 5.9, the following train accelerates to reach the leading train without any overshoot in the fuzzy PD coefficient tuning case. The main reason of the overshoot is the acceleration values produced by the controllers (see Figure 5.10). In Figure 5.7, the fuzzy PD controller is used whereas in Figure 5.8, the coefficients of the PD controller are tuned by using a FTM, which permits to tune the coefficients of the PD controller more precisely. But the following distance between trains in the fuzzy PD coefficient tuning case is longer than that in the case of the fuzzy PD control as shown in Figure 5.11. Since the main purpose of this chapter is to use GBPNs as a modeling method and provide a safe transportation by leaving a safe following distance between trains, the trade-off between the settling time and the overshoot is not considered here.

### 5.3.1 Measurement Noise

Next, it is assumed that a measurement noise and step type disturbances are applied to the system in order to evaluate the accurateness of the controllers. Step type disturbances can be encountered when entering or exiting the tunnels. On the other hand, many environmental conditions or other electronic components can cause a measurement noise. The block diagram of the system with a measurement noise and disturbances can be seen in Figure 5.12.

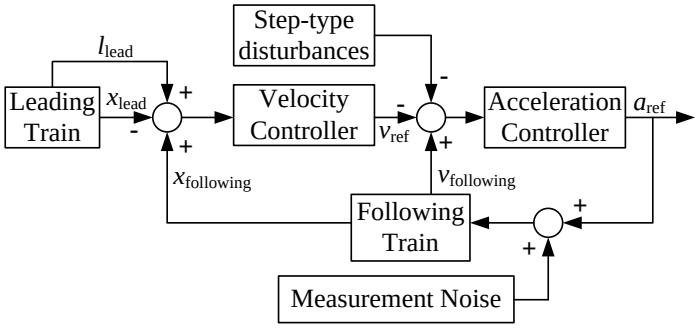


**Figure 5.12 :** The control architecture with measurement noise and disturbance.

Step type disturbances are applied to the system at 41km with 3m/s magnitude, 126km with 2m/s magnitude, and 182km with 5m/s magnitude, respectively. On the other hand, the measurement noise, which is the sum of the white noise and sinus type noises, is shown in Figure 5.13. Simulation results with the measurement noise and disturbances are given in Figure 5.14, Figure 5.15, and Figure 5.16.

