Evaluation of Railway System Performance under changing Levels of Automation using a Simulation Framework

By

Krishnan Guruvayur Venkateswaran

A Thesis submitted to the University of Birmingham for the Degree of Doctor of Philosophy

THE UNIVERSITY OF BIRMINGHAM

Birmingham Centre for Railway Research and Education

School of Electronic, Electrical and Systems Engineering

The University of Birmingham

September 2019

Abstract

Modern mainline railways are under constant pressure to meet the demands of higher capacity and to improve their punctuality. Railway system designers and operators are increasingly looking to use automation as tool to enable proactive strategies to optimise the timetable, improve the reliability of the infrastructure & rolling stock, to allow for a more dynamic command & control system which can respond to passenger demand and overall to linearize the response behaviour of the system under duress.

In the first part of this thesis, I, the author, will discuss the development of automation over the years and the techniques that have been developed to analyse automation changes in a system. Further to this, I outline the various changes to the railway technology over the last century in brief.

In the second part, I apply the techniques described earlier to design an automation framework to develop a grade of automation for the railway system to meet the demands of improved capacity and performance. Further to this, I develop parallel testable levels of automation using existing railway technology to demonstrate the effectiveness of a framework developed using the methodology discussed before. These levels are then tested on a network topology using micro-simulation to verify if they produce improved capacity and performance.

In the final part, A case study is developed for the network from Kings Cross station to Welwyn Garden on the East Coast Main Line with the traffic dense branch line from Hertford north joining this line. The network is simulated under similar conditions to that adopted for the theoretical network and the results are compared with the previous outcomes.

Results from the above studies have several significant outcomes. Firstly, the methodology developed over the course of this thesis can produce automation levels that are distinct from each other. Secondly, these simulation results show that there is a step change in the performance of the systems when organised into distinct levels of automation. Thirdly, and perhaps the most important conclusion from the studies, I show that automation of a single railway sub-system does not yield beneficial results unless there are complementary solutions produced for the surrounding sub-systems.

In the theoretical phase of the study, the journey time calculations were repeated for 5000 iterations using a Quasi Monte Carlo framework. The results indicate a clear separation between each of the level and stages of automation proposed within the framework. The results from the simulation show that the reduction in journey times between the various levels can be as much as 5%. In the case study, the results were not as distinct but the overall trendlines indicate a reduction in journey times for both intercity and suburban services.

Publications produced during the research period:

- Venkateswaran, K., Nicholson, G., Chen, L. & Pelligrini, P. 2017. D3.3.2 Analysis of European best practices and levels of automation for traffic management under large disruptions *In:* IFFSTAR (ed.) *Capacity for Rail.* UIC.
- Venkateswaran, K. G., Nicholson, G. L., Roberts, C. & Stone, R. Impact of Automation on the Capacity of a Mainline Railway: A Preliminary Hypothesis and Methodology. 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pages 2097-2102.

Acknowledgements

This thesis signifies the end of the first phase of my research career. At the beginning of my time here as a researcher, I thought that a research study was a single person affair. I am glad that over the course of this study I stand truly corrected regarding my earlier view.

Over the course of this thesis many people have been kind enough to spend some time talking to me regarding my work at meetings, conferences, seminars or even over a cup of coffee. I would like to extend my thanks to all my colleagues and friends for their kindness, support and patience over the years.

I would like to thank the Engineering Physical Sciences Research Council (EPSRC) for funding my PhD. I would also like to thank Network Rail for supporting my PhD financially. A special thanks to Dr Meena Dasigi at Network Rail, whose regular insights helped to shape the contents of this thesis to be relevant to the questions being asked in the railway industry.

I would like to thank my supervisors, Prof. Clive Roberts and Prof. Bob Stone who have been essential support during the research and write up period. I would like to thank them for all the guidance I received from them, in the form of professional advice and support. I would be remiss if I fail to mention their backroom support in the down days when things are not working quite as well. I would like to specifically thank Prof. Felix Schmid for mentoring me over the years and without whose input my thesis would stand incomplete.

Lastly, I would like to thank my family for their support and encouragement over the years. My parents have always encouraged me to be curious and helped me be the person I am. My mother, Gowri V., for teaching me the importance of being industrious at ones work. My Father, Venkateswaran G. K, for teaching me the essence of doing work consistently. My (younger) brother, Hariharan G. V, for his fantastic support over the years to me and my parents, especially when I have been away in England for the last 8 years. Lastly, to my dear wife, Dr. Lavanyaa Manjunatha, for her patience and love extended to me as a fellow academic.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
ARS	Automatic Route Setting
ATC	Automatic Train Control
ATO	Automatic Train Operation
ATOC	Association of Train Operating Companies
ATP	Automatic Train Protection
AWS	Automatic Warning Systems
bot	Machine with the characteristics of a robot
BRaVE	Birmingham Railway Virtual Environment
BSI	British Standard Institute
CaSL	Cancellations and Significant Lateness
CBTC	Communications Based Train Control
CCC(S)	Command Control and Communication (System)
CCS	Command and Control System
CTC	Centralised Traffic Control
CUI	Capacity Utilisation Index
DfT	Department for Transport
DWF	Door Width Factor
EEPROM	Electrically Erasable Programmable Read Only Memory
EMU	Electric Multiple Unit
ERTMS	European Railway Traffic Management System
ETCS	European Train Control System
EU	European Union
FT	Framework Testing
GB	Great Britain
GSM-R	Global System for Mobile Communications – Rail
HET%	Heterogeneity Percentage
HIL	Hardware In the Loop
IECC	Integrated Electronic Control Centre
IEEE	Institution of Electrical and Electronic Engineers
KXW	Kings Cross to Welwyn Garden City
LTE	Long Term Evolution (Radio System)
MC	Monte Carlo
NDAS	Networked Driver Advisory System
NR	Network Rail
OHLE	Overhead Line Equipment
ORR	Office of Road and Rail
PPM	Public Performance Measure

QMC	Quasi-Monte Carlo
RBC	Radio Block Centre
RSSB	Railway Safety Standards Board
SAHR	Sum of Arrival Headway Reciprocals
SDAS	Standalone Driver Advisory System
SPAD	Signal Passed at Danger
SSI	Solid State Interlocking
SSHR	Shortest Sum of Headway Reciprocals
TMS	Traffic Management System
TPS	Train Protection System
TPWS	Train Protection and Warning System
TS(S)	Train Stop (System)
UIC	International Union of Railways *(Union Internationale des Chemins de Fer)*
UITP	International Association of Public Transport *(Union Internationale des Transports Publics)*
UK	United Kingdom (GB and Northern Ireland)

1 Introduction

The progress of the human civilisation closely parallels the efficiencies gained in the energy required to do work. This has been possible through the development of tools, systems, mechanical arrangements, and machines.

Each age of human progress can be credited to the invention, application and development of machines. The earliest flint tools made cutting and hunting easier, the development of the axle that made trade possible, the plough revolutionised agriculture, the development of steam engines made the industrial age a reality and with the most recent application of silicon transistors that have enabled the growth of the digital age.

Initially, machines were introduced as aids to the human in reducing the energy and time required to do a task. Over time, machines have integrated into the workflow to perform sequential tasks. Tasks that were considered mundane and boring due to their repetitive nature can be now assigned to machines.

The computing age has enabled machines to transcend the physical solitary unit and enabled them to perform highly complex tasks which require social interactions. The added layer of social interaction between the digital machines (bots) has led to the development of automaton that can individually either sense the world or analyse the stimulus from the environment or action responses to complete tasks.

Automation can then be defined as the application of these automatons to do work. All modern fields of work such as research, medicine, manufacturing, business and service are looking to adopt automation. The goals of introducing automation are varied within each of these sectors but chiefly it reduces wastage, increase efficiencies and improve the overall safety of the system.

1.1 Background

The transportation sector is adopting automation as a means of increasing system efficiency and also to improve the safety of the system by minimising human error. The aviation sector was the first to adopt automation as a mean of providing an independent system to fly a plane parallel to the human pilots. Both rail and road have been increasingly looking to automation as means of improving their respective system efficiencies so as to meet the ever greater demand for transportation.

The railway sector has witnessed contrasting attitudes towards adopting automation. Urban/metropolitan railways (metro) have welcomed automation. Cities have grown to cover vast areas and are still growing. This has driven demand for transportation into the city for work, business or pleasure. Metro systems are required to provide services at very short time intervals (termed as headway) between trains. As such, each part of the service is critical to the operation. At the same time, it is important to note that the increased demand for service places a pressure on the system may compromise the integral safety of the systems. Automation of the system is seen as a method to control the system performance variance and to reduce the probability of triggering conditions that may result in an unsafe state of operation. Major cities such as London, Dubai, Copenhagen, Paris and Beijing have used automation to develop fully autonomous metros to meet the demand and to improve passenger safety. Figure 1 shows the metros around the world that have adopted the Unattended Train Operation according to UITP.

Figure 1: Adoption of unattended train operation in large cities around the world (Metros, 2016)

Contrastingly, although the demand for rail connections between cities and suburban areas around cities has increased, mainline railways still require huge amounts of human intervention and have not seen much interest in adopting higher levels of automation.

In the UK rail context, delays on the mainline are reported through two metrics. The first metric is the Public Performance Measure (PPM), which is used to report the punctuality of services on the mainline. In the UK, a railway service on the mainline is reported to be on time if it arrives within 5 minutes of the published timetable arrival time. The second metric is Cancellations and Significant Lateness (CaSL) is used to report services that have been cancelled, due to technical or operational issues, or services that have been significantly delayed while in service across the network.

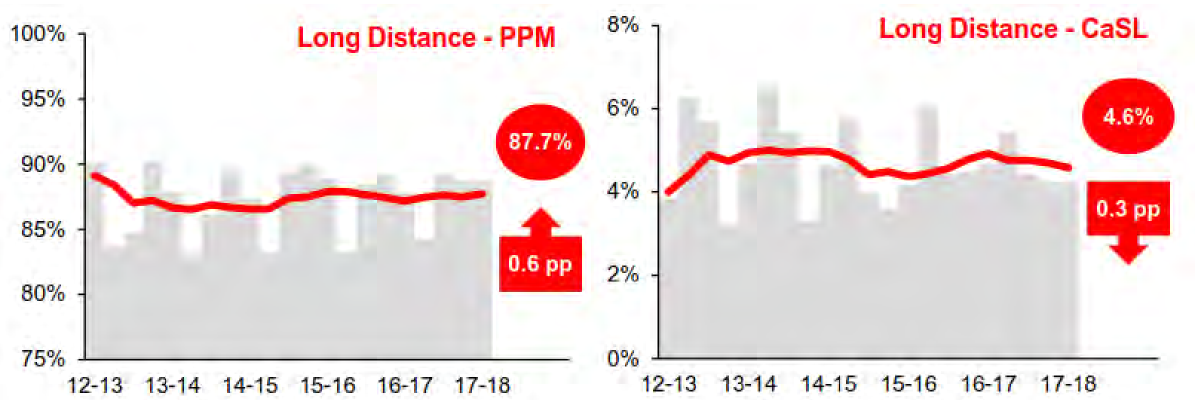


Figure 2: PPM and CaSL, Long Distance Sector, 2012-13 Q2 to 2017-18 Q2 (ORR, 2017)

Figure 2 shows the annual delay data across the UK mainline. The PPM is at 87.7% which is significantly below the 92% target set by the DfT for the UK rail network. The graph on the right in Figure 2 indicates CaSL of 4.6%. The CaSL metric is an indicator of the reliability of the rail sector. Independent industry reports by DfT, RSSB, ATOC and Network Rail all highlight the need to employ automation to reduce the unreliability and delays on the railway network.

1.2 Objectives

The railway system consists of subsystems that contribute towards carrying passengers or goods (rolling stock), detecting and transmitting instructions to the train (train management), management of the infrastructure such as track, over-head equipment, switches & crossings and stations and platforms where passengers board and alight a train.

Individually these subsystems can be modelled and automated but the interaction between the subsystems produces variability within the whole system. In metro networks, automation of subsystems is straightforward as these interactions can be controlled. This is because the whole network is designed to carry only passenger traffic, with similar rolling stock, with a set service pattern and all enclosed within a restricted geographic area (city). UIC has produced standard guidance of urban automated transport systems.

In the case of mainline railway systems, producing a framework for automation is more complicated. Mainline railway networks are designed to carry passenger and freight traffic, each service can be delivered using various rolling stock. The networks consist of technologies from different eras interacting with each other. Introducing automation onto railway mainline may lead to undesirable loss of capacity and more significantly the loss of safety. Therefore, the introduction of automation into individual subsystems needs to be studied and the interaction between the subsystem and its interfaces, either with other subsystems or external actors, needs to be carefully modelled.

The objective of this thesis is to propose a framework to study the introduction of automation into individual subsystems. The thesis has the following aims

1. To study the impact of automation on a system, user and its environment,
2. Develop a framework that aids in defining the current state of automation within the system,
3. Using simulation to test and verify the framework developed,
4. Test the framework with a real world case study.

While it would be great to present an exhaustive study to answer to the above questions, the scope of the work presented within this thesis must be restricted with time and energy in mind. The following assumptions have been made

1. Each subsystem will be changed such that subsystems cannot have differing technologies. Such subsystems would require a formal expression and a subsequent modelling in simulation to study the effects.
2. The response from the machine is considered to have a uniform distribution within the simulation.

1.3 Thesis Structure

The thesis structure is shown in Figure 3 below

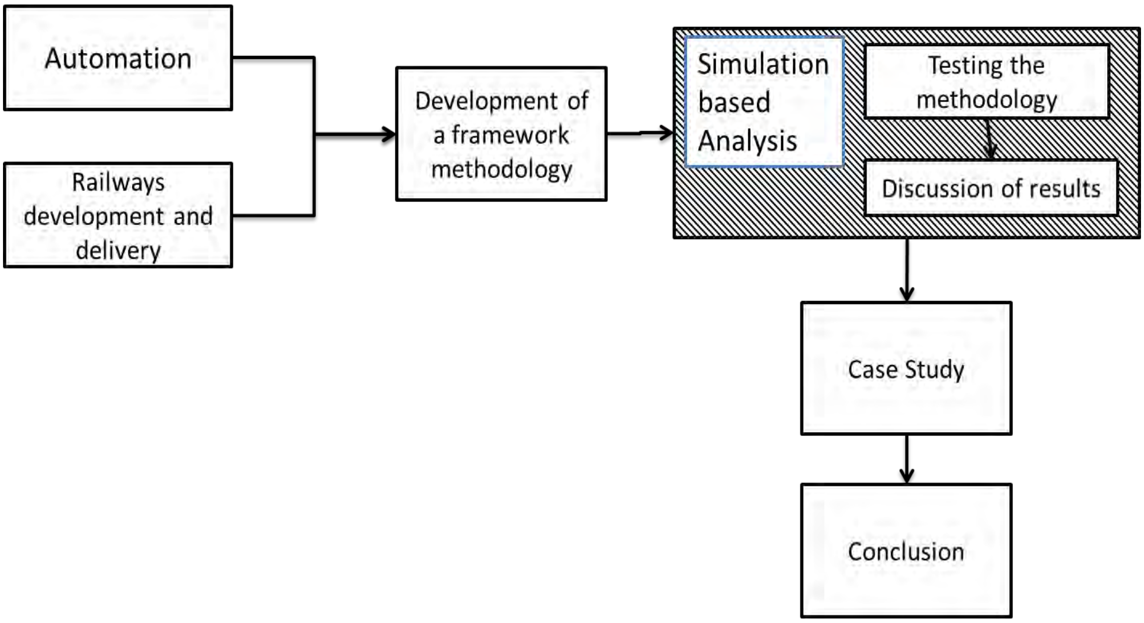


Figure 3: Thesis structure

Chapter 2 presents a literature review focusing on the developments and changes regarding the philosophy of automation. The chapter provides a background for using a visual representation to grade automation into various levels.

Chapter 3 presents a discussion around railway systems thinking, modern mainline railways and highlights the relationships between each of the subsystems.

Chapter 4 presents a discussion around simulation and application in the railway industry. This chapter presents an outline of existing simulation architectures and provides a background to Monte Carlo based simulation. The theory from this chapter is used to construct the simulation architecture in the following chapters.

Chapter 5 builds on the literature surveyed in chapters 2 and 3 to propose a framework to map increasing levels of automation. Each subsystem is discussed in the context of increasing levels of automation and a suitable roadmap is presented in the chapter.

Chapter 6 discusses the testing of the developed framework using simulation methods developed in chapter 4. A Monte Carlo based approach is identified and a suitable simulation architecture is specified to test the framework. The chapter ends with a discussion of the results from the simulation.

Chapter 7 presents a case study on a real-world railway line between Kings Cross and Welwyn Garden City. The section of railway on the East Coast Main Line is chosen and simple automation framework is constructed. The simulation architecture and a discussion of the results are presented at the end of this chapter.

Chapter 8 provides the conclusion for this thesis. There is a general reflection on the various aspects of the thesis. The important outcomes are summarised and recommendations for future work is specified.

At the back of the thesis two appendices are provided as follows:

Appendix 1: Code for the simulation

Appendix 2: Information from the route sectional appendix for the route between Kings Cross and Welwyn Garden City

1.4 Contributions of this Thesis

Work presented within this thesis has been included as a chapter section within a Capacity 4 Rail project. The particular Capacity for Railway project is a pan European railway project whose objective was to analyse various techniques and technology measures that can be used to increase capacity across the EU railway corridors. The work presented within this thesis was used to propose a roadmap for future technologies that can be used on the railway to improve passenger and freight capacity. (Venkateswaran et al., 2017).

A conference paper containing a preliminary hypothesis for the impact of automation on mainline railways was presented at the 2015 IEEE conference on intelligent transportation Systems. (Venkateswaran et al., 2015)

2 Automation and the Work Process

2.1 Automation and its Challenges

Complicated systems contain within them complex interactions between various subsystems. A multitude of such task interactions makes it too cumbersome to manually track information transfer, to mitigate against risk arising from the decisions and to maintain the reliability of the overall system. Introduction of automation into such systems reduces the complexity of managing these interactions by assisting or providing automatic control for task execution.

Automation, according to Sheridan (Parasuraman et al., 2000) can be defined as an automatic response to a situation that is initiated by either a machine or a human. The human automatic response is the culmination of years of experience in handling of tasks and routine execution of actions. The automatic response that is mechanised is what we usually term as automation. This response of a machine could be the culmination of sensing activity, through a sensor array; problem detection, through signal processing and stimuli identification; a decision making an entity, such as a computer; finally, a reaction is solicited from the system through a mechanised manner.

The earliest (and most persistent) methodology of assigning tasks for the machine component of a human-machine system to perform was generated by Paul Fitts (Fitts, 1951). This list of tasks came to be known as the Fitts's List for automation. Put simply, Fitts' List defines which tasks or jobs of a human-machine system should be allocated to the human and which should be allocated to the machine (a computer in present-day contexts). This list essentially meant that the human operator's sphere of influence would only apply to tasks that required cognitive decision making and all such activities that would require time sensitivity would be completely assigned to machines. Such a vision can be seen in the 1965 design for the Victoria line trains on the London underground where the engineers used the function allocation system to design a highly automated metro line but left the safety critical task of closing doors unto the train operators (Dunton et al., 1965).

Table 1: Fitts List for possible areas of function allocation (Fitts, 1951)

Humans are Better At	Machines are Better At
Perceiving patterns of light or sound	Applying great force smoothly and precisely
Detecting small amounts of visual, auditory or chemical energy	Responding quickly to control signals
Improvising and using flexible procedures	Storing information briefly, erasing it completely
Storing information for long periods of time, and recalling appropriate parts	Reasoning deductively
Reasoning inductively	
Exercising judgement	

Over the years significant shortcomings of such a system have been outlined by researchers such as Rasmussen, Rosenbrock, Bainbridge, Sheridan and Parasuraman to name a few (Sheridan and Verplank, 1978, Rasmussen, 1983, Rosenbrock, 1984, Bainbridge, 1983) but the underpinning aspirations that drove the development of the original list still remain to this day (de Winter and Dodou, 2014). According to de Winter and Dodou, the Fitts list was decades ahead of the automation era yet it still predicted correctly the potential shortcomings of system automation in the future. Chiefly Fitts list outlines

- The need to pay specific attention to states of the system where automation may fail.
- The degradation of skills because of the human operator becoming used to an automated system.
- The change in the role of the human from operator to supervisor and the increased need to analyse and drive the training of the operator in this direction.
- The need to keep the human involved in the systems such that there is some meaningful activity. This is essential to maintain alertness of the human actor involved.
- The need to arrive at different levels of automation to better design the roles of the human and the machine in the resultant system.

These conclusions cover the application of automation in various industries and help in defining the direction of the study presented within this thesis.

Levels of Automation

Sheridan and Verplank (Sheridan and Verplank, 1978) laid the initial foundation work for the formalisation of levels of automation when designing the system for the first undersea teleoperator systems. The authors first define the sub-functions of any decision making process as:

1. Request: ask from other party.
2. Get: fetch what is requested or necessary.
3. Select: choose from among options for intended action.
4. Approve: agree or disagree with a selected decision.
5. Start: initiate implementation.
6. Tell: inform what was done.

Further to this they propose the 10 distinct levels for automation in human –machine decision making. Table 2 extracted from the manuscript, shows each level and the task allocation between human and machine in a simplified form.