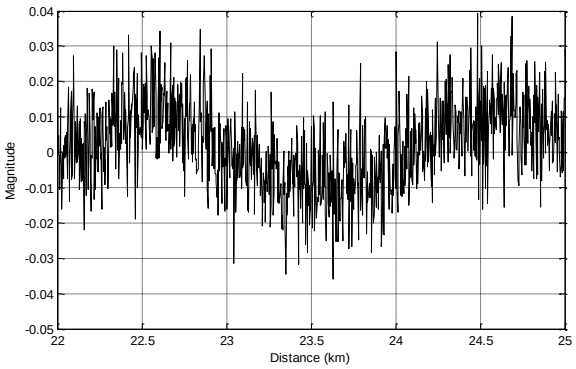


**Figure 5.13 :** Measurement noise.

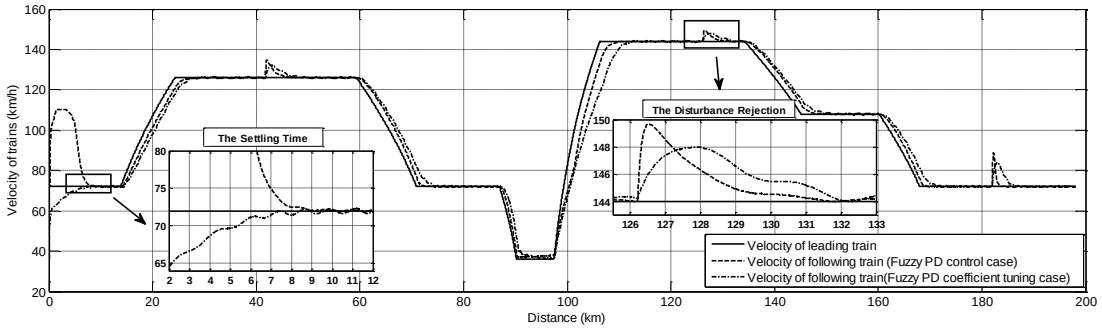


**Figure 5.14 :** Comparison of the controllers: velocity graphs.

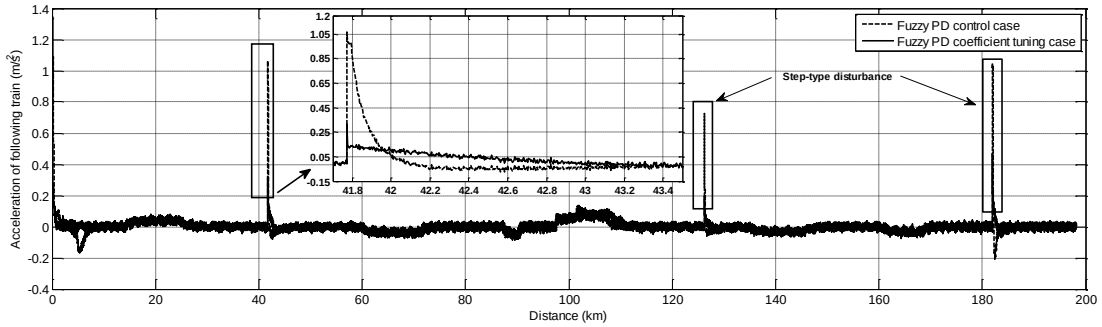


**Figure 5.15 :** Comparison of the controllers: acceleration graphs.

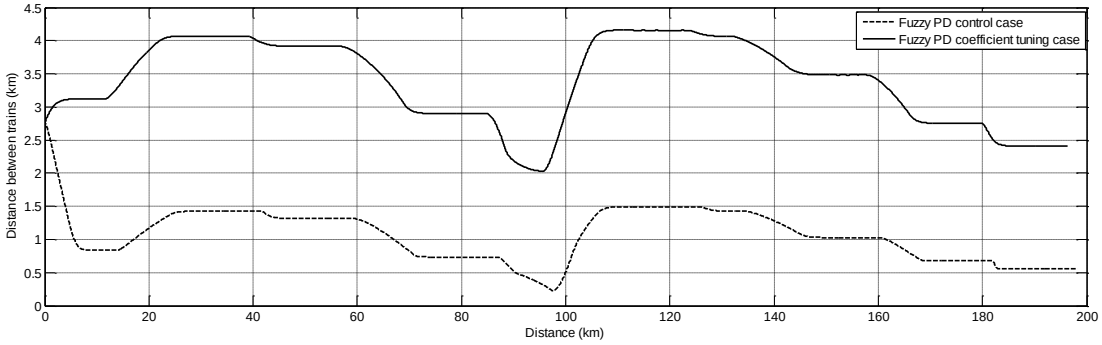


**Figure 5.16 :** Comparison of the controllers: graph of distance between trains.

The effect of the disturbances and the measurement noise is obvious from Figure 5.14 and Figure 5.15, but the controllers overcome all these effects. An unwanted situation occurs in the graph of the distance between the trains when the measurement noise is applied for both controllers.

Unlike Figure 5.11, the following train approaches to the leading train in low velocity values. However, in the fuzzy PD coefficient tuning case, the controller also keeps the following distance at the safe level. Besides, the acceleration value did not increase as the fuzzy PD controller case when the step disturbances are applied as shown in Figure 5.15.

Moreover, in addition to the high overshoot values in the velocity graphs given in Figure 5.9 and Figure 5.14, the settling time in the fuzzy PD controller case is also longer than that in the fuzzy PD coefficient tuning case with a small margin (see Figure 5.9 and Figure 5.14). Besides, the disturbance rejection performance of the fuzzy PD controller is better than that in the fuzzy PD coefficient tuning case.

## 5.4 Concluding Remarks

In this chapter, the batches PNs formalism is used for modeling moving-block train control systems. It is important to note once again that the hybrid structure of the GPBN approach allows to combine both fixed-block and moving-block signaling system characteristics. The movement of trains from platforms to the main line and vice versa has the characteristics of the fixed-block signaling systems whereas the movement of trains on the main line has the characteristics of the moving-block signaling systems. In addition to this, the similarity between moving blocks and batches allows us to adapt batch characteristic to train characteristics (length, location, speed, and acceleration) and allows different speed control schemes in the batch place. This also makes the monitoring of train movements very easy. Moreover, speed control of the following train is realized by using two different control methods.

## 6. Conclusion

Since the railway systems are considered as safety-critical systems and the safety of the transportation and travel mainly relies on the signaling system, the development steps of such safety critical software must be carried out very carefully. The recommendations of the international railway related safety standards and the national rules have to be considered by the signaling system engineers to satisfy the required safety level and fulfill the requirements.

In this thesis, the main concepts of the railway signaling systems that can be divided into two main groups as fixed-block and moving-block railway signaling systems are discussed using their discrete event system (DES) models. The fault diagnosis approach for DESs is applied to the interlocking software development process for fixed-block railway systems. Diagnosability of the system is studied using Petri net models of railway field components. In addition to point machine faults mentioned in [35], the route reservation procedure and faulty conditions in wayside signals are considered, and the diagnosability property is verified based on the necessary and sufficient condition of [26] in the untimed setting. Next, according to the recommendations of the railway-related safety standards, a control architecture including two controllers and a coordinator is studied for fixed-block railway systems. Decision making strategies of the control architecture including fault diagnosis are presented based on the Petri net models.

Both Petri nets and finite state automata are highly recommended as modeling tools in railway related functional safety standards such as EN 50128 [3]. The main reasons for selecting Petri nets instead of finite state automata are the simplicity of their conversion to software blocks and traceability of faults in the developed software [79]-[81].

Diagnosability analysis for fixed-block railway signaling systems is an intermediate step between modeling the system and testing the developed software. Constructing a diagnoser from the Petri net models and checking the diagnosability

of the system can be time-consuming but enables signaling software designers to verify their models before proceeding to the test phase of the developed signaling system software. Just testing the modules according to an automatic testing procedure takes 2~3 weeks even for a medium-sized railway field [4]. Therefore, constructing a diagnoser for verification is useful since it enables us to handle the design errors before the testing phase and shortens the whole software development period mentioned in the V-model. This intermediate step can be realized before passing to the coding step in the V-model.

Finally, consideration of the train movements in a single railway line as moving-blocks is dealt as a speed and acceleration control problem in two levels and the application of fuzzy PD controllers to solve this problem can be considered as another contribution of this thesis. In the first level, a hybrid technique using a Generalized Batches Petri Nets (GBPNs) approach with controllable batch speed [36] is slightly modified and used for modeling the system. In the second level, two fuzzy PD controllers are designed to control velocity and acceleration of the following train. Modeling the moving-block systems by using the GBPNs approach allows us to represent the hybrid (both discrete and continuos) behavior of the moving-block systems in a formal way.

To conclude, the international agreements such as deployment of ERTMS and national research projects of Turkish State Railways and private companies continuously accelerate the development of railway systems in Turkey. From this perspective and the geological location of Turkey which connects the Europe and Asia, it seems that the importance and role of Turkey in the railways will increase day after day. In particular, the completion of the Marmaray tunnel project can be considered as an important sign.

## References


[1] **IEC 61508-7**. (2010). Functional safety of electrical/electronic/programmable electronic safety-related systems, Part 7: Overview of techniques and measures, *European Committee for Electrotechnical Standardization*, Brussels.

[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] **EN 50128.** (2001). Railway Applications, Communications, signalling and processing systems, Software for railway control and protection systems, *European Committee for Electrotechnical Standardization*, Brussels.

[4] **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.

[5] **Denault A.** (2006). Introduction to Software Systems Lecture 18, URL: http://www.cs.mcgill.ca/~adenau/teaching/cs206/lecture18.pdf. date retrieved 13.11.2014.