Table 2: Sheridan and Verplank's early classification for Human (H) – Machine (M) interactions into levels of automation (Sheridan and Verplank, 1978)

Description of interaction	**Actions of operator**
1. Human does the whole job up to the point of turning it over to the computer to implementation	gets options from outside, (H)
	selects action (H)
	starts action (H)
	requests options, (H)

Description of interaction	Actions of operator
2. Computer helps by determining the options	gets options, (M)
	selects action, (H)
	starts action, (H)
3. Computer helps determine options and suggests one, which human need not follow	requests options, (H),
	gets options, (M)
	requests select action, (H)
	selects actions (M)
	selects actions – can be different – and starts action (H)
4. Computer selects action and human may or may not do it	requests options, (H)
	gets options, (M)
	requests select action, (H)
	selects action (M)
	approves selected actions (H)
	starts action if approved(H).
5. Computer selects action and implements it if human approves	requests options, (H)
	gets options, (M)
	requests select action, (H)
	selects action, (M)
	approves selected actions, (H)
	starts action if approved (M)
6. Computer selects action, informs human in plenty of time to stop the selected action	requests options, (H)
	gets options, (M)
	requests select action, (H)
	selects action, (M)
	approves start actions, (H)
	starts action if approved or (M) (if approval timer has expired and human has not disapproved).
7. Computer does whole job and informs human when the action is done	requests select action, (H)
	gets options, (M)
	selects action, (M)
	starts action, (M)
	tells action to human. (M)
8. Computer does whole job and tells human what it did only if human explicitly asks	requests select action, (H)
	gets options, (M)
	selects action, (M)
	starts action, (M)
	tells action, if human requests. (M)
9. Computer does whole job and tells human what it did and it, the	requests select action, (H)
	gets options, (M)
	selects action, (M)

Description of interaction	Actions of operator
computer, decides if the human should be told	starts action, (M)
	tells action if computer approves. (M)
10. Computer does whole job if it decides it should be done, and if so tells human, if it decides he should be told	requests select action, (H)
	gets options, (M)
	selects action, (M)
	starts action if computer approves, (M)
	Tells action if computer approves. (M)

Sheridan further refined the above list to fit a 10 point scale, which has since been used by many others to place levels of automation framework on top(Sheridan, 1992). This list is identical to the previous one with the change being that levels, where computers have a higher level of autonomy, are given priority over the rest lower levels which contain human interactions.

Table 3: Levels of Automation according to Sheridan (Sheridan, 1992)

High	
	10. The computer decides everything, acts autonomously, ignoring the human
	9. The computer decides everything, acts autonomously and informs the human only if it, the computer decides to
	8. The computer decides everything, acts autonomously and informs the human only if asked
	7. The computer decides everything, executes automatically, then necessarily informs the human
	6. The computer decides everything and allows he human a restricted time to veto before automatic execution
	5. The computer suggests one decision/action alternative and executes that suggestion if the human approves
	4. The computer suggests one decision/action alternative
	3. The computer narrows the selection of decision/action alternatives down to a few
	2. The computer offers a complete set of decision/action alternatives
	1. The computer offers no assistance: a human must take all decisions and actions
Low	

Automation of a task requires some basic understanding of task execution. Many techniques and methods in the discipline of Human Factors start from creating a detailed task analysis. Successful task execution amongst humans can be defined under three general categories of behaviour. They are:

- Skill-based behaviour: Certain tasks require the operator to exhibit a particular level of fundamental skill. Classic examples include reaction time, perceptual-motor tracking and visual search.
- Rule-based behaviour: Systems impose on the human operator certain rules they have to follow in order to guarantee conditions such as safety and boundary condition integrity for the system. In the railway context, this would be following the signal aspects and carrying out emergency

procedures. Such actions require the operator to perform the rule book procedures from memory.

- Knowledge-based behaviour: Under certain conditions, the operator might be expected to execute tasks that are outside of their regular routine. Such tasks might require the operator to have knowledge about the fundamental concepts on which the system has been built. For example, if a train driver is expected to initiate an unplanned stop on seeing an obstruction on the track, they would have to estimate the stopping distance based of the train from the current speed.

Rasmussen (1983) detailed each of these behaviours and proposed a systematic method to quantify their impact on task execution. This model is known as Rasmussen’s model for behavioural trinity (Sheridan, 2001). Figure 4, shows a flow diagram for each behaviour level.

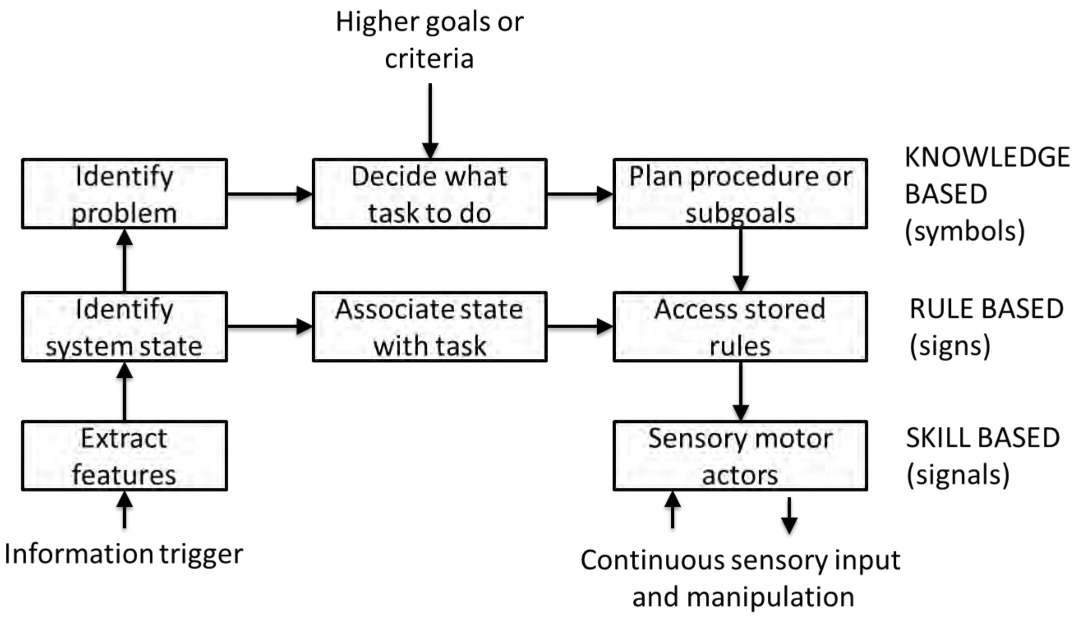


Figure 4: Rasmussens behavioral Trinity, redrawn after Sheridan (Sheridan, 2001)

The above figure depicts the behavioural activity through the system on the vertical axis and each step on the horizontal axis shows an abstraction of information to a higher level and then a simplification.

In parallel Bainbridge (Bainbridge, 1983) proposed very rudimentary stages/levels of automation that would recognise the different levels of collaboration required to complete a task between a human and a computer. The stages are:

1. Instructions and Advice: The computer would provide advice and notifications for the human operator and instructions on certain functions. A railway example is that the computers provide a driver with advice on throttle positions to save energy. In aeronautics, a pilot might be provided with a computer based diagnostic service in order to isolate faults on the system. (§ 4, McGee et al., 1998)

2. Mitigating Human Error: The computer would override the human and correct the system, effectively reverting to routine operation levels.
3. Software Generated Displays: The computer could help visualise the various processes occurring within the system, bringing rich, intuitively formatted and displayed data to the operator, which could lead to better decision making.
4. Relieving Human Workload: The human would make decisions and rely on the computer to execute the various tasks required to implement the decision, thus relieving him or her of the 'mental burden' required to perform both aspects of the task.

The theme of staging automation was received very favourably in the Ergonomics and Human Factors community, as it allowed design engineers to better describe the aspects that they would like to automate. Sheridan, Parasuraman and Wickens (Parasuraman et al., 2000) reduce Rasmussen behavioural model to 4 important stages.

1. Information acquisition,
2. Information analysis,
3. Decisions selection,
4. Action Implementation.

Human decision making follows a similar strategy for assessing and performing tasks. Roth et al. (1999) outline the tasks that train dispatchers, in addition to setting scheduled routes for trains, have to undertake.

1. Monitor train movements beyond their territory,
2. Anticipate delay,
3. Balance multiple demands on track usage,
4. Make rapid decisions,
5. Communicate and Exchange Information.

From looking at these tasks we can assign each of these tasks to the categories proposed by Sheridan, as shown in Table 4.

Table 4: Tasks as outlined by Roth (Roth et al., 1999) and mapped as per Sheridan

Monitor train movements	Acquire Information
Anticipate delay	Analyse Information
Balance multiple demands on track usage	Make decisions and Select Actions
Communicate and Exchange Information	Implement Actions

To add to the above, Sheridan proposed varying grades of automation that would describe the role of the computer in the system. Each of these grades indicates a spectrum from systems with no automation to systems with complete automation.

With the increasing implementation of automation, its potential use and abuse are outlined by Parasuraman and Riley (1997). The paper presents specific examples that outline the pitfalls of use, disuse, misuse and abuse of automation systems. It is concluded that, for a successful implementation, it is necessary to account for the expectations of the stakeholders such as the operator, manager, designers and regulators, with regard to the process being executed. Such careful designing would ensure that systems would rarely behave abnormally and the human interaction with such systems would be smooth.

Further to this work, Parasuraman, Sheridan & Wickens (Parasuraman et al., 2000) developed a framework that allows designers to articulate at a high level the degree of automation for different processing stages within a system. As shown in Figure 5, two systems, A and B, with varying levels of automation across different information processing stages can be described using this framework. Following this technique, various authors have used similar approaches to describe the differences between each system they would like to outline (Onnasch et al., 2014, Sharples et al.).

Further to the discussion on automation levels and stages, Sharples et al., (Sharples et al., 2011) have spent time defining each stage and level for the use analysing the impact of automation on a system. Figure 6 below contains a brief description of each.

From the above discussion, it is obvious that the route we would like to take in order to propose appropriate automation framework for the railway system.

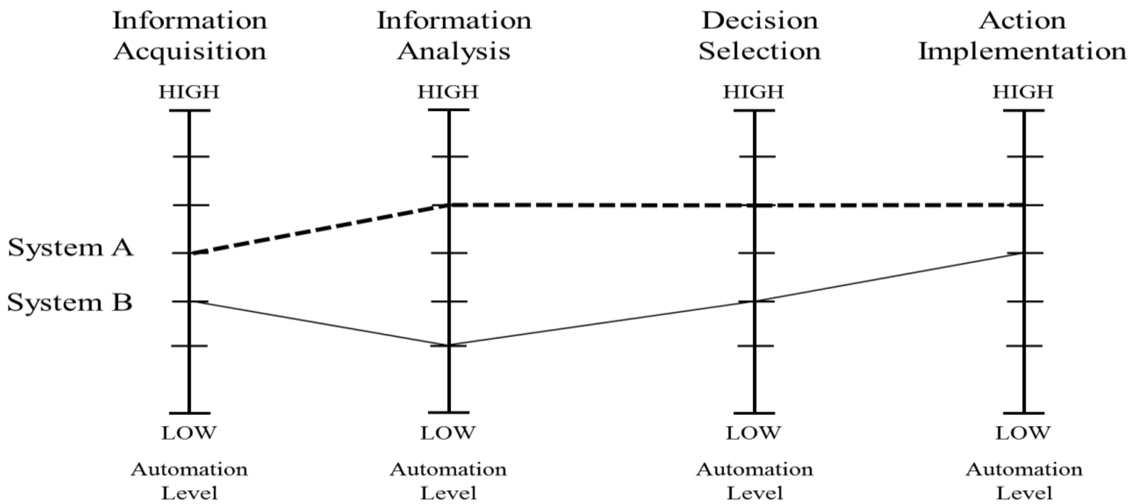


Figure 5: Degrees and Levels of Automation of two different systems, redrawn (Parasuraman et al., 2000)

Information Acquisition	Information Analysis	Decision Selection	Action Implementation
•None: Human gathers all information without assistance from computer or technology, using senses for dynamic information and paper-based sources for static information;	•None: Human analyses all information ;	•None: Human makes all decisions, without any support;	•None: Human implements all actions and communications;
•Low: Human gathers all information but with assistance from IT (telephone/fax/email/CCF/TRUST);	•Low: Computer analyses information as it is received and detects conflicts only as they occur;	•Low: Computer provides decision support to the human to help ensure decision is not unsafe;	•Low: Computer augments human's physical labour (e.g., hydraulic assistance on lever);
•Medium: Information acquisition is shared between the automation and the human;	•Medium: Computer gives a future prediction based on basic information for the short term(e.g. current trains on the workstation);	•Medium: Computer performs basic decision making (e.g. first come first serve and run trains to timetable) and leaves perturbed modes to the human;	•Medium: Computer implements physical actions, but human is required to perform communications (possibly with assistance from information and communication technologies);
•High: Computer and technology provide the majority of the information to the human;	•High: Computer gives a future prediction based on fuller information (e.g. trains arriving in future, infrastructure state, and current situation on other workstations), and highlights potential problems/conflicts over a longer period of time;	•High: Computer performs mid-level decision making (e.g. apply set rules to delayed trains) and has basic plans for implementation during perturbed operations;	•High: Computer implements physical actions and basic communications but human is required to perform complex or unusual communications;
•Full: Computer gathers all information without any assistance from human.	•Full: Computer gives a long-term future prediction using all relevant data (e.g. up-to-date information on train speeds, infrastructure state, etc.).	•Full: Computer makes all decisions under all circumstances using complex algorithms to determine the optimal decision (e.g. based on a high-level prediction of the future state and optimal conflict resolution) and provides flexible plans for disrupted operations.	•Full: Computer implements all actions and communications.

Figure 6: Sample descriptions of levels against stages of automation (Sharples et al., 2011)

2.2 The Impact of Automation on the Human and the System

Any discussion on the development of automation thought is incomplete without answering the question regarding its impact on human skill and how automation might influence the interaction with the machine. Rosenbrock (1984) wrote a seminal article about the replacement of human labour with automation. The question, at that time, was to understand the degradation of the skill level of a human operator of an automated system. In his paper, Rosenbrock further reinforces the concept that the skills and rules components of task execution can be automated but knowledge-based behaviour is difficult to implement. He concludes that, unless the feedback from the human into the system is not systematically analysed, any attempts at automation would bring unsatisfactory results. At the time, scepticism about the effectiveness of conducting human factors analysis was abundant and many technologists did not think it necessary to understand the Human-Machine Interaction. Hindsight has shown that systems with poorly-understood or poorly-designed human-machine or human-system interfaces can have disastrous consequences (Meshkati, 1991, Munipov, 1992, Denton, 1987).

Further reflection about automation leads one to the fundamental question about the role of the human operator post-automation. The conventional belief is that once automated, the system would be able to run without any human help or intervention. This obviously is not what can be observed with conventional automated systems. Such observations were crystallised in a paper aptly titled 'Ironies of Automation' by Lisanne Bainbridge (Bainbridge, 1983). She postulated that in tasks post-automation, human operators in the system would have to:

1. Supervise and monitor the automated system,
2. Intervene when the system is not able to cope with a control.

However, the irony of these roles is that they require the operator to be extremely skilled and knowledgeable about the process. Having insufficient experience at task execution may degrade the skill level of the operator (Onnasch et al., 2014) and create a very unsafe system. To further complicate this process, the operator's attitude towards automation could influence the system's overall effectiveness.

In the same paper where Parasuraman, Sheridan and Wickens (Parasuraman et al., 2000) propose a framework to interpret automation into levels and interactions, as shown above, they brief about four main areas of human performance which would be affected by an automated system. The below stated criterion should be considered as an evaluative criterion for approving a particular automation design.

- Mental Workload (MW): A well designed system would effectively reduce a human operator's task execution workload, allowing them to concentrate on system goals such as collating information and hazard detection.
- Situational Awareness (SA): An automated system will affect a human operators awareness levels for that system. A chief reason for the loss of situation awareness is that now there is a computer between the operator and the physical environment. This could lead to the operator blindly trusting the system to provide actions that are safe and reliable without there being a feedback loop from the environment.
- Complacency: If a system is highly automated then the operator might not expect areas where the automation system might fail. Some sort of complacency check will need to be designed to ensure that the operator is paying attention to the system.
- Skill Degradation: This is a major area of study as the risk of deskilling the human operator is high in highly automated environments. In the event of an automation failure, the operator might not be able to take over from the machine.

Linda Onnasch and group (Onnasch et al., 2014) provided insight into the trade-off between the above stated criterion for systems at varying levels of automation based on the levels and stages of automation as proposed by Parasuraman et al.,(2014). For this analysis, they considered 4 cases of two competing systems (A & B) under automation.

- Case 1: Pure Levels: The two systems being compared are placed into two different levels but for the same stage.
- Case 2: Pure Stages: The two systems are at the same level of automation but at different stages.
- Case 3: Aggregation: The System A is at a lower level of automation in an earlier stage whereas system B is at a higher level at a later stage.
- Case 4: Confound: The system A is at a higher level at an earlier stage and System B is at a lower level at a later stage.

Figure 7 shows all the four cases according to the degree of automation. The authors state that each of these four cases depicts systems with increasing degree of automation. For each of these cases, the trade-off variables are analysed and shown in Figure 8. It is evident from that figure that the degree of automation of the system increases the loss of SA increases, the MW of the operator falls as well. Routine Performance improves with an increase in automation as the task workload drops but performance under failure will decline sharply when the system reaches the critical point 'a'.

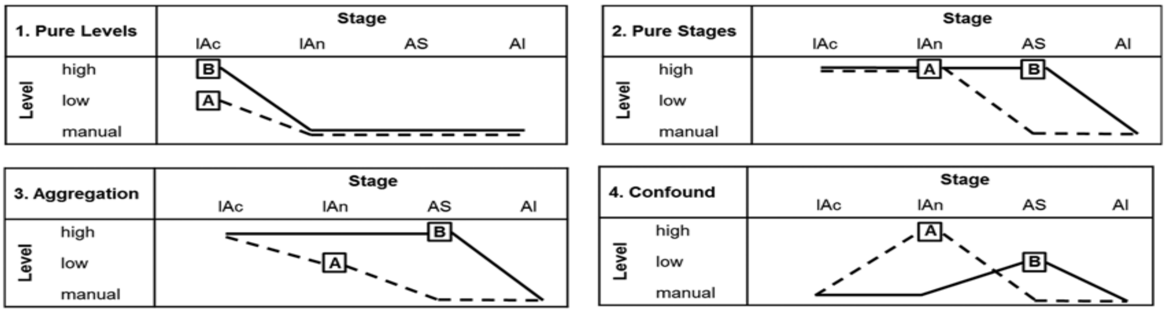


Figure 7: Four Cases for analysing the trade-off between the human performance variables (Onnasch et al., 2014)

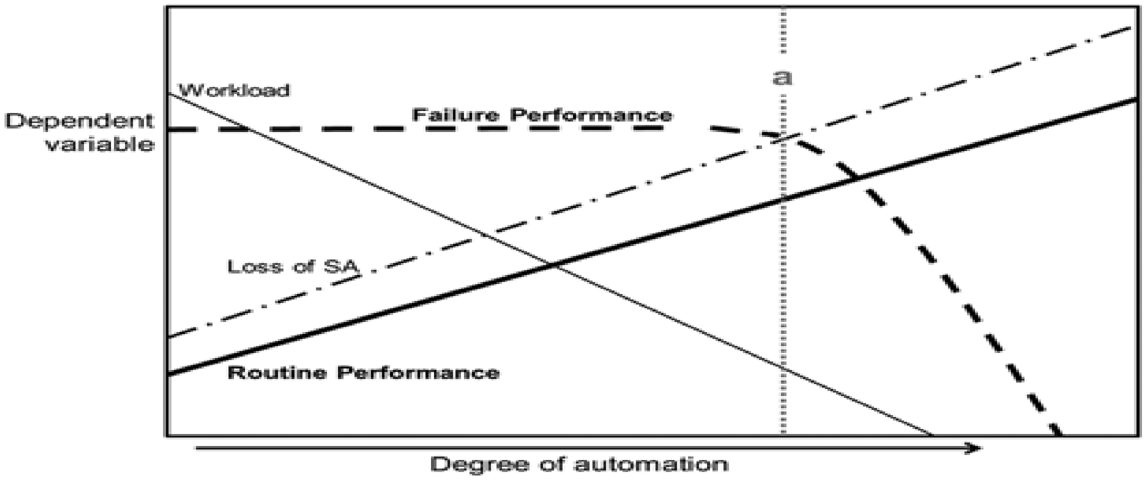


Figure 8: Trade-off of variables, with degree of automation.(Onnasch et al., 2014)

2.3 Summary

This chapter discusses the challenges with implementing automation in a system. This includes allocation of function, organising the tasks according to behaviour and organising the automation level of a system being analysed. The second half of the chapter discusses the impact of automation on human workload, situational awareness and performance when the automation levels are changed within a system.

Chapter 5 of this thesis introduces a framework to increase levels of automation within a system. The concepts presented here will be used to discuss the automation level within a system and to construct a framework to analyse the performance of a system at various levels of automation. The next chapter will discuss the development of systems thinking within the railway and the evolution of the modern railway from simple operational requirements. This chapter also provides an outline of the various systems of measuring and quantifying delays.

3 Railway System: Terminology and Operational Criticality

This chapter is a discussion on railway systems, terminology used and the systems of measuring performance. The first half of the chapter is a discussion on the evolution of railway control systems. The latter half of the chapter is a discussion on system performance and delays. Both these concepts are important to the discussion in the following chapters on automation framework. Concepts presented in this chapter will help us to identify factors for the simulation and also to understand the results from the simulation in the following chapters.

3.1 Railway Control Systems

Control on the railway has the following functionalities / objectives:

- Setting route: manipulation of signals and points along the route,
- Granting movement authority: allowing trains onto a route,
- Ensuring separation between trains on the same track, achieved through proper planning of the timetable.