[6] **Parhami, B.** (1994). Voting Algorithms. *IEEE Transactions on Reliability*, *43*, 617-629.

[7] **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.

[8] **Latif-Shabgahi G.** (2004). A Novel Algorithm for Weighted Average Voting Used in Fault Tolerant Computing Systems. *Microprocessors and Microsystems*, *28*, 357-361.

[9] **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.

[10] **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*,
URL:http://online.journals.tubitak.gov.tr/openAcceptedDocument.htm?fileID=338940&no=72903. Date retrieved 13.11.2014.

[11] **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.

[12] **Oblonsky, J.** (1956). Some Features of the Czechoslovak relay computer SAPO. *Journal of the Nachrichtentechnische Fachberichte*, *4*, 73-75.

[13] **Wensley, J. H. et al.** (1978). SIFT - Design and analysis of a fault-tolerant computer for aircraft control. *Journal of the IEEE Transaction on Computers*, *66*, 1240-1255.

[14] **Avizienis, A. et al.** (1971). The STAR (self-testing and repairing) computer: An incestigation of the theory and practice of fault-tolerant computer design. *Journal of IEEE Transaction on Computers*, *C-20*, 1312-1321.

[15] **Everett, R. R., Zraket, C. A. and Benington, H. D.** (1957). SAGE - A data processing system for air defenses. *Proceedings of the Eastern Joint Computer Conference*, 148-155, Washington, USA.

[16] **Pearson, L.V.** (1973). Moving Block Railway Signalling. *PhD thesis*, Loughborough University of Technology, UK.

[17] **Cassandras, C. G. and Lafortune, S.** (2008). *Introduction to discrete event systems*, 2nd ed., Springer.

[18] **David, R.** (1995). Grafcet: A powerful tool for specification of logic controllers. *IEEE Transactions on Control Systems Technology*, *3(3)*, 253-269.

[19] **Ramadge, P. J. and Wonham, W. M.** (1989). The Control of Discrete Event Systems. *Proceedings of IEEE*, *77*, 81-98.

[20] **Murata, T.** (1989). Petri nets: Properties, Analysis and Applications. *Proceedings of IEEE*, *77*, 541-580.

[21] **Ushio, T., Onishi, I. and Okuda, K.** (1998). Fault detection based on Petri net models with faulty behaviors. *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, 113-118, San Diego, California, USA, 11-14 Oct.

[22] **Ng, K. M., Reaz, M. B. I. and Ali, M. A. M.** (2013). A review on the applications of Petri nets in modeling, analysis, and control of urban traffic, *IEEE Transactions on Intelligent Transportation Systems*, *14(2)*, 858–870.

[23] **Hagalisletto, A. M., Bjork, J., Yu, I. C. and Enger, P.** (2007). Constructing and refining large-scale railway models represented by Petri nets, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *37(4)*, 444–460.

[24] **Kristoffersen, T., Moen, A. and Hansen, H. A.** (2003). Extracting high-level information from Petri nets: a railroad case. *Estonian Academy of Physics and Mathematics*, *52*, 378-393.

[25] **Giua, A. and Seatzu, C.** (2008). Modeling and Superviosry Control of Railway Networks using Petri Nets. *IEEE Transacions on Automation Science and Engineering*, *5*, 431-445.

[26] **Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K. and Teneketzis, D.** (1995). Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, *40(9)*, 1555-1575.

[27] **Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K. and Teneketzis, D.** (1996). Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology, 4(2),* 105-124.

[28] **Sampath, M., Lafortune, S. and Teneketzis, D.** (1998). Active diagnosis of discrete event systems. *IEEE Transactions on Automatic Control, 43(7),* 908-929.

[29] **Chung, S. L.** (2005). Diagnosing PN-based models with partial observable transitions. *International Journal of Computer Integrated Manufacturing, 18 (2-3),* 158-169.

[30] **Wen, Y. L. and Jeng, M. D.** (2004). Diagnosability of Petri nets. *Proceedings of the IEEE International Conference on System, Man and Cybernetics,* 4891-4896, The Hague, Netherlands, 10-13 Oct.