The railway control functionalities have evolved due to various additions and adaptations made to the system over the last century. To better reflect these changes, one could divide the history of railway control into various periods. For our discussion, the separation between individual periods occurs when there is fundamental change in approach towards the functionalities listed above. Figure 9, summarises each period with the general control philosophy and an approximate year of introduction. The dates are an indication for when a technology was introduced. There are operational railways today which use the control philosophy from an earlier period.

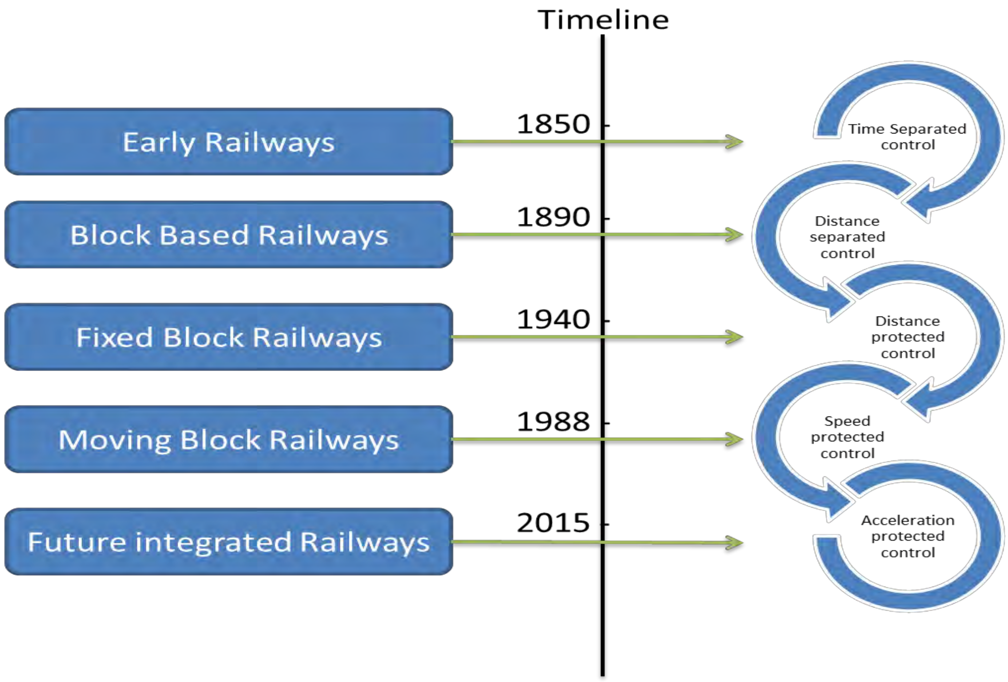


Figure 9: Evolution of Railway Systems (Author, 2019)

3.2 Transitions in Railway Philosophy

3.2.1 Early Railways - Time Separated Control

The first period of the railways is dominated by the concept of Time Block Signalling. The length of the track is broken down into a succession of signal command posts. Historically, each post was manned by a policeman who would be equipped with a flag and a lamp. The station master would be responsible for dispatching trains from the station based on a timetable. No information was broadcast about the status of the train ahead once it had travelled past the station. The safety of the system was guaranteed by the station master, the signal policemen and the train driver. Train driving was primarily carried out on a line of sight basis. As such the speed of the railway was lower to allow the driver to spot obstacles on track.

3.2.2 Block Based Railway- Distance Separated Control

The second period of the railways begins with the introduction of the Regulation of Railways Act in 1889. Prior to this act all railway companies were obliged only to provide safety measures on board their trains. The Armagh Disaster (Wolmar, 2005), which killed around 100 people, most of whom were children, made it imperative to regularise and standardise safety on trains. The Act made it mandatory that all rail companies should provide the following:

- Inexhaustible Brakes,

- Safe separation of trains,
- Interlocking of Signals and Points.

These changes contributed to the era of Absolute Block signalling. The policemen with lamps were replaced by more specialised personnel called signalmen and semaphore-based signalling was put in place. Initially, the signalmen were physically located next to each of the signals and they were tasked with setting the signals to the right indications once a train had passed. They were also responsible for the operation of points on their section of the line.

Detection of the trains also became an important aspect of the control system. Prior to this, trains could be tracked only by the stations they had visited. With the requirement that trains need to be separated from each other, it became of vital importance to track the position of the train along the route. All along the track the signalmen would let the station master know about the position of the trains as they went past them. This could be the first example of the self-organisation of the railway into a joint cognitive system. A joint cognitive system can be described as any system where a combination of man and machines synchronise their actions in order to achieve a coherent cognitive output (Woods and Hollnagel, 2006).

The concept of Movement Authority was first introduced during this second period. Each train, prior to departure, was issued at a station with a physical token by the station master. The token defined the right to occupy the track in a section. The telegraph revolution further enhanced this system in that token machines could be installed in between stations. Such systems helped stop duplication of tokens in a track section. This reduced the need for signalmen along the railway to validate the occupation of track within railways block. The function of the signalmen shifted to setting the route for a train and releasing the route once the train had passed. The drivers drove the train to a semaphore signal which relied on the positions of the arm to indicate the movement authority for the next block. For night time driving, the signalling post was fixed with a lamp (Gas/Electric) and the arm had coloured panes that would stand in for the arm position in the night.

3.2.3 Fixed Block Railway - Distance Protected Control

The introduction of colour light signals – red, green and yellow - which could be controlled remotely, together with the introduction of motorised point machines, meant that signalmen, who were originally dispersed along the line, could now be organised into signal boxes and junction boxes. Finally, the signals and points could be interlocked using a single lever frame movement - interlocking would no longer be dependent on actions of many signalmen working along the line. This enabled a single source of information on the occupancy of the track.

Unfortunately, the introduction of signal boxes added further complexities to the system. Signalmen sometimes forgot where they had parked a train on the line and this led to a number of disasters, mostly in the form of rear end collisions (Rolt, 1966). This in turn led to a necessary regulation, now infamous within the signalling community, called Rule 55 (Rolt, 1966). The rule required that drivers had to send someone from the train crew out to the signal box within 3 minutes of a train coming to a complete stop.

A new age of machines powered by electricity and the telegraph ushered in a new, third period of railway control. The dependence on human skill and cognition for the detection of trains was proving to be unreliable with the increase in traffic. Assistive measures were required to aid human operators in their control of the railway. The introduction of track circuits made it easier for signalmen to track trains. Track circuits, in a very simplified sense, are giant switches made up of track sections which actuate when a train is travelling on them. This information was then relayed to the signalman who would then set a route for the train. The railway engineers soon realised that they could interconnect the preceding signals with the track circuits and create Automatic Signals that would operate themselves when the next block was occupied. These measures reduced the workload of the signallers, allowing them to concentrate on the more important task of setting routes for a train across their section of the track and not have to worry about setting individual points and signals across the section. (Rolt, 1966)

The introduction of electronic relay on the railways was another major step in improving the safety of the railways. Interlocking mechanisms occupied large signal boxes and the mechanisms required a lot of effort to maintain. Relays were introduced, mainly on the interlocking boards, so as to replace mechanical interlocking mechanisms. The new electronic relays were smaller and could be used to control a larger geographic area. In addition, when a relay became faulty, it was very easy to replace the unit without the huge overhead of closing the entire signal box down, as was the case for mechanical interlocking. The development of automatic signals contributed to the development of mimic signal boards. These boards contained within them a representation of the signalling layout of the section being controlled with lights to represent to current state of the signal and indicators for the points. The signaller now had to press a button to set the entry and exit point for the train and the route would be set and released after the train had passed through the section.

The Industrial Revolution had left most of the major cities and towns along the railway system heavily polluted. This led to dangerous conditions where the dense London smog prevented drivers from sighting the signals clearly until they were very close to the signal post. Trains would run past signals that were at danger and this problem soon became endemic on the railway. Such incidents on the railways are known as SPADs, or Signals Passed at Danger. In order to mitigate these problems, several systems were recommended and used on each main line. The accident at Harrow and Wealdstone in 1954 (Wilson, 1954) played a key role for the railway to adopt a standardised system. The system provided a visual display inside the driver’s cab that would indicate to the driver the aspect of the signal ahead. Hence the Automatic Warning System (AWS) was born. The AWS cab display took the form of a black (signal at clear) and black/yellow (signal at caution) sunflower indicator. This system was a warning to the driver in poor sighting conditions. The Ladbroke Grove disaster in 1990 (Cullen, 2000) highlighted the fact that it was not sufficient that drivers were only informed about the aspect ahead. There was a need for a system that would have the driver acknowledge the AWS and would have to apply the brakes if an acknowledgement was not received. This complementary system is what is known today as the Train Protection and Warning System (TPWS).

This is important for the development of the railways because it helped shape the modern era of signalling. Concepts of Automatic Train Protection and Automatic Train Control owe their development to the introduction of systems like the AWS. Within 10 years of introduction of the AWS system, engineers were busy designing a new London underground line, the Victoria line, (Dunton et al., 1965), one that would be completely automated for driving. This system would read the signals from transmission loops and drive the trains according to the aspect on a block. The Victoria line being a metro/urban railway was beneficial to its quick development and introduction.

Fixed block signalling is separation of trains spatially using colour light signals. The most basic type used on the UK mainline is called 2 aspect signalling. As the name suggests, a block consists of two signals: the home signal which guards the entry into the block and an advanced/repeater signal that indicates the aspect of the signal ahead. The repeater/advanced signal is usually placed within braking distance of the home signal so that the driver can stop before the home signal. The blocks can be spaced out as required and this type of signalling system is still in use on line with low density traffic as it is relatively simple and inexpensive to implement and maintain. More sophisticated systems were developed to cope with the increase in traffic on the mainline. This was possible with the addition of aspects and spacing the blocks regularly. Figure 10, Figure 11 and Figure 12 below show the various forms of fixed block signalling systems that are available for separating trains between each other. The separation distance or headway between the trains can be calculated with the formulations as stated in Table 5 below.

For this thesis it is important to distinguish between standard four aspect signalling and signalling augmented with European Train Control System (ETCS) level 1 (illustrated in Figure 13 below). Four aspect signalling has AWS and TPWS applied to it, here the drivers read the signal ahead and acknowledge the aspect ahead. A failure to acknowledge would result in a signal passed at danger (SPAD) condition and, subsequently, the train computer would brake itself to a stop. ETCS Level 1 signalling is an overlay of data transmission modules onto the two, three or four aspect systems. These modules are called balises or beacons that transmit aspect information alongside speed limit for the section ahead. ETCS Level 1 consists of an onboard computer that reads the beacons for the aspect information. The onboard computer is then able to calculate standard and emergency braking curves. These curves serve as a guide to the driver on when to start braking to not exceed movement authority. This overlay allows for the introduction of Automatic Train Protection (ATP) that can automatically apply emergency brakes when the train does not follow a calculated speed trajectory. ATP can be used as a safety mechanism which allows for reduced safety margins thereby decreasing headway between two following trains.

Table 5: Formula to calculate Headway (block length) for 2-, 3- and 4-aspect signalling

Signalling System	Formula	Eq.
Two Aspect	$Headway\ (m) = Sighting\ Allowance + 2 * Braking\ Distance + Separation\ Distance\ (distance\ usually > 15s\ @\ line\ speed) + Overlap + Train\ length$	(1)
Three Aspect	$Headway\ (m) = Sighting\ Allowance + 2 * Braking\ Distance + Overlap + Train\ length$	(2)
Four Aspect	$Headway\ (m) = Sighting\ Allowance + \frac{3}{2} * Braking\ Distance + Overlap + Train\ length$	(3)

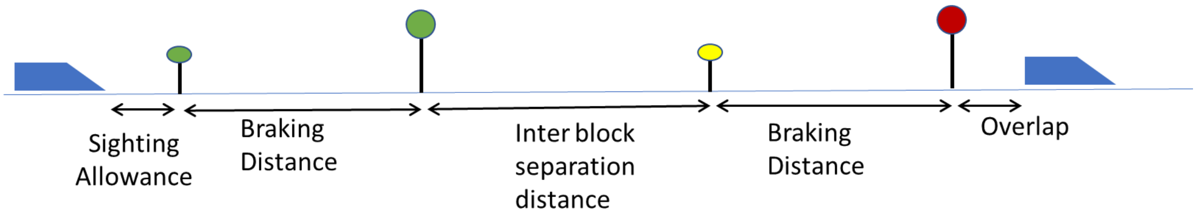


Figure 10: Two Aspect System (Hall, 2016) redrawn by (author, 2018)

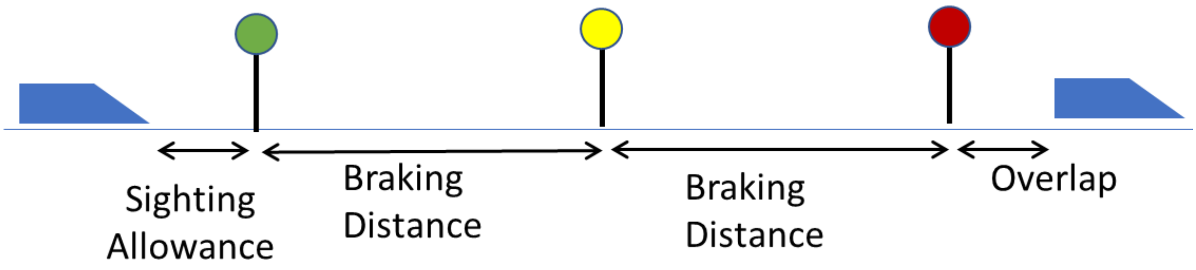


Figure 11: Three Aspect System (Hall, 2016) redrawn by (author,2018)

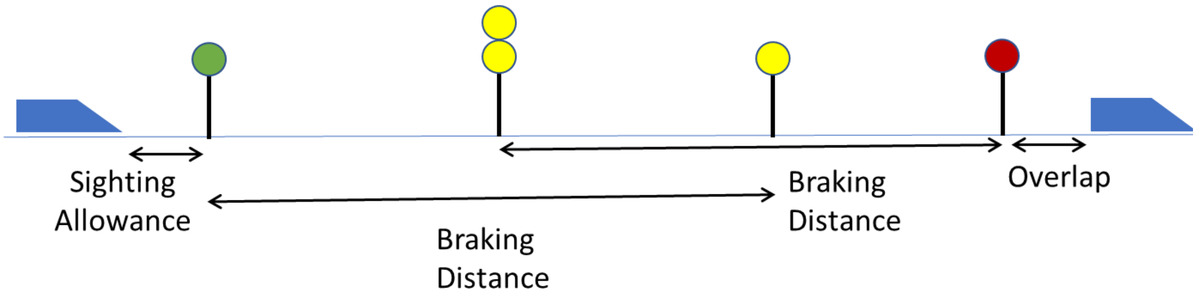


Figure 12: Four Aspect System (Hall, 2016) redrawn by (author,2018)

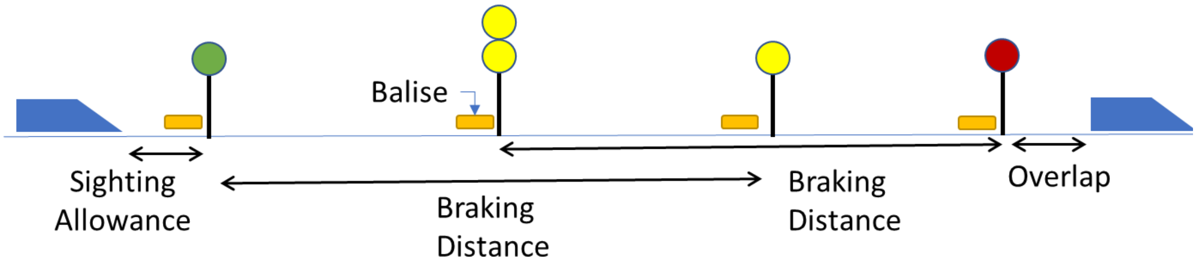


Figure 13: ETCS Level 1 (author, 2018)

The modern era of mainline signalling begins with the introduction of solid-state interlocking. This enabled the development of Integrated Electronic Control Centres (IECC) across the GB network. Computers would now perform Automatic Route Setting (ARS) for trains, based on a pre-planned timetable database, accompanied by sets of instructions on routing. The human operators on the system would now supervise the performance of the system at a high level, intervening in the case of perturbations occurring across the network.

From the previous reading of the development of modern railway control measures, it should be possible to draw the conclusion that, whenever the complexity of control has increased, some form of mechanisation or automation has been integrated within the system such that it simplifies the tasks of the human participants within that system.

3.2.4 Moving Block Railway – Speed Protected Control

Moving Block Systems arose due to the need for greater capacity on railways. Such systems do away with the traditional wayside infrastructure. In this system trains are separated from each other based on the speed of the train ahead. This system can be viewed as a replacement for the traditionally used signal block systems with radio block systems. This is a move away from having fixed infrastructure block markers to virtual block markers.

In ETCS Level 2 signalling system, the movement authority is transmitted through radio transmission technologies such as GSM-R and of late LTE. The system uses the principle of virtual signals to separate the trains. The current aspect is transmitted to the cab and the driver sees a virtual representation of a colour light signal. ETCS Level 2 can be overlaid onto 3, 4 or 5 aspect virtual signalling schema according to the needs of the mainline railway. The train reports its current position to a central Radio Block Control (RBC) over the GSM-R network. The trains are sensed using traditional track circuits and axle counter technology.

The ETCS Level 3 signalling system is a train centred signalling technology. Here the transmission of information relates to the trains current position and speed. The current position is transmitted by the train itself and the system does not rely on external machines to validate this information, therefore the positioning system is required to be of the highest accuracy. The central block controller separates trains based on the actual location of the trains, therefore eliminating excessive safety margins. An absolute safety block is transmitted to the trains for safe separation by the RBC, commonly referred to as moving block. ETCS Level 3 can be further modified with the moving block being maintained statically or changed dynamically. A static block would be independent of the following trains speed. A dynamic moving block would change the distance to the train ahead according to the current speed of the train. Figure 14 and Figure 15, below show illustrations of ETCS Level 2 and level 3.

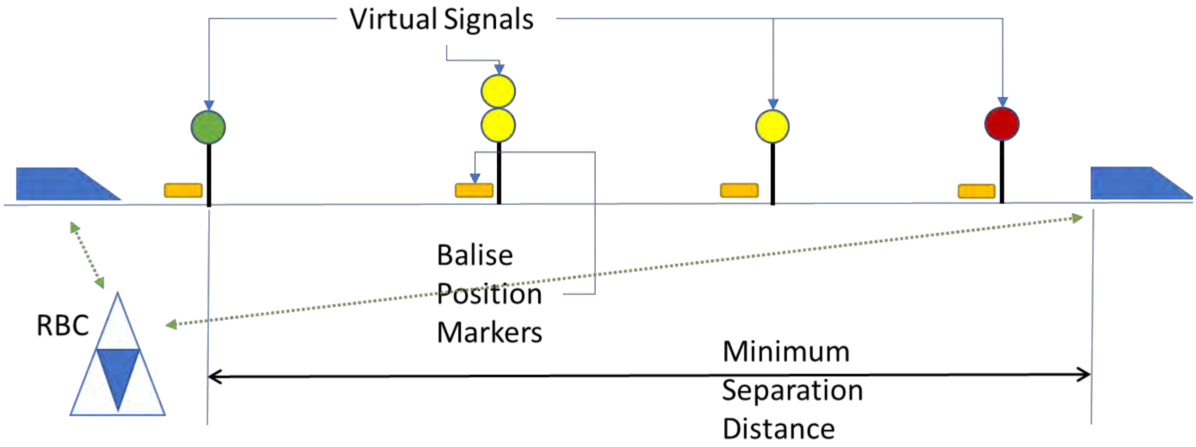


Figure 14: ETCS Level 2 - fixed virtual signal block signalling (author, 2018)

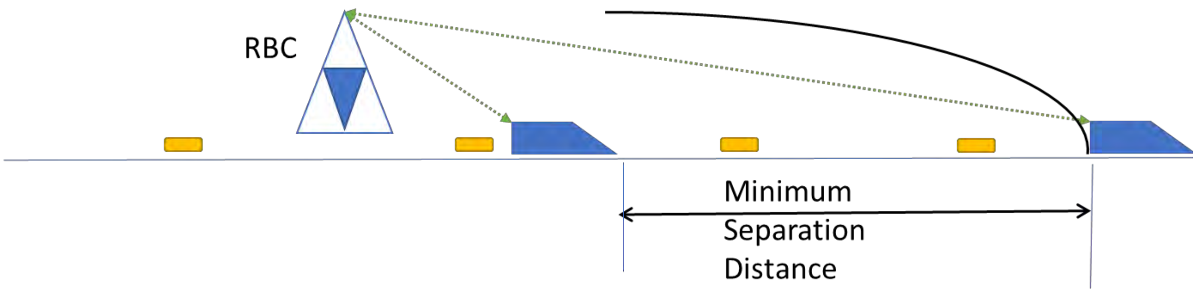


Figure 15: ETCS Level 3 - moving block signalling (author, 2018)

3.2.5 Future Integrated Railway – Acceleration Protected Control

The previous sections described the development of railway control strategies and their impact on the separation distance between the two trains. This separation headway in the previous sections is determined by the speed at which the train ahead is moving. This service headway between trains is critical to maintaining the safe operational interval between them. In essence, the control system in all the above mentioned methods is calculating when the train is needed to brake so that it can stop without running into train ahead.

The later systems (ETCS Level 1, Level 2 and Level 3) erode the fixed safety margins which limit the capacity of the railway. In such systems the deceleration and acceleration patterns of the trains determine the disruption or delay on the network. As such future systems will look to minimise delays by coordinating the speeds of the trains on the network to minimise the delays due to starting and stopping. To achieve a uniform service pattern, it's also important to make driving uniform and avoid errors due to the decisions made. To enable this, there will be a clear need to automate

- Driving: To regularise the driving style, stopping and starting the stations,
- Signalling and dispatch: The systems will need to respond to the traffic in a timely manner and routes need to set and trains dispatched at the right time,

- Platform interface: Ensuring passenger interaction is smooth and safe.

This projection of a future mainline railway can already be seen on railway networks in urban cities. Such networks usually require homogeneity of driving, signalling and platform interfaces to achieve sub 180 second headways.

3.3 Measuring System Performance

The control system discussed in the previous section establish a pattern of service on the railway network. The Service provided by the railway systems, in general, have the following objectives

- Reliable and Consistent: The service provided to passengers needs to be reliable. The railway cannot afford to run perfectly at times and then run at degraded levels of service at other times. The service needs to run consistently so that passengers can plan times accordingly. Hence there is a need to publish a timetable.
- Safe: The service needs to be safe and should expose the passengers and operators to minimal levels of danger. This might include the use of technologies and procedures that lead to a safe usage of the railway system. (Signals, axle counters, track circuits, automatic doors etc.)
- Accessible: The railway service cannot be provided only to people living at the termini, but also it needs to ensure that people living along it have adequate access of usage to its service. Stations need to be spaced at regular intervals so as to provide access to service.
- Capable: Service providers must ensure that the railway provides adequate no of seats on the railway for the passengers according to the demand. This can also be applied to freight railways by substituting passenger seats with carriage capacity to carry goods.

These objectives collectively form the parts of the system performance metrics of a system. The system could be at a degraded state to due to unserved capacity, unreliable service or due to larger environmental events that lead to inconsistent service. (Pender et al., 2013)

Hence, the overall level of performance of the railway could be measured with the help of public performance measure (PPM). This index is defined by the percentage of trains that arrive at their destination on time (Sameni and Preston, 2012) . It serves also as a measure of Quality of service. Arriving on time at a destination reflects onto the punctuality of a train. Any deviation from regular punctual service is termed as a delay (Yuan, 2008a). The threshold level for punctuality is variable between train operators and the types of train service provided across the world. For e.g., Japan defines punctuality as low as 10-15 seconds within arrival times whereas countries like Netherlands define it at 5 minutes. So, a train is considered punctual if it arrives at a destination within the threshold limit of time. Any train that exceeds this limit is said to be delayed.

3.4 Statistical Analysis of Delays

According to Carey and Kwieciński (1994) delays could be classified into three generic types:

1. Initial Delays: Delays that occur when the train enters a network,
2. Original Delays: Delays that occur due to technical failures inside a network, bad weather, increased dwell times and speed restrictions,
3. Knock-on delays: Delays passed on by the train in front to the following traffic.

These delays can be modelled in simulation software and appropriate curves could be obtained for them(Higgins and Kozan, 1998).

The subject of simulation covered in this thesis is focused towards finding methods of producing delay distributions from the perspective of a railway system. Hence the delays arising to the system would most naturally be from the subsystems that interact. The delays could be classified into four major groups.

- Infrastructure based delays: these delays arise from failure due to the infrastructure on a network. The driver would be told to restrict his speed and delays would accrue over the entire network. These are generally a form of original delays. Landex and Nielsen (2006) in their paper discuss the techniques to model delays to passengers when selecting a particular route on their journey. They use a stochastic technique of calculating the passenger delay caused and compare it with the train delay. They further adapt their model to fit a series of timetables and hence make passenger delay calculation possible for different levels of service. Similarly Lindfeldt (2006) discusses the influence that inter-station distance and station length has on creating original especially in the context of operating on bidirectional lines. Many techniques have been introduced for reducing these kinds of delays. Condition monitoring of track infrastructure is being undertaken in order to guarantee reliability of service. Condition monitoring techniques ensure that most system faults could be caught before the system fail. Green et al. (2011), discuss the use of such systems to monitor the health of DC 3rd rail system.
- Signalling Based delays: These arise due to route setting issues, speed restrictions due to signal failures. These could be considered a hybrid of both original and knock on delays. Capacity utilisation is an important for the profitable running of any railway. Hence train operators need to ensure that they have enough recovery time in their timetable to bring the system back to normal operation. Yuan and Hansen (2007), put forward a method to optimise capacity utilisations by approximating the knock-on delays possible on a railway network.
- Rolling Stock based delays: Malfunctioning of rolling stock, in-service brake applications and technical glitches causes delays. This forms an initial delay which means the train is already late when arriving at a source station. Schlake et al. (2011) discuss the use of condition monitoring on-board trains and those placed by the wayside in order to monitor the health of railcars.
- Human decision-making errors: These delays arise due to mistakes made by humans, like passing signal at danger, over speed issues and signal/platform setting errors. These are

generally the most critical on the network because negligence by the operators and controllers could have disastrous consequences like the Ladbroke grove incident (Cullen, 2000) where a driver passed a signal at danger.

3.5 Capacity and Delays

The definition of railway capacity is often debated among researchers. In general, capacity of a railway network can be described as the capability of the network to handle a given throughput of trains running through its infrastructure. More specifically it can be defined as follows:

"Capacity is a measure of the ability to move a specific amount of traffic over a defined rail line with a given set of resources under a specific service plan" — Krueger (1999).

By this definition, it could be inferred that capacity of the network is subject to the quality of the track, the type of signalling and the timetable of the planned services. Abril et al. (2008), further identify that capacity on the railway can be subdivided into four types.

- Maximum theoretical capacity: This is the absolute maximum number of trains that would be able to run on a network section with the separation between trains being at the theoretical minimum. This capacity is possible if there is no variation in the system parameters. It is possible to calculate a theoretical capacity based on empirical formulae.
- Practical capacity: If a timetable is planned according to the theoretical capacity, it would be highly sensitive to delays on the railway. There needs to be allowances made for the addition of buffer time to be consumed in the event of delay. This addition produces a timetable that best reflects the practicable flow of traffic on a railway network.
- Used capacity: This is the actual amount of traffic passing over a given network section for a period.
- Available capacity: The amount of capacity that is unused by a timetable is called available capacity. It is the difference between the theoretical and practical capacities of the network. It is the reserve capacity available on a network in which additional services may be added.

Capacity utilisation can be described as the relationship between the practical capacity and the theoretical capacity of a line. It is expressed as a percentage and most railway regulators around the world follow different methodologies to identify the optimum capacity utilisation of their network. The GB railway network uses the Capacity Utilisation Index (CUI) measure to express the level of usage of the infrastructure. CUI was originally proposed by Gibson, Ball and Cooper. Gibson (2003) in their ground breaking research paper on identifying a scientific method (CUI) for finding the track access charges, while minimising delays due to over utilisation of available capacity also describe the relationship between capacity utilisation and reactionary delay on a railway network

Most studies about capacity utilisation use one or more of the parameters described by Rietveld to record delays on a railway network. Carey (1998) classifies delays into two major groups based on their point of origin on the system.

- Those delays that are caused due to a loss of reliability of the subsystems, for example, a train breaking down, point failures and signal failure are called primary or exogenous delays.
- The delays that are accumulated over the system due to the primary delays are called as secondary or reactionary delays.

Capacity utilisation is, in general, a function of the reliability of a network. A network with high reliability is likely to have low capacity utilisation and vice-versa. If a timetable has a high capacity utilisation, such that two trains run very close to each other, any deviations from the timetable for a train in front will impose delays on the trains behind. We could then surmise that an increase in reliability of a network, R(N), would be inversely proportional to the probability of a reactionary delay, P(D), occurring on that network:

$$P(N) \propto 1/\ P(D). \qquad (1)$$

Studies on the nature of reactionary delays have essentially followed two major methodologies. The first methodology depends on the development of analytical models and then manually conducting a regression analysis on the dataset. This can help identify a distribution pattern in the dataset that can be explained by using a polynomial function such a log, log normal or exponential function.

Gibson (2003) in their paper about the calculation of track access charges, suggested the use of an exponential distribution as a suitable relationship between reactionary delays and the capacity utilisation. They used a T statistic method to identify the parameters of the exponential distribution. The drawback of using the CUI method is that gaps between heterogeneous traffic are ignored in favour of measuring the overall capacity utilisation over the defined time period. Also, identifying the contribution of subsystems for the delay is challenging. The method presented in Gibson (2003) therefore serves as a good analytical model when forecasting delays but cannot adequately reflect the level of automation used on a network. An investigation into the contributing factors would need the reliability of the subsystems to be leveraged into the delay relationship in equation (2).

Vromans (2005), proposes an alternative approach to the calculation of capacity utilisation by considering the inter-train headways, called shortest sum of headway reciprocals (SSHR). Also, a second term is introduced which is the summation of the arrival delays at specific station, called the sum of arrival headway reciprocals (SAHR). This methodology is an enhancement of the Gibson et al. method as it allows for the interaction between fast and slow running services on the line. In a subsequent paper Vromans et al. (2006) discuss the reliability of a network running heterogeneous services. After having derived the weights from fitting the observed delays on the Swedish network to the SSHR and SAHR they proceed to calculate the optimum capacity utilisation for the given network.

Haith et al. (2013), propose an alternative approach to the CUI method based on the Vromans SSHR and SAHR measures. They define a measure for expressing heterogeneity (HET %) in the timetable with which it is possible to express the gaps between the heterogeneous services. The authors point out that the HET method and the CUI method produce the same result for capacity utilisation for a homogenous case, but the HET method produces a more adequate answer when the traffic is heterogeneous. The authors present a regression analysis, with a methodology like the one presented in the CUI method which uses a T statistic method, using the R squared statistic method.

The second methodology depends on using simulation to analyse and predict the response of a network to delays. Seifer (2008), outlines the basic parameters of simulation for the railway networks. There has been an increasing interest in using micro simulation models for the calculation and estimation of delays. (de Fabris et al., Malavasi et al., VanLandegen and Chen, Malavasi and Ricci, Meli et al., Confessore et al., Tomoeda et al., Nunez et al.)

Micro-simulation software such as OPENTRACK and RAILSYS are being used to perform stochastic optimisation for timetables, and to analyse reaction to various delay patterns. Using these methods it is possible to estimate the various types of delay distribution for events such as variations in dwell times, boarding and alighting time, and running times along track sections (Yuan, 2008a).

The aim of this literature survey is to highlight the chief methodologies adopted within the European rail network; the Gibson method is used chiefly within the GB, Gibson et al. (2002) and the Vromans method is used for calculating capacity utilisation in the Netherlands (Vromans et al., 2006) . The content presented in the above papers relies on the post processing of observed data and does not account for loss of reliability of the subsystems. A much more detailed literature survey about the research conducted on the relationship between railway capacity and delays has been undertaken by Mattsson (2007). Seifer (2008) has put together a detailed compendium about the modelling and simulation methods used in the estimation of various kinds of delays on the railway network.

3.6 Summary

The objective of this chapter was to present the research regarding the various railway control philosophies, the formulation of services, the taxonomy of delays, and the nature and usage of capacity. In summary, railways have had significant philosophy changes where the control has progressed from early railways which were time separated to moving block railways which are speed protected against each other. In chapters 5 and 6, the framework and simulation are discussed with respect to levels of automation. The signalling technologies described (four aspects, ETCS Level 1, 2 and 3) in this chapter will be utilised as a framework for increasing levels of automation. The discussion on delays will be used to extract factors affecting operation in the simulation ahead.

The next chapter introduces simulation in the railway context and provides an overview of simulators, both Hardware in the Loop and the Software in the Loop, currently being used in the industry. Also, the chapter discusses the mathematical approach using Monte Carlo simulation in order to generate and control the inputs to the simulation.

4 Simulation and Simulation Methods

4.1 Introduction

A simulator is generally an amalgamation of software and hardware. This definition holds true for most industrial and academic simulators. It can be generally understood that simulators are not just limited to academia or industry; most games, available initially through arcades and most recently with the use of in-home consoles, are also simulation environments. Hence, the above definition has been specifically tailored to exclude those simulators whose primary purpose is recreation.

4.2 Railway Simulation Context

Broadly simulators in the railways are used in three main areas

1. Operational planning: Changes to a network's components during operational period could impact its performance. Simulators help to ensure minimum degradation of performance during the transition period.
2. Component Modelling: During the development phase it is essential to understand and appreciate the complexity of that new system. Simulators help designers to decide on the design requirements for a component on a system.
3. Human based modelling: Simulators can help gauge human engagement, understand the impact of ride comfort on passengers and help in management of employees across the railway system.

The literature survey that follows is aimed to reflect adequately the spectrum of the simulators available and provide an insight into their applications before describing the simulation architecture used during this research.

During the design phase of a railway it is important to understand the capacity of that system. The achievable capacity is often the determinant of the type of technology adopted on a railway network. In the railway context capacity could be measured by the number of train's available, number of seats available and also it can be described by the number of passengers that it has carried per mile. Confessore et al. (2009), discuss a simulation-based approach for estimating the commercial capacity of railways. The simulation model proposed is computer based. It looks at existing train paths and finds the minimum departure times at each station that will not cause a delay on the line. The authors first propose optimisation of the existing timetables and compress the existing train-paths such that it does not cause delays on the mainline railway. They then feed this into a simulator that looks at the existing infrastructure and provides a viable Commercial assessment of the capacity on a line.

The paper attempts to provide a commercial method for assessing the capacity on a network (Verona, Italy) so that infrastructure managers can analyse where they can improve the system to produce the throughput required to satisfy the increased demand.

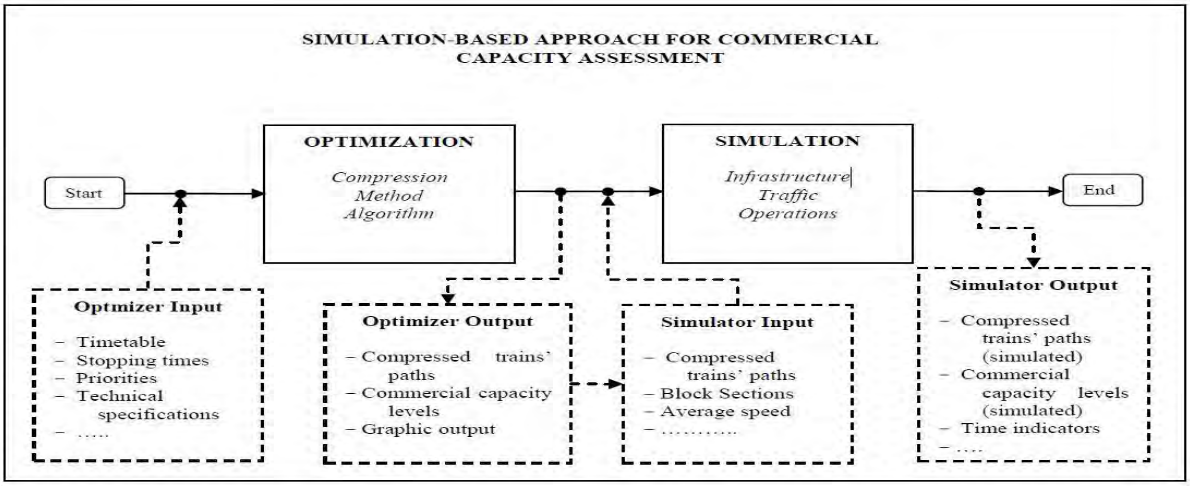


Figure 16: Simulation based approach (Source:(Confessore et al., 2009))

Similarly Kaakai et al. (2007), in their paper on the evaluation of the design of Railway Transit Stations (RTS) recommend hybrid petri nets based simulation model instead of a traditional discrete event based simulation models and microscopic simulation methods. This paper is an algorithm based model. Flow rates of passengers to and from a passenger are analysed and weights are assigned to each area of a station. Then a petri net is used to sieve through the data to find hotspots around the station. This result is then correlated with similar flow modelling done in other methods. The authors discuss exhaustively the various types of flow simulation models that currently exist and their disadvantages. This paper is written from the perspective of a station design which forms a fundamental part of the railway system.

Hardware in the Loop simulators help to test out new components on a railway. This is essential for a successful integration with minimum impact on system performance. Enrico Meli et al.,(Meli et al., 2008) show that accuracy of Hardware In Loop (HIL) test rig generally depends on the correlation between simulate environments and the physical conditions. The authors propose the development and implementation of a 3 dimensional model that would help simulate accurately the dynamics of a railway vehicle. This paper has arisen from an earlier research activity that focused on the use of HIL test rig used for the development of a Wheel Slide Protection (WSP) system. This paper shows the development of a physical simulator from a concept as shown in the block diagram below.

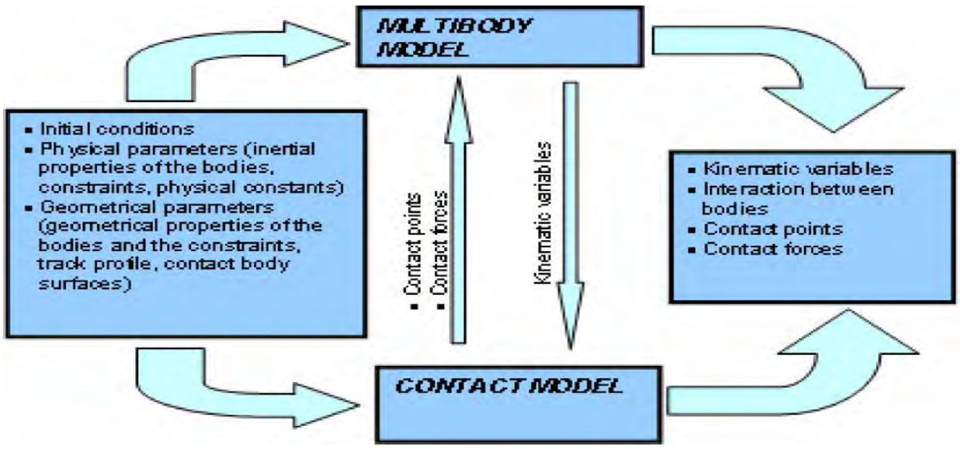


Figure 17: Numerical procedure Block diagram showing the basic concept behind the HIL simulator{source: (Meli et al., 2008)}

Simulators can help analyse driving patterns across a railway automatically. A Single train simulator is a system that inputs the route length, gradient, train characteristics and speed restrictions and calculates the energy usage pattern across that particular route. Such simulators are useful for train companies in estimating the energy supply that would be required for their rail network. Additionally, such simulators can also help researchers understand braking pattern and Coasting strategies. Wong and Ho (2004), in their paper on Coast control, outline the use of a Single train simulator to identify patterns for implementing coast control on trains. Coasting is a standard method that is used by drivers to conserve energy by utilising the train's momentum and the gradient to move the train. The authors identify the current strategy of using coasting board's on the route to inform the drivers when to start or stop coasting is only helpful during set intervals of operation (Refer Figure 18). The authors discuss the feasibility of developing a dynamic coasting methodology that would allow coasting at different times (peak and off-peak) during operation. The identification of possible coasting strategies is undertaken with the aid of a Single train simulator.

Grube and Cipriano (2010), discuss the problem of metro hold times and its impact on passenger wait times on platforms. They approach the problem with two different strategies and compare it with an open-loop strategy that has a constant wait times. The first strategy is to hold the train in proportion to the to the departure time of previous train and the operational headway on the line. The second strategy is to have a Model Predictive Control on the headway times, by which the authors define a horizon and use the possible values to variables in the future to determine the current hold time. They test each method with the help of a simulator that models the operational conditions on a metro line. The test data used is from the Chilean systems from Santiago and Valparaiso.

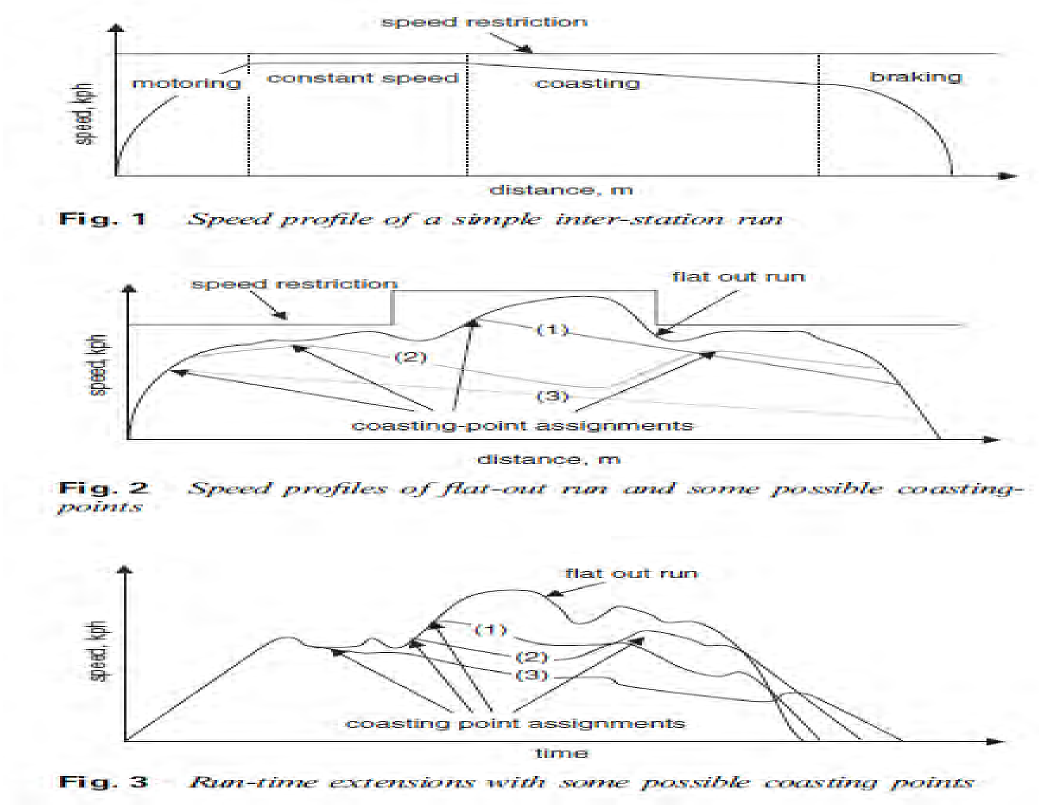


Figure 18: Identifying multiple strategies of coasting. Source:(Wong and Ho, 2004)Table 6, summarises their findings. The authors conclude that neither strategy is seen to be better than the other and hence a recommend doing a Cost-Benefit analysis before choosing either one.