[31] **Ramirez-Trevino, A., Ruiz-Beltran, E., Rivera-Rangel, I. and Lopez-Mellado E.** (2007). Online fault diagnosis of discrete event systems. A Petri net-based approach. *IEEE Transactions on Automation Science and Engineering, 4(1),* 31-39.

[32] **Lefebvre, D. and Delherm, C.** (2007). Diagnosis of DES with Petri net models. *IEEE Transactions on Automation Science and Engineering, 4(1),* 114-118.

[33] **Cabasino, M. P., Giua, A. and Seatzu C.** (2010). Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica, 46(9),* 1531-1539.

[34] **Cabasino, M. P., Giua, A., Lafortune, S. and Seatzu C.** (2012). A new approach for diagnosability analysis of Petri nets using verifier nets. *IEEE Transactions on Automatic Control, 57(12),* 3104-3117.

[35] **Ghazel, M.** (2011). Monitoring and Diagnosis of Discrete Event Systems using Time Petri Nets: A Railway case study. *Fault Detection: Theory, Methods and Systems,* 69-95.

[36] **Demongodin, I.** (2009). Modelling and analysis of transportation networks using batches Petri nets with controllable batch speed, *Applications and theory of Petri Nets, Lecture Notes in Computer Science, 5606,* 204-222.

[37] **Durmuş, M. S., Takai, S. and Söylemez, M. T.** (2014). Fault Diagnosis in Fixed-Block Railway Signaling Systems: A Discrete Event Systems Approach. *IEEJ Transactions on Electrical and Electronic Engineering, 9(5),* 523-531.

[38] **Durmuş, M. S., Takai, S. and Söylemez, M. T.** (***in press***). Decision making strategies in fixed-block railway signaling systems: A discrete event systems approach; *IEEJ Transactions on Electrical and Electronic Engineering, 10(2).*

[39] **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, 6(6),* 403-410.

[40] **Hall, S.** (2001). *Modern Signalling Handbook,* Ian Allan Publishing, England.

[41] **White, C.** (2010). Interlocking principles. *Notes of IET Professional Development Course on Railway Signalling and Control Systems*, 65-78.

[42] **Clark, S.** (2010). A history of railway signalling: From the bobby to the balise. *Proceedings of IET Professional Development Course on Railway Signalling and Control Systems*, 7-20, Birmingham, UK, 7-11 June.

[43] **Bonnet C.F.**, (1996). *Practical Railway Engineering*, Imperial College Press, ISBN 1-86094-012-9.

[44] **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.

[45] **Petersen, J. L.** (1998). Automatic Verification of Railway Interlocking Systems: A Case Study. *Proceedings. of the 2nd Workshop on Formal Methods in Software Practice*, 1-6, Clearwater Beach, FL, USA, 04-05 March.

[46] **Kantz, H. and Koza C.** (1995). The ELEKTRA Railway Signalling-System: Field Experience with an Actively Replicated System with Diversity. *Proceedings of the 25th International Symposium on Fault-Tolerant Computing*, 453-458, Pasadena, CA, USA, 27-30 June.

[47] **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.

[48] **Calvert, J.B.** (2009). URL: http://mysite.du.edu/~jcalvert/railway/ctc.htm. date retrieved 13.11.2014.

[49] **Trains of Turkey**. (2012). URL: http://www.trainsofturkey.com. date retrieved 13.11.2014.

[50] **URL-1** <*http://www.railsigns.uk/home.html*>. date retrieved 13.11.2014.

[51] **Yıldırım, U., Durmuş, M. S. and Söylemez, M. T.** (2012). Automatic Interlocking Table Generation for Railway Stations using Symbolic Algebra. *Proceedings of the 13th IFAC Symposium on Control in Transportation Systems*, 171-176, Sofia, Bulgaria, 12-14 September.

[52] **Palmer J.W.** (2010). The need for train detection. *Proceedings of the 11th IET Professional Development Course on Railway Signalling and Control Systems*, 52-64, York, UK, 5-9 June.