Table 6: Summary of results from simulations conducted by Grube and Cipriano (2010)

Strategy	Mean wait time, s/passenger	SD, wait time, s/passenger	Mean travel time, s/passenger	SD, travel time s/passenger	Mean total travel time, s/passenger	I, min/pass
open loop (base)	4.52	4.53	18.46	1.36	22.98	27.5
simple	3.04	2.92	19.24	1.91	22.28	25.32
MPC	2.31	1.56	18.19	1.57	20.5	22.81

Modelling the operational aspects of a new railway is a tough challenge. Hyun-Kyu and Sung-Kyou (2006), propose the development of an integrated railway environment that can simulate multi-train railway operation. Initially, they create a simulation based model of the various components of a railway system. They then proceed to model the interactions between these components. Lastly, they develop an integrated architecture of a railway system with these components and their interaction. This work demonstrates the use of a simulator to model the real world and help researchers and designers to work around the operational constraints of a new railway. The authors propose to use this research to help aid the R&D work into future railway system.

Figure 19: Railway Operations being visualised on a single track system. Source: (Hyun-Kyu and Sung-Kyou, 2006)

In a departure from the conventional techniques, Krivka et al. (2013) highlight the development of hardware for rail traffic simulator. The authors develop a model railway with the scale of G (scale 1:22.5, gauge 45mm) to demonstrate the reliability of railway safety algorithms when interacting with human operators. The authors first model the communication topology and then design a vehicle with communications equipment fitted into the scale model. Additionally, each train is equipped with a camera and separate desk with a control panel is placed for human drivers to drive the train (Refer Figure 20). The goal of the experiment was to allow human users to drive as aggressively as possible with the system always falling to safe operation. This activity was aimed to be a teaching aid for students learning about railway control systems.

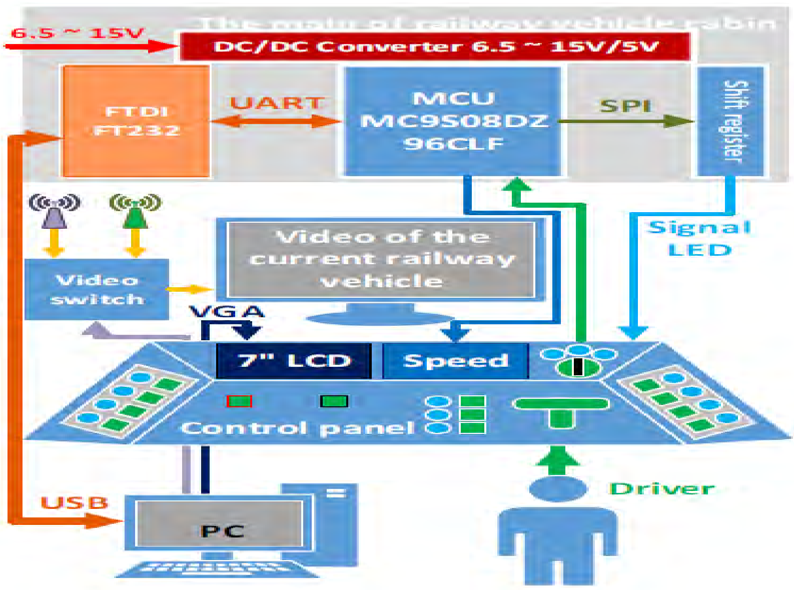


Figure 20: The simulators block diagram as proposed by Krivka et al., Source:(Krivka et al., 2013)

Railway operations are dependent on passenger behaviour. Passenger flow changes due to incidents on the railways could make on time operation difficult to contain. Tomoeda et al. (2009) have developed a real-time network simulator called “KUTTY” that takes into consideration the possible routes that passenger might take during operations. It is also used to redistribute the schedule of the trains in order to avoid overcrowding of the trains. Figure 21, shows the design flow for the simulator.

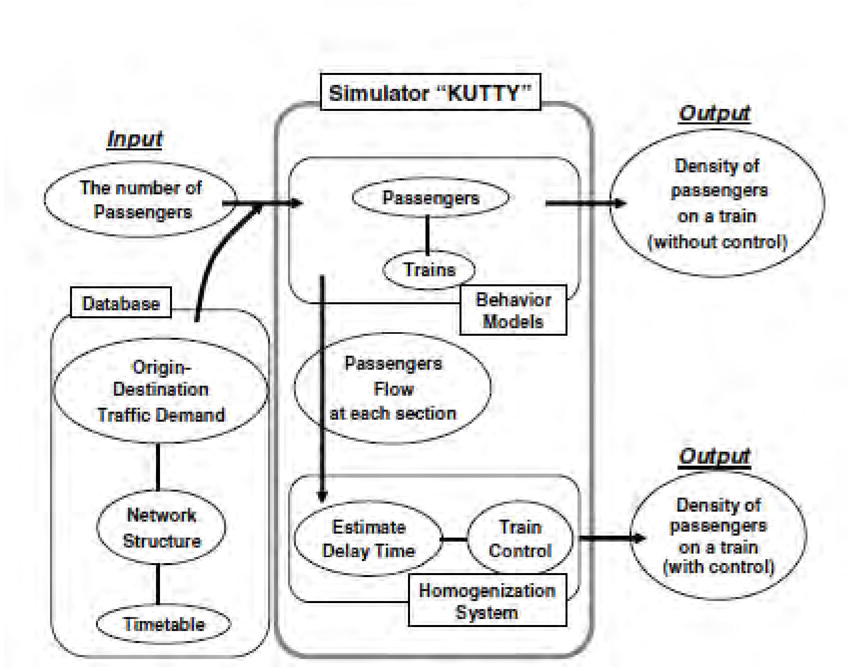


Figure 21: The design flow of "KUTTY". Source: (Tomoeda et al., 2009)

Simulators in the railway often find a major use in understanding the ride characteristics of a rolling stock. It is used to analyse the impact that forces have on the comfort levels of a passenger. On railways that have a lot of curved transitions trains must decelerate in order to be able to pass through that section. Some trains can tilt on these sections and hence can travel at similar speeds without slowing down. Kwangsoo et al. (2008), discuss how a tilting train simulator could be used to study the ride comfort on board a tilting train. Specifically, they measure the heart rate and blood pressure variability of a passenger with varying degrees of tilt applied in a simulator. Figure 22, shows a rendering of the proposed simulator. They conclude that such research could help estimate the best tilt angles on a curvy railway like the Korean National Railways.

Situation awareness of railway controllers plays a huge role in ensuring the safety of the railway during operations. As such when a new system is replacing an older system there is a need to retrain the controllers in these newer systems. Gaming simulations could be used to identify possible areas that might influence decision making among controllers. Authors Lo and Meijer (2013) propose techniques for measurement of situation awareness for gaming simulations. Already such techniques are used extensively for individual levels in simulators for airplane cockpits and driver training. The authors propose to use a similar framework to measure situation awareness across a group/team in a multi-actor approach.

Figure 22: Graphical rendering of a tilting train simulator. Source (Kwangsoo et al.)

Another area of concern is planning for adequate use of resources on a railway. The most important resource on a non-automated railway is the crew that run trains. Crew management is an important activity in a railway, as the working hours of a crew member is limited and regulated by various associations and unions. Chahar et al. (2011) have developed a strategic crew planning tool that would enable Norfolk Southern (NS) to evaluate impact of changes to the crew service rules or train service on railway performance. They discuss the application of such a simulator in determining the various bottlenecks on a railway network and help manager's roster their crews in advance to accommodate the changes to the system. Such measures can help companies like NS adopt strategies that would reduce operating costs.

The literature survey reflects the varied applications of railway simulators across the industry. The literature presented above is a sampling of the activities for which simulators are used in the rail domain. Simulator classification is difficult because there are no concrete methods of evaluation of a railway simulator outside its current context. But there is a need across industry and academia to better understand the applicability of a simulator in understanding a problem. The following section will discuss in detail the author's efforts to put forth a classification methodology.

At this point an outline of the broad category of simulation relevant to subsequent discussions is warranted. Essentially, there are three broad contexts. These are:

Micro Simulation: These are local simulations about a particular scenario and controlled input variables. These types of simulations are generally used for short term analysis of large systems. But it is important to remember not to generalise results from such simulations into a wider context as micro simulations only provide us with an understanding of how the system's impact on a particular environment, it leaves us clueless about the impact of that environment on the system. For examples Hewison (1981) studied the impact of construction of tunnels on the MURL project. Using micro simulation techniques, they author could predict the impact that the tunnelling would have on existing traffic on the railway lines. .

Macro Simulation: These types of simulations often encompass large system with variable input-output conditions. In addition to the inputs, there are a set of controls which would control how a system would react to certain stimuli at different points in time. In the railway context, simulations that involve timetabling are highly complex as the input can be quite constrained, but the controlling factors would consider time of day, passenger demand, freight demand, permissible speed limits, etc. Ward(1982) analysed the impact of un-timetabled traffic being inserted into the timetabled day, this is a classic example of macro simulation as it helps us understand behaviour under unpredictable events occurring to the system.

The third and most useful of all simulations are **Macro-Micro (meso) Simulations**. These types of simulations help analysts model an entire network and study the impact of micro level adjustments such as signalling, station dwell times and rolling stock changes Schlechte et. al.(2011)provide us with a bottom approach where they detail a microscopic bit of the system and then aggregate these subsystems to form a whole system. They allow the user to make microscopic changes and then simulate the system to provide the possible outputs, then they dial down to the initial microcosm and study the impact changing other parameters has on it.

4.3 The Monte Carlo and Quasi-Monte Carlo methods

A regular physical experiment needs to be set up to tight controls on the input variables. However, for a real world system such as the railways it is impossible to run the experiment without any noise or error creeping into the inputs. The error accumulation can often lead to mistaken conclusions or inferences of the simulation.

The Monte Carlo (MC) method was developed to analyse the output from a system given a set of random input variables. The input variables are a result of random well distributed samples from a uniform distribution. The Monte Carlo estimate of a function f(u) integrated over a set B of n random samples is given as

$$\int_B f(u)du \approx \frac{1}{N} \sum_{\substack{n=1 \\ x_n \in B}}^{N} f(x_n) \tag{1}$$

Where x_1, x_2, ..., xN are N independent random samples from the uniform distribution within the space of a unit cube I^s. 's' represents the number of dimensions of the function f(u) and B is a subset of I^s. The probabilistic error bound of the estimate is $O(N^{-1/2})$.

The Monte Carlo method consists of four main steps

1. Identifying the factors for the simulation: The problem needs to be formulated and the input parameters need to be identified. Identifying important factors contribute towards elimination of uncertainty reduces the error accrual.
2. Statistical modelling: The random variables need to be analysed for their statistical distribution function, dependence or independence with the output or between themselves.
3. Random number generation: Generating the random variables reflecting the parameters of the factors identified previously.

4. Iteration: it is necessary to run enough number of Monte Carlo simulations to increase the reliability of the results.

The chief features of interest with this method are

1. Convergence on simulation is computationally robust, provided enough number of simulations is carried out. The minimum number of simulations is dependent on the minimum amount of simulations required for majority of the outputs to fall within the safe/acceptable region.
2. The number of dimensions to the problem, i.e. the no random input variables does not affect the simulation. This stresses that all input factors need not be captured in order to produce a simulation that lies within the acceptable neighbourhood of the result space.
3. It's a simple method to implement for engineers who are not well versed with statistics or other formal methods of developing proof. This means that even a badly designed model is not very far from the truth.

However, the MC method is not perfect and suffers from the following vulnerabilities:

1. The confidence in the final calculation steps is not a guarantee. The accuracy of the results is not predictable and is heavily reliant on the quality of the sampling for the input factors.
2. Given the probabilistic error bound given as $O(N^{-1/2})$, it should be noted that the nodes N should be independent random samples. A random sample is strictly a theoretical concept and a real world generation requires picking from some series of pseudo random numbers.

Following from the above discussion, if we can construct a set of nodes N explicitly whose absolute error value is not greater than the average then we can improve the error bounded ness of the MC method. This formulation is termed as a Quasi-Monte Carlo (QMC) method.

The QMC approximation of the MC estimate can be written formally as

$$\int_B f(u)du \approx \frac{1}{N}\sum_{\substack{n=1 \\ x_n \in B}}^{N} f(x_n) \qquad (2)$$

Where $x_1, x_2, ..., x_N$ are N independent deterministic points, and B is a subset of unit cube I^s. The deterministic error bound of the QMC method can be given as $O(N^{-1}(\log N)^{s-1})$.

The advantages of the QMC method over the MC method are

1. Because of its deterministic procedures and deterministic sampling of points the error bound on the calculation can be tightly controlled. Therefore, achieving a higher precision with the resultant calculation.
2. Using the same computational effort as a MC computation, a QMC computation of a function can converge towards a result of higher accuracy.

The QMC approach is a suitable method for constructing the simulation architecture. The simulation architecture will need to be tested for varying automation levels. The steps outlined for the application of the MC approach can be as a skeletal framework for the simulation architecture.

4.4 Implementing a Quasi Monte Carlo Simulation

A Quasi Monte Carlo (QMC) based simulation requires a practitioner to understand the various factors associated with a process. The following example demonstrates the use of a QMC framework to test the sensitivity of dwell times to the door characteristics. The example is limited to the preliminary identification of the factors and the formal equations for dwell time.

The overall journey time of a train is composed of two elements. The first is the running time which is based on the trains' traction and braking characteristics. The second element is the dwell time of the train when it stops at stations. For urban rail transportation (light rail, metro and suburban rail), such as London Underground, the dwell time is critical to maintaining the headway between trains. Nigel Harris (Harris, 2006) bases the calculation of dwell time on the basis of 5 components.

- Alighting Time: Time taken by the passengers leaving the carriage through a door.
- Boarding Time: Time taken by the passengers to board the carriage through the door.
- Interaction Time: Time taken between the last people to alight and the first people to board where there is an interaction between the two sides.
- Machine Time: Time required to disarm/arm the interlocking between the traction system and the door opening and closing. This interlocking between the door and the traction equipment is employed as a safety function to prevent the train from moving with its doors open.

The overall expression to calculate dwell times can be calculated using the equations below.

$$Alighting\ Time = 1.5 * \left(1 + 0.9 * \left(\frac{T}{VC}\right)\right) + (A^a * DWF) \tag{1}$$

$$Boarding\ Time = 1.5 * \left(1 + 0.8 * \left(\frac{T}{VC}\right)\right) + (B^b * DWF) \tag{2}$$

$$Interaction\ Time = 0.027 * A * B \tag{3}$$

$$\text{Dwell Time} = \text{Alighting Time} + \text{Boarding Time} + \text{Interaction Time} + \text{Machine Time} \tag{4}$$

In the equations above, the factors plugged into the equation are as follows.

- Values 0.9 and 0.8 are taken from (Harris);
- T is Number of through passengers;
- VC is Vestibule Capacity;
- A is number of alighting passengers;

- B is number of boarding passengers;
- DWF is the door width factor which relates to the effective door width of the train.

The equations above are applied to the most critical door position during peak hour loading rates for a given railway network. For our discussion, let us assume that we are carrying out a Monte Carlo simulation to identify the sensitivity of dwell time to the input factors DWF, VC, A and B.

The first step will be to determine the input distributions for DWF, VC, A and B. The limits and the distributions assumed are from the study on door factors presented by Nigel Harris (Harris, 2006). The next step in the process generates random numbers for the inputs. The random number sequence is important as it determines the quality and the reliability of the output being generated. Sobol's and Niedretter's are types of pseudorandom sequence generators that can be used to seed the simulation. The numbers generated are then transposed onto the required distribution identified in the first step.

In this case the simulation is created as a simple function which is controlled using loops that yield the overall dwell times for a given combination of DWF, VC, A and B factors. The simulation is first tested for stability, i.e, the number of iterations required for the simulation to provide a stable output and then tested for convergence, i.e., identification of regions in the experiment space where the output makes sense in the real world. The resultant dataset obtained from the simulation is then fed into a sensitivity analysis.

Therefore, a Monte Carlo simulation can be used to generate a spatio-temporal model that can be used for further analysis. A quasi-MC simulation is an improvement on an MC model in that it can be used to reduce the number of required simulations by 'folding' the sample input space and therefore increase the stability of the simulation.

4.5 Implementing the Simulation

The framework developed in Chapter 5 is tested in Chapter 6 using a simulation test bed. This simulation framework is split between a macro-simulation controlling the Monte Carlo simulation, Implemented in Matlab and the microsimulation of the railway structured with BRaVE.

4.5.1 Macro Simulation with Matlab

Matlab is a popular matrix based academic tool for constructing simulation experiments. It contains multiple toolboxes that aid rapid modelling and simulation of experiments. In this context the parallel computation toolbox was used to run multiple simulations in parallel to one another with Matlab. This resulted in saving almost 90 hours of computation time. The architecture for parallelisation is produced below under section 1.4.3. Following from the discussion on quasi-Monte Carlo method, the following steps are implemented using Matlab.

4.5.1.1 Generation of Random Numbers

Random numbers are generated using a Sobol quasirandom {Sobol, 2001} sequence generator for all the variables. For each of the variables based on the type of distribution (uniform, normal or exponential) mean and standard deviation are specified.

Sequences that are uniformly well dispersed within their neighbourhood are said to be of low discrepancy or quasirandom. These sequences are different to ordinary uniform random numbers generated in the interval (0,1). A regular uniform random number generator produces an independent number for every trial. A quasi random sequence on the other hand is aware of the locale of the previous point when generating the current point.

Sobol (Sobol, 2001) sequence is an example of an LPT sequence of base 2. They are quasi random sequences of very low discrepancy. Other families of similar sequences include Niederreitter, Faure, generalised Niederreitter and Niederreitter- Xing sequences.

The generation of Sobol based sequences is available freely online for any researcher to use. The file is reproduced with changes to suit the framework implementation in the attached Appendix 1.

4.5.1.2 Simulation Setup

The simulation is setup is critical for the accuracy of the results. The Matlab code passes the name of the file, the random variables generated, and the variable markers necessary to run the simulation in BRaVE. The simulation then returns the results back to Matlab.

The output data from the simulation in the form of Run time and Dwell time is stored using Matlab. Due to the parallel nature of the code, concurrency of access is an issue. There needs to be a fine tuning of minimum simulation time per iteration to minimise the overall run times.

4.5.2 Visualising the Data

The data is visualised using a box and whiskers plot and scatter chart using the statistics toolbox in Matlab. The scatter chart is used to show gradation between each level of automation. A box and whiskers plot are used to show the relative position of each level of automation with respect to the other. Using a box and whiskers plot it is easy to ensure that most of the simulation lies within the acceptable region of simulation.

4.5.3 Microsimulation with BRaVE

The Birmingham Railway Virtual Environment (BRaVE) was developed at the University of Birmingham by Dr. David Kirkwood (Douglas et al., 2017, Nicholson et al., 2015) for performing whole system microsimulation. It is written and developed using the JAVA integrated development environment.

BRaVE is a time integrated simulation that calculates the present state of the system per discrete time step. The total result of the integration is visualised as trains running on track. It is possible to produce perturbations and knock on delays during the simulation either directly through the simulation or alternatively through programming the API. It has been used in the past to carry out capacity &timetable analysis, network analysis, power system analysis and energy research.

For the research work carried out, Brave can be considered as a time based solver. All the random value generated from Matlab is sent to Brave using a plugin interface. A separate programming interface was written from within BRaVE to receive and assign the variable to the various components.

The components of simulation are

1. Signaller: Controls the movement of a train within the simulation. The signaller can be programed to allocate the resource in a time bound manner. It is possible to inject delays into the system using the signaller.
2. Driver: Controls the driving style of the train. The thesis required the modelling of a driving profile to represent human driving. Such a profile was implemented and used from within this component.
3. Dispatcher: Provides the dispatch notice to the train on arrival at a station. This component controls the departure of trains from a station.
4. Routing algorithm: On approach to junctions, BRaVE is configured with routing algorithm in order to prioritise the traversal across junctions. The default algorithm is a First Come First Serve (FCFS) algorithm. This algorithm is maintained during all simulations in order to not add further complexity to the simulation.

All the programming written to co-ordinate the simulation is reproduced in Appendix 1.

4.5.4 Overall Simulation Architecture

The simulation workflow is shown in Figure 23. The first step is to generate Quasiradom points using sobol generator. The next step will be to create a matrix that controls the simulation, the input factors and collect the outputs.

Figure 24 above shows the overall parallel implementation of the architecture. For this research a computer rig contain 10 dedicated processors was used. This reduced the computational time from by 90% with an average of 180 mins per set of 1000 iterations post parallelisation.

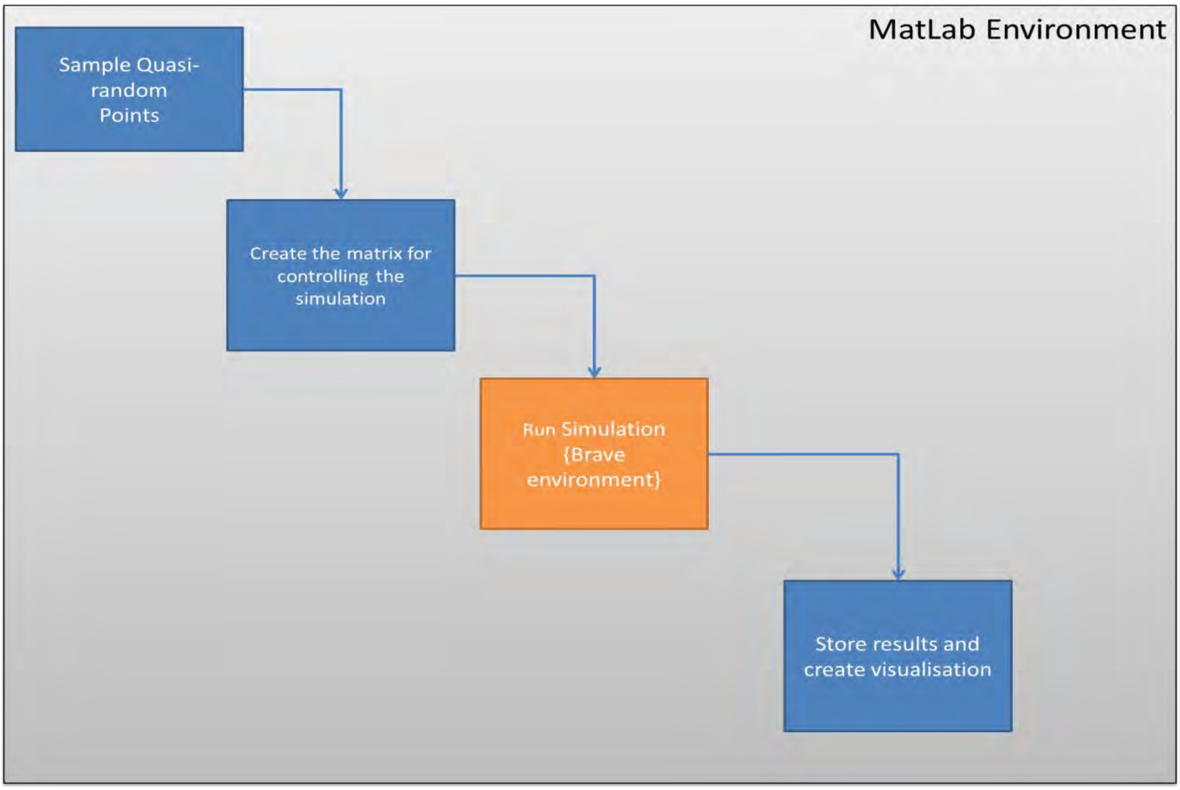


Figure 23: The simulation workflow.

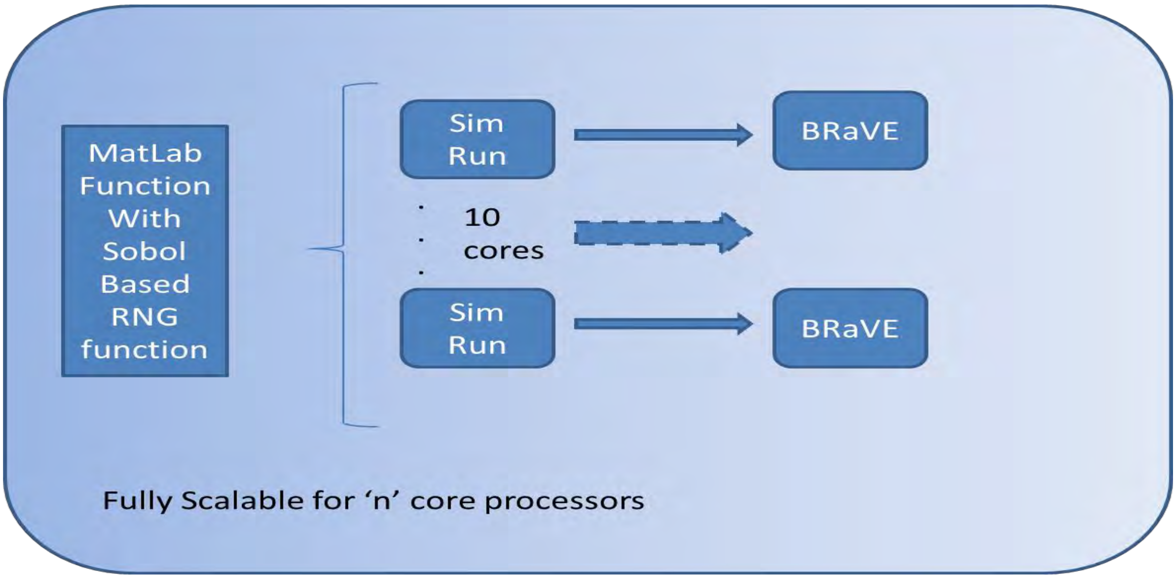


Figure 24: Overall simulation architecture.

4.6 Summary

The chapter presented an overview of the various simulation methods available. This included some software architectures for simulation, some hardware simulation and software based simulations. The differences between macro and micro simulations was brought out.

The chapter provided a brief introduction to Monte-Carlo and Quasi Monte Carlo simulation. An example on modelling the operation of a railway door was used to show the application of QMC simulation to the problem. An overview of the simulation architecture proposed is shown here. Chapter 7 will cover these areas in much greater detail.

The next chapter presents the development of a framework for studying automation. This chapter will pull the information presented on automation in chapter 2 and the study of railway technologies presented in chapter 3.

5 A Framework for Increasing Levels of Automation

This chapter covers the development of a framework to identify the various stages of automation with in the railway context. The concept of this chapter is to answer two vital questions

1. Can we develop a frame work for identifying successive levels of automation?
2. Are the developed levels easily distinguishable from each other?

As it can be assessed from the previous chapters (Chapters 1 and 2), the railways are not new to technology induction. The various technologies can be arranged successively using the methods pointed out by Sheridan and Parasuraman.

5.1 Developing a Framework

A framework is a set of protocols, guides and scaffolding required to construct a desired end-goal. Such a framework should be robust enough to capture the central theme for each subsystem.

The development framework can be summarised into steps as the following.

1. Identify task for automation,
2. Analyse and Identify interacting subsystems,
3. Establish an automation end goal for the task,
4. Identify factors that affect the variability of the task,
5. Create a level of automation plot,
6. Test the levels for distinction,
7. Show distinctions between each level.

The steps of such a framework can be detailed as shown in Figure 25. Initially we need to define the task that we are automating, such as train driving, platform supervision, route setting and so on. Regardless of changes to the technology it must be understood that this identified task needs to be carried out through interactions between the subsystem. Further the sub-functions need to be detailed as a subsystem could have multiple functions. It is also important to establish an end automation goal, usually this is a completely autonomous level where human intervention is limited or not present at all. Finally, it is important to show that the developed levels are contiguous and distinct from each other. Figure 26, shown below is from the UGTMS standard which shows the organisation of operation in the railway context.

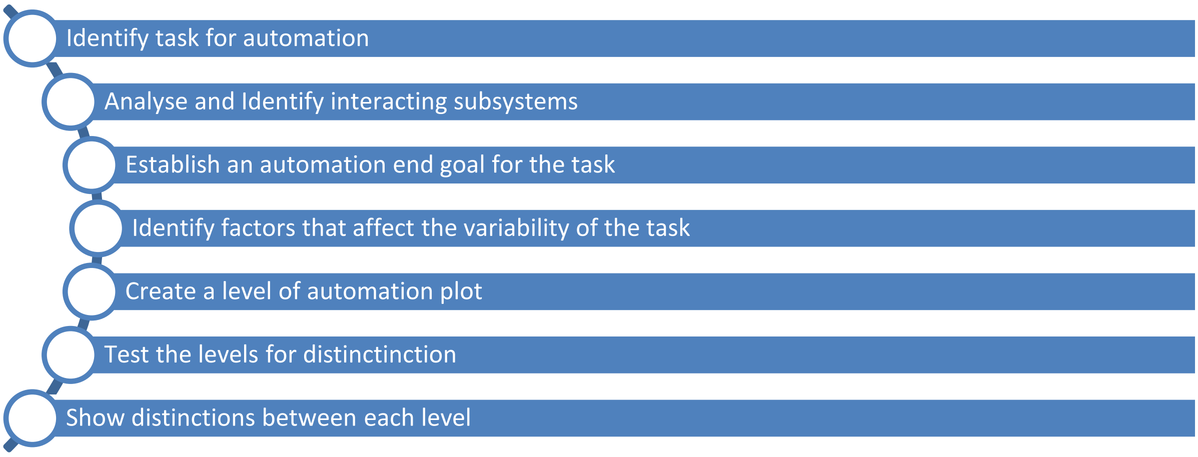


Figure 25: Framework for automation improvements

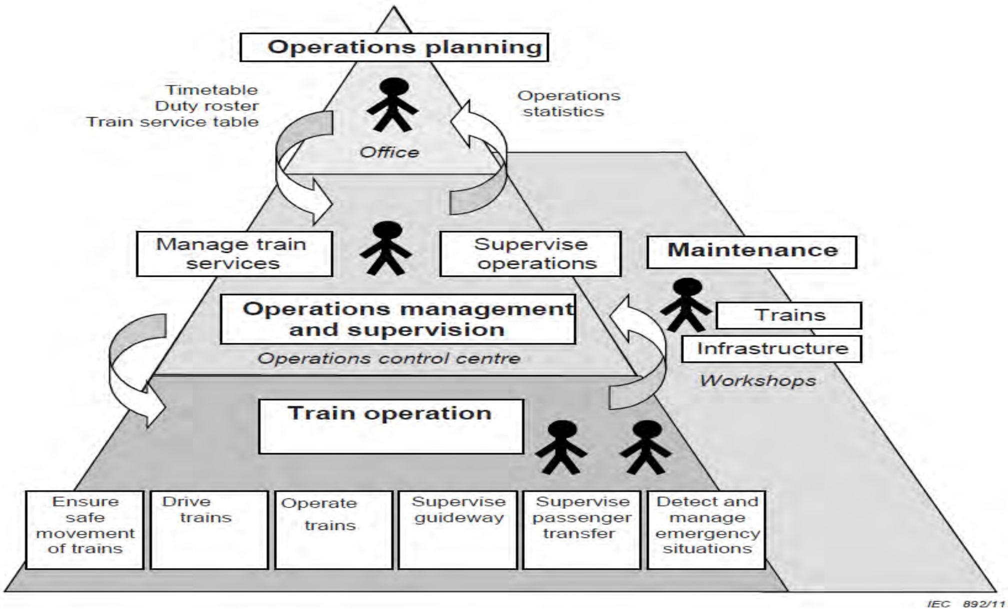


Figure 26: Organisation of Operation (BSI, 2014)

5.1.1 Identify Task for Automation

From the literature on automation, it is evident that not systems that are subject to automation but tasks that are being automated. Rasmussen, Sheridan, Verplank and other researchers provide examples of tasks that are suitable to be automated. So, it is critical that we identify a specific task being automated. For the purpose of demonstration let us analyse a simple task of fare collection.

Many railway business owners are aware of fare evasion by passengers and employ staff to check and validate fares both on board a train as well as at the station. However, fare evasion still occurs because there are gaps in the coverage by staff or sometimes the operator chooses not to employ staff at all stations but only provide them at the main junctions on a network for financial reasons.

The staff task to be automated would include the subtasks of:

1. Check tickets,

2. Validate tickets,
3. Identify fake tickets,
4. Identify passengers who have purchased the wrong tickets,
5. Apply fines and penalties depending on the offence,
6. Report the numbers to the ticket office.

If the system owner requires the above tasks to be automated, then they will have to detail the work process of each task.

5.1.2 Analyse and Identify Interacting Subsystems

Once a task process has been identified it is important to analyse and create a map of all interacting subsystems in the task. This could be done as a simple use case diagram showing each subsystem as an actor. Although it seems like a simple process, this step is quite critical to develop the appropriate automation system.

Continuing from our example above of a ticket checking system, a ticket checker would be interacting with two or three subsystems depending on the location of work. At a station the staffs responsible acts as barrier between the station complex and the platform. At peak times the efficiency of the station staff is critical in ensuring that all the passengers make it to the platform quick and safe. On trains ticket checking staff need to cover the length of the train and check and validate tickets for newly boarded passengers at different stations enroute. So overall we can identify the following subsystems

1. Station,
2. Platform,
3. Rolling stock (trains).

5.1.3 Establish an Automation End Goal

In terms of automation it is important to define what the end goal is. A proper requirements document elicited from the stakeholders will set out clearly achievable end goal. Also for designers it is important to understand what level of automation is required for achieving the desired result. In the case of the automatic ticket gate, this term can be used to describe the process being automated.

5.1.4 Identify Factors that Affect the Variability

Some tasks will always contain within them various conditions for variability. From the ticket barrier example, the critical factor for the functioning of the automation task would be the time taken for processing each passenger who uses the system. Something tedious and slow will make the passengers impatient and cause them to rush onto the platform which at peak times can be dangerous. The other factor critical would be the procedure to handle tickets that are invalid, so that people queueing behind the passenger are not held up for a long time. Tabulating these factors and modelling a regression to identify the distribution will help in improving the accuracy of the testing to be done later.

5.1.5 Create a Level of Automation Plot

Once the subsystems, factors and goals have been identified it is critical to describe the system in terms of how much sophistication is required to achieve the task. To do this the designers have to classify the automation task into data acquisition, data analysis and action selection and implementation segments. And on each segment describe how much responsibility is given to the computer/machine. This basic construction can aid designers in understanding the difference between basic subsystems with no computer interaction to the one where most tasks are done using computers.

5.1.6 Test the Levels for Distinction

Finally, in order to verify if the automation is solving the problem it was designed to solve, it is important to construct an accurate simulation model and possibly test the automation solution at a suitable location.

5.2 Automation on the Railway

Automation has played a major role in Railway Engineering ever since the Armagh Railway Disaster of 1889 (Rolt, 1966). The disaster contributed to the framing of the first Railway Regulation Act (1889) that required the railway companies to provide three essential safety requirements (often referred to as 'lock', 'block' and 'brake'). The companies had to (a) ensure adequate interlocking between the signalling and points (lock), (b) provide safe separation of trains (block) and, most importantly, (c) provide an inexhaustible braking system on their trains (brake). Each of these provisions has demanded the use of some form of automation to guarantee safety.

Interlocking has advanced from basic mechanical interlocking, signals and points interconnected with each other using a single lever, to the modern forms of solid state interlocking, where logic for an array of points is programmed into Electrically Erasable Programmable Read-Only Memories (EEPROMs) and executed. Safe separation distance between trains, which was maintained in earlier days with a single railway policeman with a lamp, has now evolved into a highly automated system that allows for dynamic distance separation between trains. Automatic Brakes were probably one of the first automation systems introduced on the railways. These were air pressure brakes that would activate if there was any loss of pressure along the length of the train. Although braking technology has progressed significantly since then, with the introduction of eddy current based brakes for example, Compressed air brakes are still the standard (RSSB, 2011) for fail-to-safe brakes maintained even today. Automation in general has extended into other areas of the railway system such as door operation, train driving, condition monitoring, and revenue management.

Mainline railways, unlike metro systems, are complex socio-cognitive systems. That is, they involve the use of both humans (drivers, controllers, station managers and designers) and machines (point machines, signalling centres, solid state devices and trains) so as to produce a single output, which is a journey on such a system. Automation of such a complex system requires an understanding of how such systems interact with their various interfaces, most prominent of which is the human interface with this system. Therefore, it is necessary to delineate the primary activities. A railway system can be defined as

"A wheel on rail system, that provides transportation of a person, or a commodity, from source to destination with a predefined level of repetition, punctuality and security."

Using this definition, it would be prudent to infer that the goals of a railway system are to:

- Provide the capacity to carry a person or goods,
- Provide safe passage of its occupants across the system,
- Adhere to a predefined schedule for running more than one service.

These three goals can be further elaborated into the following subsystems:

- Rolling stock (trains), to provide seats and carry freight,
- Command and control systems that can detect and route trains to the right destination,
- Stations, so that passengers can alight or depart on a journey,
- Infrastructure to establish and maintain track, signalling and power systems for the rolling stock.

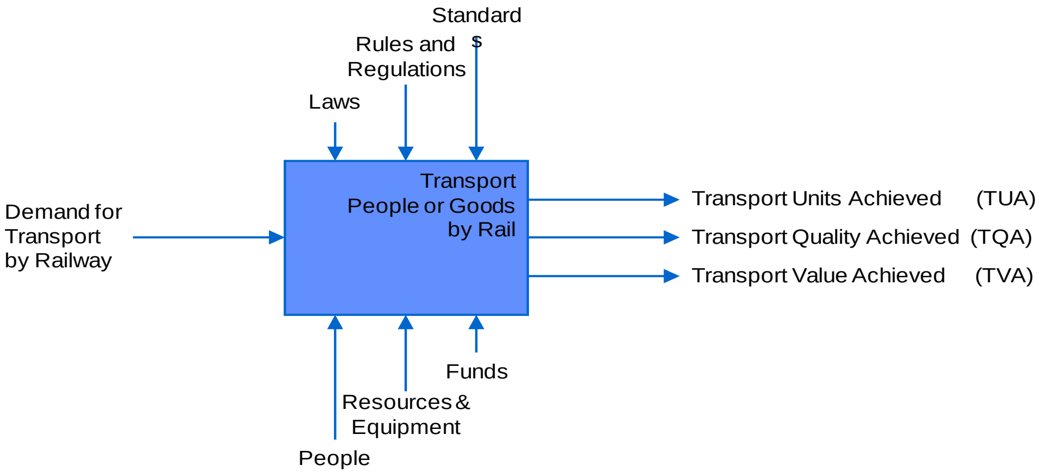


Figure 27: IDEF0 notation for a railway system (Schmid, 2001)

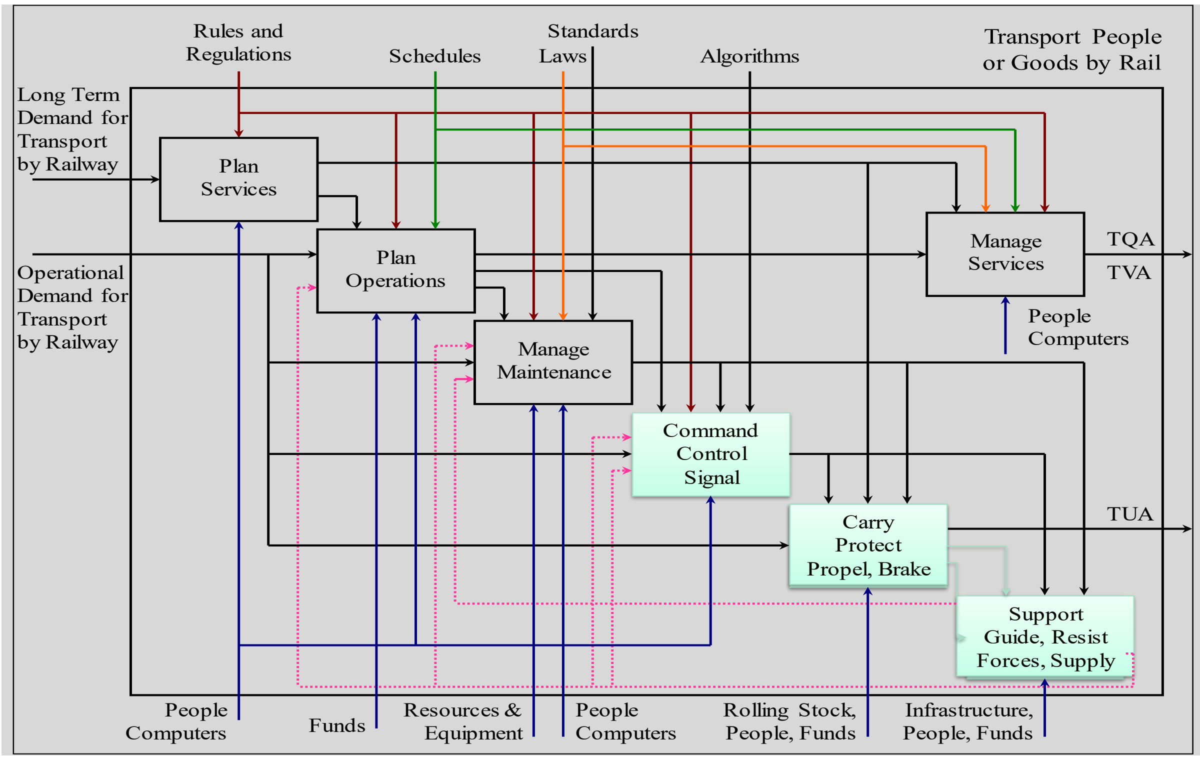


Figure 28: IDEF1 diagram expanded to show entire railway subsystem (Schmid, 2001)

Figure 27 and Figure 28 show the entire railway system as an IDEF0 and an IDEF1 notation diagram. The Figure 27 shows the major inputs and outputs for the railway system, with the controls and resources being applied on the railway environment. The diagram is an overview of the system from a functional view and hence has quite high level inputs. Figure 29 on the other hand shows an expanded view of the railway system showing the transition from the overall controlling diagram of the railway to specific relations between various subsystems. For the purpose of a future discussion it will be useful to specify that for my thesis TUA would be the capacity achieved for the railway system in trains per hour (tph).