[53] **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.

[54] **Knight, J.C.** (2002). Safety critical systems: challenges and directions," Software Engineering. *Proceedings of the 24rd International Conference on Software Engineering*, 547-550, Orlando, FL, USA 23-25 May.

[55] **Kuepper, G. J.** (1999). 150 years of train-disasters - practical approaches for emergency responders. *9-1-1 Magazine*, issue. *September/October*, 30-33.

[56] **IEC 61508-1.** (2010). Functional Safety of Electrical/Electronic/Programmable electronic safety-related systems, Part 1: General requirements, *European Committee for Electrotechnical Standardization*, Brussels.

[57] **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.

[58] **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*, 1683-1692, Stavanger, Norway, 30 May-01 June.

[59] **Kaymakçı, Ö., Anık, V. G. and Üstoğlu, İ.** (2010). A local modular supervisory controller for a real signalling system. *Proceedings of the 5th IET International System Safety Conference*, 1-6, Manchester, UK, 18-20 October.

[60] **Leveque, O.** (2008). *ETCS Implementation Handbook*, ver. 2.1, Union of Railways.

[61] **Pascoe, R. D. and Eichorn, T. N.** (2009). What is communication-based train control?, *IEEE Vehicular Technology Magazine, 4(4),* 16-21.

[62] **URL-2** <*http://www.ertms.net/?page_id=55*>. date retrieved 13.11.2014.

[63] **Lockyear, M. J.** (1996). Changing track: Moving-block railway signaling. *IEE Reviews, 42,* 21-25.

[64] **UNISIG SUBSET-036.** (2012). UNISIG, FFFIS for Balise, ver.3.0.0, *European Railway Agency.*

[65] **ERA-ERTMS-015560.** (2009). ERTMS/ETCS Driver Machine Interface, Informative Specification Document, ver.2.3, *European Railway Agency.*

[66] **Sandidzadeh, M. A., Heydari, A. and Khodadadi, A.** (2012). Genetic algorithm and particle swarm optimization algorithm for speed error reduction in railway signaling systems. *International Journal of Adaptive Control and Signal Processing, 27,* 478-487.

[67] **URL-3** <*http://www.uic.org/spip.php?article631*>. date retrieved 13.11.2014.

[68] **URL-4** <*http://www.uic.org/IMG/pdf/gsm-r_guide.pdf*>. date retrieved 13.11.2014.

[69] **Vincze, B. and Tarnai, G.** (2006). Development and Analysis of Train Brake Curve Calculation Methods with Complex Simulation. *Advances in Electrical and Electronics Engineering, 5(1-2),* 174-177.

[70] **Barney, D., Haley, D. and Nikandros, G.** (2001). Calculating Train Braking Distance. *Proceedings of the 6th Australian Workshop on Safety Critical Systems and Software*, 23-29, Brisbane, Australia.

[71] **Zimmermann, A. and Hommel, G.** (2003). A Train Control System Case Study in Model-Based Real Time System Design. *Proceedings of the International Parallel and Distributed Processing Symposium*, 118b, 22-26 April.

[72] **ERA-ERTMS-040026.** (2012). Introduction to ETCS Braking Curves, Information Document, ver.1.2, *European Railway Agency*.

[73] **Abed, S. K.** (2010). European Rail Traffic Management System - An Overview. *Iraq Journal of Electrical and Electronic Engineering*, *6(2)*, 172-179.

[74] **Wei, S., Bai-gen, C., Jing-jing, W. and Jian, W.** (2010). Research and Analysis of ETCS Controlling Curves Model. *Proceedings of the 3rd International Conference on Advances Computer Theory and Engineering*, V2-178-V2-181, Chengdu, China, 20-22 August.

[75] **Booth, P. D.** (2010). Intermittent and Continuous Automatic Train Protection. *Proceedings of the IET Professional Development Course on Railway Signalling and Control Systems*, 89-117, York, UK, 5-9 June.