Automation of a railway can varyingly be applied across the whole subsystem. Automation of a railway would involve the introduction of mechanisation or computerisation into the various activities that influence or govern the operation or performance of the above subsystems. In the figure 7 above three main subsystems are highlighted. These are:

1. Command, Control and System: This is the central command for the railway systems that ensures safe movement of trains. It encompasses the Train detection, protection and the communication systems employed on it.
2. Carry, Propel, Protect and Brake: This is the functional representation for the rolling stock. This type of definition is better to enumerate the chief functions of the physical sub-system.
3. Support, Guide, Resist Forces and Supply: This is the infrastructure section of the railway, the construction of which is central to the running of a smooth railway.

The following sub sections show a brief overview the various subsystems and their progression from simple manual systems to fully automated systems. Detailed breakup of the various technologies, their functionalities and some physical manifestations will be discussed in the following sections of this chapter.

5.3 Control, Command and Communication subsystem

Control, Command and Communication systems (CCCSs) provide for the safe, timely and efficient operation of trains on a railway. CCC subsystems have the following chief tasks:

- Detection of trains on the infrastructure,
- Analysing conflict of resources between trains,
- Allocating resources at the right time to each train,
- Ensuring there is a safety margin between consecutive trains,
- Communication of information on the railway to each type of train.

The railway control functionalities have evolved as a result of various additions and adaptations made to the CCC System over the last century (Chapter 2). Various systems can be employed to achieve the tasks mentioned above. These systems are:

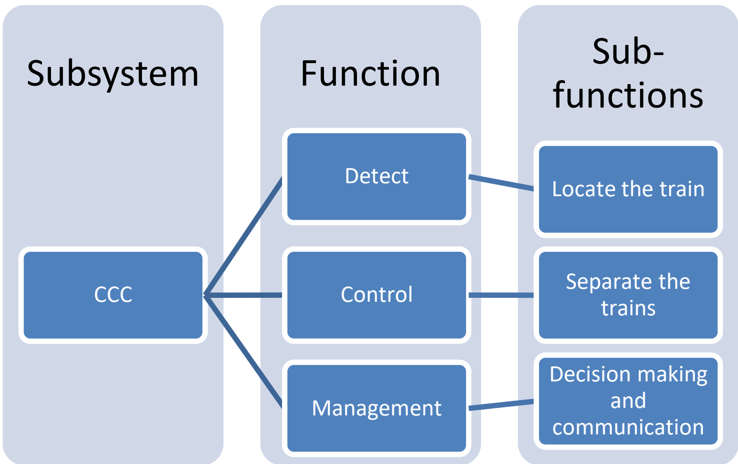


Figure 29: CCC subsystem and subfunctions (source: author)

5.3.1 Train Protection Systems (TPS)

These are systems that protect the movement authority of trains so that the safety of the system can be maintained. The TPS consist of a detection system followed by a data transmission system.

- Train stops: - These are mechanical stops set along the sides that trip the train’s brake if it exceeds movement authority (signal). Although it is an efficient system it suffers from need to be maintained regularly to ensure reliability.

- Induction based protection: - Use of magnets for AWS and TPWS to read the aspect ahead for acknowledgement from the driver. If the information is not acknowledged the train's braking system is triggered. This provides for more efficient data transmission to the train.
- Radio based protection: - Using radio transmission it is possible to track distances between trains and enforce moving block restrictions so that more trains can be accommodated onto the network. The European Train Control System (ETCS) relies on GSM-R transmission for train reporting and transmits movement authority.
- Autonomous Protection: - These are ATO systems that calculate the train position and speed and transmit it back to the control centre. They receive information continuously from other trains on the route and vary the driving accordingly.

It is important to mention that a chief function of the CCC is to maintain control over the separation of trains on the infrastructure. Chapter 2 describes the progression in the philosophy for controlling trains from fixed block to moving block systems.

- Fixed Block: Static block systems such as four aspect (or lower) blocks that ensure separation but are only able to provide for block occupation and with specific control areas set by operators at the control centre.
- Fixed Block with beacons: They are still fixed block control systems but the blocks can be shortened because the information available is a bit more continuous. Such systems have a cellular communication via beacons to the control centre that can transmit specific advisory to the trains. As such no train to train negotiation is possible.
- Moving block: The removal of signalled blocks enables a greater degree of flexibility between trains. The trains receive and transmit location information. A fixed braking distance separation is enforced from the train ahead and behind. The train control could be considered automatic but critical sections and negotiations between trains still must be mediated by the control centre.
- Operationally intelligent control: The moving block system with train to train communication that allows them to interact and negotiate safety distances between the trains depending on operational needs of the railway system.

5.3.2 Train Detection systems

Detection systems are crucial to identifying the position of the train on the infrastructure. Based on this the control of a railway system is feasible. Types of train detection system are

- Manual reporting: Early railways had a visual confirmation of a train passing a junction box and the location being telegraphed or telephoned to the box ahead. With an increase in traffic such a system becomes unsafe and inefficient.

- Track circuits: These are large electric circuits that are closed with the passage of a wheelset over them leading to detection. With the introduction of electric traction, frequency based track circuits have become necessary.
- Axle counters: These machines count the number of axles that enter and exit a section. The advantage offered over the track circuit is that sometimes track circuits depend on the axle resistance. If this changes due to environmental factors a track circuit might overlook the train and fail to detect it. Also, axle counters require minimal maintenance and installation is relatively simple.
- GPS based location: Previous systems relied on detecting a train but the introduction of faster ETCS based system requires active position reporting. A transponder installed on the train transmits the current position and speed to a central Radio Block System (RBS). This system provides the greatest accuracy with respect to location.

The position of the train on the infrastructure is relayed back to the control centre. It is the sensing requirement of a CCC system to have an accurate location description for each train along the line. This is vital information that enables real time monitoring of train movements. Various positioning systems are available but for the purposes of automation it would be necessary to move to an arrangement where the position data is transmitted autonomously. We can categorise such movements into three separate levels.

1. Track based detection: This is possible by employing track circuits and axle counters. The information is relayed back to a regional control centre using cables. When a block (refer to fixed block systems in chapter 2) is occupied the signal for the train behind is automatically set to red. No direct communication between trains is possible.
2. Continuous Position reporting: Using cellular technology it is possible to broadcast to a control centre the GPS position of the train back to the control centre. Track based detection aids in position reporting also as a reliable second channel.
3. Vehicle position reporting: The previous technologies relied on information about the position transmitted to other trains on the infrastructure through a control centre. Higher bandwidth and range on radio transmissions would enable train to train communication.

For each of these systems the automation diagram below shows the degree of automation

5.3.3 Traffic Management System

Traffic management systems on the railway function to resolve resource conflicts and route trains efficiently through the network. In order to maximise resource availability, it is important to assign a route to a train as late as possible and release it as early as possible. This is done to maximise the availability of the system. The types of route setting systems are

- Manual set and release systems with the junction signaller setting a point and then a signal. The disadvantage is that the signaller must confirm visually or orally the position of a train near the route entry point.
- Manual Traffic management (CTC): - Once relay based interlocking had been introduced, it was possible to "mimic" a local track layout and specific routes can be set by pushing buttons on the mimic board. The route can be cleared by a simple reset button. Such a system allows for the development of central traffic boxes that covers large areas on the network.
- Rule Based Traffic Management: - The introduction of Solid State Interlocking (SSI) with computers allows for programming default routes. This is much faster than previous systems and the controller needs to only intervene when there are significant conflicts due to delays.
- Autonomous Traffic Management: - Rule based systems can be further refined to set routes based on a timetable. Such a system records the strategies for conflict resolution from a human and employs these techniques to resolve conflicts autonomously.

The route management systems automatically progress from a manual system to a completely automated system. This enables the introduction of complementary systems that can enhance the efficiency of this system.

Table 7 shows the CCC systems of a railway.

Table 7: Progression of systems in the CCCs of a railway

Train Detection	Train Protection	Traffic Management
Manual	Train stops	Junction box based TM
Track circuits and axle counters	Induction based	Manual TM
Radio based detection	Radio based	Rule based-TM
	Autonomous	Autonomous TM

Traffic management systems on the railway function to resolve resource conflicts and route trains efficiently through the network. In order to maximise resource availability, it is important to assign a route to a train as late as possible and release it as early as possible. This is done to maximise the availability of the system. The types of route setting systems are

- Rule Based Traffic Management: - The introduction of Solid State Interlocking (SSI) with computers allows for programming default routes. This is much faster than previous systems and the controller needs to only intervene when there are significant conflicts due to delays.
- Intelligent traffic Management: - These systems sense automatically and acquire information about the railway system. These systems use advanced control algorithms to provide a degree of flexibility to the rule based traffic management systems. But the problem solving is limited

to a local geographic area. This system is still human supervised and troubleshooting is carried out accordingly.

- Cross network intelligent traffic Management: - Intelligent traffic management systems can be further refined to set routes based on current service demand. Such a system records the strategies for conflict resolution from a human and employs these techniques to resolve conflicts autonomously.

From the above three sub function systems we can arrange each of the CCC systems into separate progression of automation levels.

Level	Detecting Trains	Train Control	Traffic Management
1	Track based	Fixed Block	Rule Based
2	Continuous positions Reporting	Fixed Block with beacons	Intelligent
3	Vehicle position reporting	Moving Block	
4	Operationally intelligent control		Cross network intelligent traffic management

The Figure 30 below outlines the automation progression for each level of automation shown in the above table. The automation diagram has four main columns

1. Information Acquisition,
2. Information Analysis,
3. Decision Selection ,
4. Decision Implementation.

For each of these actions we can specify a grade of automation.

1. Manual: Requires Human with machine only displaying information,
2. Semi-Automatic: Human selects from a list of choices provided by the machine,
3. Automatic: Machine automatically selects the action and the human only intervenes in critical situations,
4. Autonomous: Machine requires no human intervention.

Based on these above parameters we can specify for each level

1. Level 1: Information acquisition is done by the machine, Analysis is done by the machine with limited human interaction but the selection of the course of action is done by a human in critical cases.
2. Level 2: Information acquisition and decision selection is done by the machine but only in regional areas. It is still offered as a choice to the human to select the recommendation to be communicated back to the trains.

3. Level 3: Information acquisition, analysis and decision selection is done by the machine but only in regional areas. It is still offered as a choice to the human to select the recommendation to be communicated back to the trains.
4. Level 4: Information acquisition, analysis, decision selection and implementation are completely autonomous.

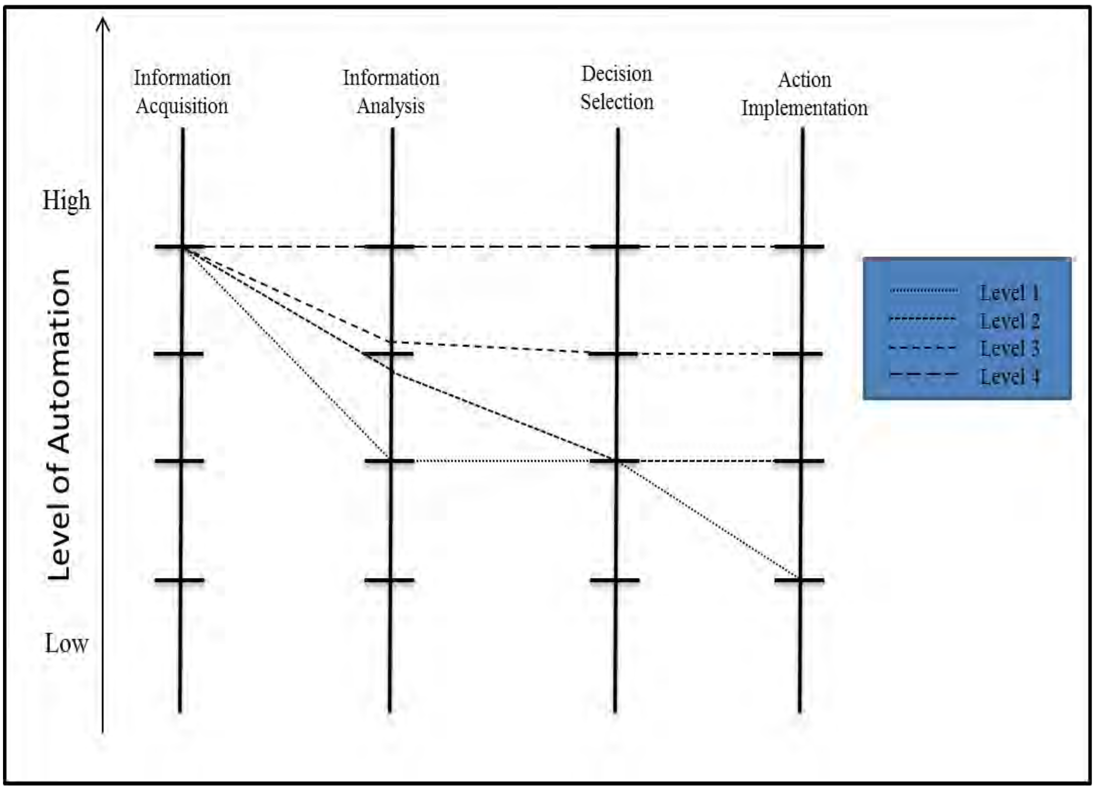


Figure 30: Grades of automation to describe automation levels in CCC

5.4 Rolling Stock

Rolling stock is a critical component of a railway system as it carries within it the passengers and goods on a journey. The chief activities on this subsystem are:

- Driving: to engage the motors and move in a safe and efficient manner,
- Dwelling: to stop at required stations or freight yards to allow the exchange of passengers and goods.

All these activities can be automated at various levels. EN62290 provides a standard in order to automate driving (BSI, 2006) and control. An Automatic Train Operation (ATO) system is used centrally to control train movements from a driving perspective. Table 8 lists the features of an urban railway system with reference to the equivalents of a mainline railway system. The objective of running a mainline railway is different to that of an urban railway system. The mainline operator is chiefly concerned with

- Achieving uniform journey times to reduce the variability of arrival and departure times. This also reduces the need for allowing a large margin in the timetable for recovery from delays.
- Achieving lower carbon targets as energy consumption is a critical issue for mainline operators (electric or diesel traction). Reducing energy consumption is not only beneficial to the environment but also to reduce operating cost.

Table 8: Comparison of features of Urban and Mainline Railway systems

Urban Railway Systems	Mainline Railway Systems
Little or no mixed traffic	Mixed traffic (freight & passengers)
Short end to end journey times, not exceeding an hour in most cases	Long end to end journey times
Intensive passenger operations	Demand based operations (high density near urban centres)
Highly monitored infrastructure	Fenced but unmonitored infrastructure
Simple track alignments	Complex layouts with remote but highly sensitive junctions
Homogeneous rolling stock	Heterogeneous rolling stock

Ideally mainline operators would like a satisfactory target which is a blend of the above mentioned objectives. This can be achieved by controlling the driving style. Driving on a mainline can be categorised into the following levels.

- Driver with Standalone Driver Advisory Systems (SDAS) – The driver is completely in control and receives information from a display unit regarding suggested driving styles. It is ultimately up to the driver to follow the advice. The driver is completely in control but maybe supervised by a warning system, a train stop or an ATP system to protect against exceeding movement authority around signals and speed limits.
- Driver with a Networked Driver Advisory System (NDAS) –The driver is provided with up to the minute information about the traffic situation. The system expects the driver to driver according to the pace set by the controller.
- Driver Attended – The train is driven by the driver, but the machine enforces speed limits and corrects the driving style to match the operational demand. The system expects the driver to drive according to an optimised trajectory calculated by the controller. The trains are equipped with an interventionist computer that enforces movement authority instructions from the Automatic Train Protection (ATP) system, actively braking the train when it over speeds or is exceeding the advised speed.
- Driver Unattended – The system is completely standalone and can run without any intervention from the driver. These are ATO equivalent systems where the driver is considered as a supervisor and only intervenes when the system is in a faulty condition. In such an arrangement

the driver could be situated local to the cab or driving from a remote location. The on-board system provides speed demand for the train control systems and uses the ATP equivalent system to provide safety.

- Autonomous – The trains are completely run by computers with no human intervention. An ATO equivalent system is integrated into ETCS system and drives the train without any need for supervision.

Table 9 shows the above mentioned levels side by side as complementary levels for automation from the perspective of Driving.

Table 9: Methods for automation with different levels of driving systems on trains

Level	Driving
1	DAS
2	Attended
3	Unattended
4	Autonomous

Like the case of CCC we can show various grades of automation using an automation diagram. The levels of automation and columns are still the same.

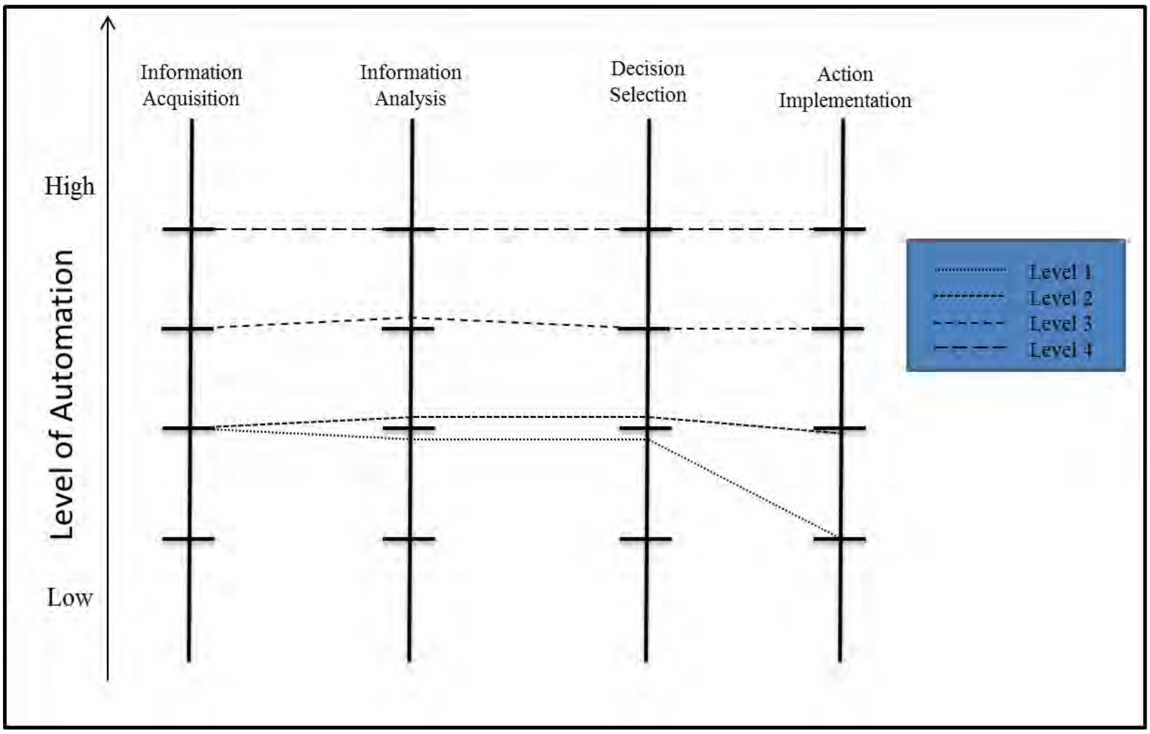


Figure 31: Grades of automation to describe automation level on driving systems

5.5 Platforms

Platforms are critical areas for automation as they involve the transfer of passengers. Delays accrued on the platform are a major contributor to the performance of the railway. Platforms are required in

5.5.1 Passenger Guidance Systems

Passengers need to be able to enter and exit trains at platforms. The key to maintaining an operation time is to enable passengers enter and exit the trains as quickly as possible so that the trains can depart on time. This can be enabled by

- Manual door operation: the doors are controlled by the driver or train staffs, whose responsibility it is to ensure that passengers are offloaded, and the doors are clear for the train to leave. The responsibility for the safety is completely shouldered by the train staff.
- Automatic door operation: the doors are released automatically on arrival in the station and close at a predetermined time interval. Interlocking with the train traction is implemented so that trains cannot leave with a door open. The system assumes complete responsibility for the safety of the passengers. The information about door open and close is released by the driver upon arrival at the station.

5.5.2 Train Dispatch

Once the train is closed for departure the dispatcher needs to be notified so as to clear the route in front. This can be done manually or automatically.

- Manual dispatch: - On platform staff verify the train is ready to depart and indicate to the driver and dispatcher. The on-platform staffs ensure that the trains leave on time and that there is no visible obstruction to the trains' movement.
- Automatic dispatch: - Once the train is ready to depart the driver could request to leave and transmit the information to the dispatcher who would release the route to the driver. Platform screen doors (like on metros) can be used to ensure that the train is clear and dispatch information directly transmitted to the train.

5.5.3 Passenger Management Systems

With overcrowding on the platform there is an added risk of a passenger ending up in a dangerous situation (like stuck in the door, fallen into the gap between platform and train or other serious emergencies on the platform).

- Platform staff will be required to monitor crowding situations and thereby closing off platform access to ensure safety.
- An active monitoring system that senses the crowding level in platform and changes the state of systems to control the crowd. This could be from changing directions on escalators to activating full evacuation systems to control a fire and alerting a station staff for assistance if there is a passenger related issue.

Freight trains also require management at yards for loading and unloading of goods at terminals and for quick turnarounds. Like platform systems, the offloading and loading can be automated.