[76] **IEEE 1698-2009.** (2009). IEEE Guide for the Calculation of Braking Distance for Rail Transit Vehicles. *IEEE Vehicular Technology Society*.

[77] **Ping, L. K., You, G. Z. and Hua, M. B.** (2007). Energy-Optimal Control Model for Train Movements. *Chinese Physics*, *16*, 359-364.

[78] **Li, Z. W., Zhou, M. C. and Wu, N. Q.** (2008). A survey and comparison of Petri-net based deadlock prevention policies for flexible manufacturing systems. *IEEE* Transactions *on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 38(2),* 173-188.

[79] **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.

[80] **Thapa, D., Dangol, S. and Wang, G. N.** (2005). Transformation from Petri nets model to programmable logic controller using one-to-one mapping technique. *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, 228-233, Vienna, Austria, 28-30 November.

[81] **Frey, G.** (2000). Automatic implementation of Petri net based control algorithms on PLC. *Proceedings of the 2000 American Control Conference*, 2819-2823, Chicago, IL, USA, 28-30 June.

[82] **Debouk, R., Lafortune, S. and Tenektzis, D.** (2000). Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications*, *10(1-2)*, 33-86.

[83] **Demongodin, I.** (2001). Generalized batches Petri nets: Hybrid model for high speed systems with variable delays, *Discrete Event Dynamic Systems: Theory and Applications*, *11(1-2)*, 137-162.

[84] **Demongodin, I., Audry, N. and Prunet, F.** (1993). Batches Petri nets, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 607-617, Le Touquet, France, 17-20 Oct.

[85] **Audry, N. and Prunet, F.** (1995). Controlled batches Petri nets, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1849-1854, Vancouver, BC, 22-25 Oct.

[86] **Demongodin, I., Caradec, M. and Prunet, F.** (1998). Fundamental concepts of analysis in batches Petri nets, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, San diego, 845-850, California, USA, 11-14 Oct.

[87] **Demongodin, I.** (1999). Extended structures of batches Petri nets, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 182-187, Tokyo, Japan, 12-15 Oct.

[88] **Qiao, W. Z. and Mizumoto, M.** (1996). PID type fuzzy controller and parameters adaptive method, *Fuzzy Sets and Systems, 78*, 23-35.

## List of Publications

### Publications Related with the Work at Osaka University

### International 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. **Durmuş, M. S.**, Takai, S. and Söylemez, M. T. (2014). Fault Diagnosis in Fixed-Block Railway Signaling Systems: A Discrete Event Systems Approach, *IEEJ Transactions on Electrical and Electronic Engineering*, Vol. ***9***, No. 5, 523-531.

3. **Durmuş, M. S.**, Takai, S. and Söylemez, M. T. (***in press***). Decision Making Strategies in Fixed-Block Railway Signaling Systems: A Discrete Event Systems Approach, *IEEJ Transactions on Electrical and Electronic Engineering*, Vol. ***10***, No. 2, 2015.

### Domestic Conference Proceedings (In Turkish):

1. **Durmuş, M. S.**, Takai S. and Söylemez, M. T. (2014). Sabit-Blok Demiryolu Sinyalizasyon Sistemlerinde Karar Verme Yöntemleri: Ayrık Olay Sistem Yaklaşımı. In TOK'14. *Proceedings of the Turkish National Meeting on Automatic Contro*l, 372-378, Kocaeli University, Kocaeli, Turkey, 11-13 September.

2. **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 Contro*l, 738-744, Inonu University, Malatya, Turkey, 26-28 September.

### Other Publications

### International Journal Papers:

1. 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*.

2. **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*.

**Domestic 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.

2. **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.

3. **Durmuş, M. S.**, Yıldırım, U., Eriş, O., Özdeş, O. and Söylemez, M. T. (2012). Sabit-Blok Demiryolu Sistemleri için Güvenli Anklaşman Yazılımı Tasarımı, *Automation Journal*, Vol. ***238***, April 2012, issue. 4, 324-332.

**International Conference Papers:**

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*, 55-59, 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*, 7-12, 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*, 83-86, 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*, 90-98, 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*, 180-185, 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*, 70-74, 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*, 110-115, 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*, 165-170, 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*, 177-180, 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*, 171-176, 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*, 723-726, 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, II*, 386-390, 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*, 10752-10757, 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*, 1090-1095, 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*, 1096-1101, 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*, 1683-1692, 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*, 471-480, 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*, 461-470, 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*, 337-342, Berlin Technical University, Berlin, Germany, 30 August-1 September.

20. **Durmuş, M. S.**, Akın, 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*, 508-512, 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*, 171-176, 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*, 1558-1563, 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'09 *Proceedings of the International Symposium on INnovations in Intelligent SysTems and Applications*, 5 pages, Karadeniz Technical University, Trabzon, Turkey, 29 June-01 July, 12-16.

**Domestic Conference Proceedings (In Turkish):**

1. **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 Contro*l, 360-364, Niğde University, Niğde, Turkey, 11-13 October.

2. **Durmuş, M. S.**, Yıldırım, U., Üstoğlu, İ., Kaymakçı, O. and Akçil, L. (2012). Kent İçi Raylı Ulaşımda Sabit-Blok Sinyalizasyon Sistemi Tasarımı: Esenler Depo Sahası. In TOK'12. *Proceedings of the Turkish National Meeting on Automatic Contro*l, 345-348, Niğde University, Niğde, Turkey, 11-13 October.

3. **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 Contro*l, 146-151, Dokuz Eylül University, İzmir, Turkey, 14-16 September.

4. **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*, 5 pages, Bursa-Eskişehir, Turkey, 7-9 April.

5. Yıldırım, U., **Durmuş, M. S.** and Söylemez, M. T. (2011). Demiryolu Sinyalizasyon Sistemleri için Otomatik Anklaşman Tablosu Oluşturulması. In EUSİS 2011. *Proceedings of the Electrical Transportation Systems Symposium*, 6 pages, Bursa-Eskişehir, Turkey, 7-9 April.

6. Yıldırım, U., **Durmuş, M. S.** and Söylemez, M. T. (2010). Anklaşman Tasarımı için Petri Ağı Denetçilerinin Otomatik Oluşturulması. In ELECO 2010. *Proceedings of the Conference on Electrical and Electronics Engineering*, 197-201, Bursa, Turkey, 2-5 December.

7. Yıldırım, U., **Durmuş, M. S.**, Kurşun, A. and Söylemez, M. T. (2010). Demiryolu Hemzemin Geçitleri için Hatada-Güvenli Sinyalizasyon ve Anklaşman Tasarımı. In TOK'10. *Proceedings of the Turkish National Meeting on Automatic Contro*l, 235-240, Gebze Yüksek Teknoloji Enstitüsü, Gebze, Turkey, 21-23 September.

8. 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 Contro*l, 241-245, Gebze Institute of Technology, Gebze, Turkey, 21-23 September.

9. Saygın, S., Yakın, İ., **Durmuş, M. S.** and Söylemez, M. T. (2009). Petri Ağları ile Demiryolu Makas Bölgelerinin Anklaşman ve Sinyalizasyonu Tasarımı. In TOK'09. *Proceedings of the Turkish National Meeting on Automatic Contro*l, 7 pages, Yıldız Technical University, İstanbul, Turkey, 13-16 October.

10. **Durmuş, M. S.** and Söylemez, M. T. (2008). Petri Ağları ile Demiryolu Anklaşman ve Sinyalizasyon Tasarımı. In ELECO 2008. *Proceedings of the Conference on Electrical and Electronics Engineering*, 447-451, Bursa, Turkey, 26-30 November.

11. **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 Contro*l, 365-369, Istanbul Technical University, İstanbul, Turkey, 13-15 November.

12. **Durmuş, M. S.** and İplikçi, S. (2007). Veri Kümeleme Algoritmalarının Performansları Üzerine Karşılaştırmalı Bir Çalışma. In AB'07. *Proceedings of the Academic Informatics Conference*, 393-400, Dumlupınar University, Kütahya, Turkey, 31 January-2 February.