Table 10: Automation increase at the platform train interface

Platform Management	Passenger Guidance	Train Dispatch	Passenger Management
Manual	Manual Door operation	Manual dispatch	Platform staff
Automatic	Automatic Door operation	Automatic dispatch	Active monitoring

5.6 Summary

This chapter has presented an outline of the framework being used to test automation. The first steps of the framework require us to identify the task for automation. The second step of the framework is to analyse and identify the sub-functions of the task. The latter half of the step dealt with identifying associated subsystems which execute each sub-function. The third step is to establish a roadmap for the automation in stages with each step of the roadmap being linked to unique automation levels. The fourth step is to identify all the factors that might influence the outcomes for automation. The fifth and final step is to test each level of automation for distinction.

The automation of a mainline railway is then discussed with levels and stages of automation being discussed for each individual subsystem. The subsystems include Control and command system with train protection systems, train detection and traffic management systems. The rolling stock (Trains) with driving and door operation functions. The platform interface with the passenger guidance, train dispatch and passenger management being broken down into distinct levels.

In the next chapter, I shall cover the testing of a set of technologies based on increasing levels of automation. The technologies are described, and a stages and levels of automation is drawn up for them using the framework described in this chapter. The following chapter will cover the testing of the technology sets and the results from the testing in detail.

6 Testing the Framework

This chapter outlines the design of experiments space for testing the framework. It is primarily consists of two parts.

1. Setting up the simulation for testing the framework,
2. Discussion on the results obtained from the testing.

The first section will discuss the design of the experiment, the setup, the various terminologies and nomenclature of the levels being tested. The second half discusses the results of the experiments from the observations made.

6.1 Designing the Experiment

In the previous chapter we developed a framework for levels of automation. Due to the large number of variables and factors in that system, it is impossible to control for all inputs factors and variables. Therefore, by constructing testable levels of automation using the guidelines from the developed framework, it is possible to construct a simulation experiment using Quasi-Monte Carlo (QMC) implementation. The simpler construction will allow the reader to better understand the shift in the system when changes are made in individual subsystems.

6.1.1 The Experiment Framework

The automation level framework shall concentrate on three major sub systems

6.1.1.1 Signalling

Signalling level shall be represented by four signalling technology types:-

- 4 Aspect signalling system,
- ETCS Level 1,
- ETCS Level 2,
- ETCS Level 3.

With respect to ETCS Level 1, it must be noted that, during the experiment process, the distance from the signal to the balise was assumed to be 100 m. This was done to make it easier to construct a test case for the ETCS Level 1 signalling category. On the real railway, ETCS Level 1 has the balise positioned adjacent to the signal. The results presented in this chapter, based on the spacing stated above, suggest a wider spread of the calculated journey times than that of the journey times for the 4 aspect signalling system. However, the results when corrected would not be entirely divergent from the results shown. In subsequent publications, following this thesis, the error has been corrected to reflect the actual ETCS Level 1 layout.

6.1.1.2 Driving

Driving tasks include two levels of automation. We can have a third level for showing supervised driving, but for the distinction to remain clear in the experimental data we have chosen the two extreme cases.

- Manual Driving,
- Autonomous Driving.

Both these types of driving are implemented using software. Therefore, for the manual driving style the system produces a curve that constantly overshoots and undershoots the target speed limit, in approximation of a human style of driving. An autonomous driving case assumes that the train traction to the maximum speed limit as quickly as possible and brakes as late as possible to preserve capacity.

6.1.1.3 Platform supervision

Platform supervision will be done on three levels of automation. For the first level there are present dedicated train staffs to open and close the doors, whereas for the second level the departure is completely controlled by staff on the platform. The third and final level contains a completely autonomous solution.

- Train staff supervised,
- Station staff supervised,
- Automatic through platform screen doors.

For the above progression for each level of automation, Table 11 shows a construction for the experimental conditions that are necessary to be tested for the framework. It is important to note that we can discount certain combinations as being impractical for implementation.

Table 11: Experiment framework for testing (author, 2018)

<table>
<tr><th>Framework testing level</th><th>Signalling</th><th>Driving</th><th>Platform supervision</th></tr>
<tr><td rowspan="4">1</td><td rowspan="4">4 Aspect</td><td rowspan="2">Manual</td><td>Train Staff Supervised</td></tr>
<tr><td rowspan="2">Station Staff Supervised</td></tr>
<tr><td rowspan="2">Automatic</td></tr>
<tr><td>Automatic</td></tr>
<tr><td rowspan="4">2</td><td rowspan="4">ETCS 1</td><td rowspan="2">Manual</td><td>Train Staff Supervised</td></tr>
<tr><td rowspan="2">Station Staff Supervised</td></tr>
<tr><td rowspan="2">Automatic</td></tr>
<tr><td>Automatic</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">ETCS 2</td><td rowspan="2">Manual</td><td>Train Staff Supervised</td></tr>
<tr><td></td></tr>
</table>

<table>
<tr><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2">Automatic</td><td>Station Staff Supervised</td></tr>
<tr><td>Automatic</td></tr>
<tr><td rowspan="4">4</td><td rowspan="4">ETCS 3</td><td rowspan="2">Manual</td><td>Train Staff Supervised</td></tr>
<tr><td rowspan="2">Station Staff Supervised</td></tr>
<tr><td rowspan="2">Automatic</td></tr>
<tr><td>Automatic</td></tr>
</table>

For the framework above the infrastructure can be modelled as a simple network with a branch line. The line contains a total of 12 stations with 2 of them being on the branch line.

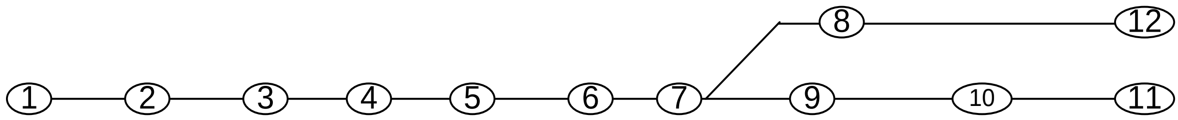


Figure 32 : Single Line for the purpose of simulation

Placing the signalling as a constant for testing the driving and platform supervision levels we can construct the following Table 12 as a condition checker that will evaluate all possible combinations of signalling with driving and platform departure sequence.

The reader shall note that a change in automation level implies that the signalling on the infrastructure has been changed. This was denoted as framework testing level in the previous Table 11.

It should be noted that in with two variables having 2 and 3 levels it would require 6 experiments to check all conditions. However, in Table 12 the experiment case for the driving level of ‘manual driving’ and platform departure of ‘automatic dispatch’ is omitted. This condition may never exist as the driver will be expected to guarantee the safety of departure which will be the same as train staff supervised platform departures.

Table 12: Combination of all possible test conditions

Level	Driving	Platform Departure
1	Manual	Train Staff Supervised
2	Manual	Station Staff Supervised
3	Automatic	Train Staff Supervised
4	Automatic	Station Staff Supervised
5	Automatic	Automated Dispatch

6.1.2 Defining the Factors

Error! Reference source not found., shows the various factors with their distributions that have been tested for regression.

Table 13: Factors contributing to delays summarised from Hansen and Pachl (Hansen, 2008, Yuan, 2008b, Yuan, 2008a)

Delay Factor	Distribution	Delay Factor	Distribution
Initial	Lognormal	Transfer time	Kernel
Arrival	Lognormal	Free running trains	Weibull
Alighting	Normal	Trains stopping at signal	Weibull
Boarding	Normal	Train detection	Weibull
Departure	Exponential	Conflict detection	-X-
Conflict Inhibition	Kernel	Route allocation	Kernel
Conflict Resolution	-X-	Route setting	Kernel

Based on the three subsystems, the factors for driving the simulation can be broken down into:

1. Signalling subsystem:
   - Route set and release times,
   - Dispatch times.
2. Driving subsystem:
   - Driving profile for the manual and automatic driving.
3. Platform subsystem:
   - Machine operation times for the doors ie. The time for releasing the interlocks and the actual door open and close times,
   - Dwell time including the arrival time, alighting time & boarding time.

From the factors identified previously we can assign the following distributions to the table below.

Table 14: Probability distributions of factors selected for simulation

Factor	Description	Distribution
Machine operation times	Distribution of machine operation times	Uniform
Dispatch Time	Departure from station as set by a station dispatcher	Exponential
Route Release Time	Control dispatcher releasing and setting the route	Normal
Route Setting Time		
Dwell Time	The overall dwell time at the platform	Normal

Figure 33 below shows the interaction between the various factors and the subsystems. The x's mark the primary impact of the factor on the subsystem whereas the o's mark the secondary impact of the factors on the surrounding subsystems.

	Arrival Time	Running Time	Door Operation	Departure Time	Route Setting	Platform Control	Platform Crowding factor
Track		x			x		
Signalling	o	x		x		x	x
Rolling Stock Doors		o	x	x		o	o
Driving style	x	x	o			o	o
Station			o	o		x	x
Crowding			o	x		x	x
Dispatcher	x	x		x	x		o

Figure 33: Interdependence between the subsystem and the factors identified. The 'x' denotes primary factors and 'o' denotes secondary impact

6.1.3 Quasi-random Number Generation and Transformation

Using Sobol' (Sobol, 2001) quasi random number generator a primary random point is generated in Matlab. Each randomly generated point is then transformed into the factor distributions as per the transformation below.

Transformation of each point x for the factor into the above specified sequences

- Uniform distribution: $X = lower\ limit + (lower\ limit - upper\ limit) * x$
- Normal distribution: $X = x * sigma + \ mu$
- Log-uniform distribution: $X = 10^{(lower\ bound + (lower\ bound - upper\ bound)*x)}$ where ,
- $\binom{lower\ bound}{upperbound} = \log_{10}\binom{lower\ limit}{upper\ limit}$
- exponential distribution: $X = e^{(x*a+b)}$

6.1.4 Simulation

The overall simulation workflow contains of a macro simulation with Matlab and microsimulation in the form BRaVE. This architecture was already previously detailed in Chapter 4.

Testing the 4 framework levels across 5 testing levels produces 20 cases. But in the railway context we need to check for how automation responds to traffic on its network. The 20 test cases will need to be repeated against a timetable.

For the purpose of simplicity and elimination of noise in the results, only a homogeneous service pattern, i.e, services containing the same stopping pattern on the railway. The other variable that is controlled during a simulation run is the speed of the service.

The speed of the service is set to two levels

1. Uniform: A uniform speed of 125 km/h is imposed on all framework testing levels.
2. Variable: each framework testing level is assigned an individual run speed shown in the Table 15 below.

The service pattern for all test cases is the same. Two services are present in the timetable. The first one originating at station 1 and carrying through the junction 'til the final station stop at station 11 on the hypothetical route shown in Figure 32. The second service route branches after station 7 and completes service at station 12. The traffic density for the service is set for two levels

1. Uniform: A timetable containing 18 trains per hour.
2. Variable: each framework testing level is assigned an individual traffic density as shown in Table 15 below

Table 15: Table to show the variable speed and traffic density levels against each testing framework level

	Framework testing level 1	Framework testing level 2	Framework testing level 3	Framework testing level 4
Variable speed	125 km/h	180 km/h	220 km/h	300 km/h
Variable traffic density	21	24	30	31

The Table 16 below shows the test cases for each major simulation run.

Table 16: Table showing the Test case levels for simulation

Test Case	Speed	Traffic Density
1	Uniform	Uniform
2	Variable	Uniform
3	Variable	Variable

There should be 4 test cases ideally with speed and traffic density each containing 2 levels. But the case of Uniform speed with variable traffic density is omitted from Table 16. This is because demanding that the network produces more traffic at the same speeds does not follow. The traffic would produce delays and all journey times will be inflated by the end of the simulation.

Increasing the speed of the trains increases the space available on the railway between trains. Therefore, the logical step to make was to increase the space in the time table, i.e., variable speed with the traffic density kept uniform. This is followed by moving to the case of varying traffic densities in test case 3.

Figure 35 shows the simulation workflow for the experiment. The left-hand side shows the overall process workflow and the right hand side shows the cascaded function calls during every iteration. It should be noted that the set function ensures that the right files are selected before running the traffic on the simulator.

The simulation workflow consists of 3 test cases, with 20 experimental case outputs each. The Figure 34 below shows the final multi-layered design of experiments for this research.

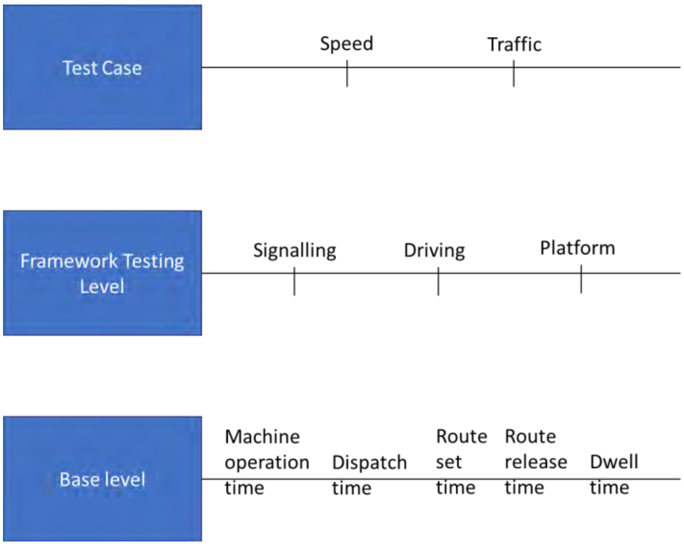


Figure 34: The overall design of experiments space for testing the framework (author, 2018)

Appendix 1 contains the software code written for conducting the simulation. It consists of code written in Matlab for the test cases, framework testing levels, and the Brave API code written in Java for the base levels.

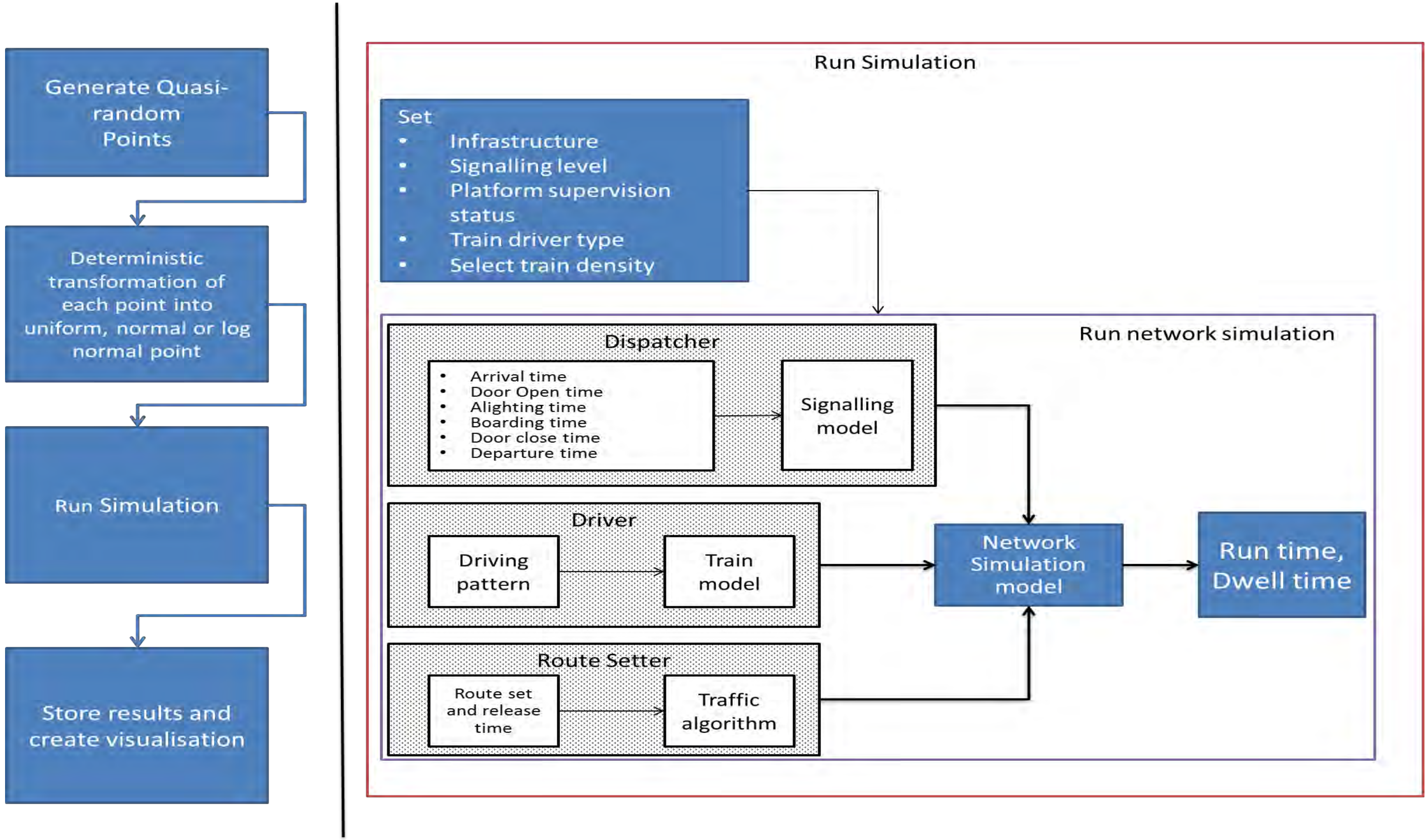


Figure 35: The visualisation for the simulation workflow for each individual run of the experiment.

6.2 Discussion of the Results from the Simulation

This section consists of three major sections. The first section discusses the data visualisation and representation. The second section discusses the results per individual test case and the final section is some additional comments and observations from the overall results

6.2.1 Visualising the Output Data

One of the early discussions during the thesis study was to find an effective visualisation to represent the results of the simulation. The visualisation was required to be simple enough for a lay reader to understand the results of the simulation and the observations being drawn without in a simple manner.

The challenge in representation lies with the multiple layers of the experiment design. The output of the overall simulation was a single point representing the overall average journey time for a train in the iteration run. The result set from a lower layer testing level consists of 2000 such points. Therefore, each framework testing level consists of 10,000 points.

There were two chief questions that required to be answered by any chosen representation.

1. Can we comment on the homogeneity of the result space?
2. If the journey time clusters around a mean for each level, what is spread of data?

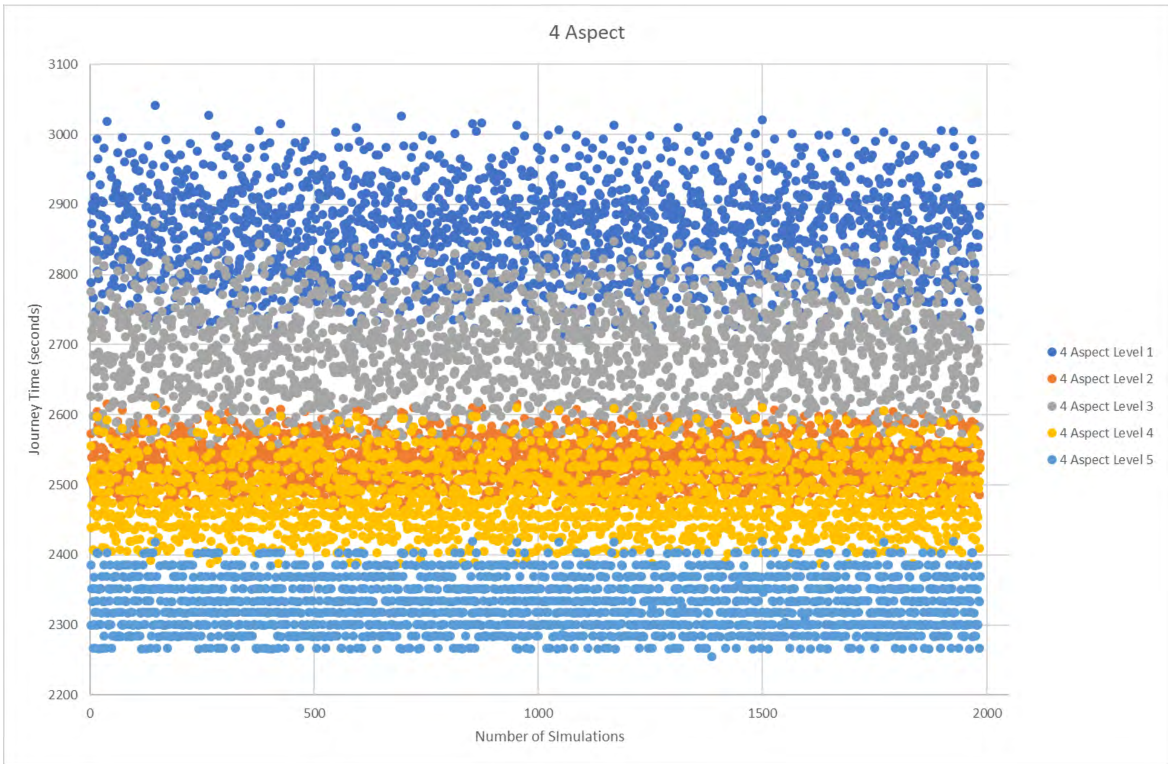
To answer question one, a simple scatter chart was chosen. Such a plot would let a reader understand very quickly if the result space was well mixed with no distinctions between each level or if each level created an individual cluster around a mean journey time.

To answer question two, a box and whiskers plot was chosen. This plot consists of a rectangular box, the area of which represents the overall spread of the data. A line in the middle of the box denotes the mean of the results and the whiskers are drawn to point to outliers.

6.2.2 Results and Discussion

The results are split into the three test cases. Each test case shows the output in a table followed by a general description of the outputs and specific comments and observations.

6.2.3 Test Case 1 – Uniform Speed & Uniform Traffic Density



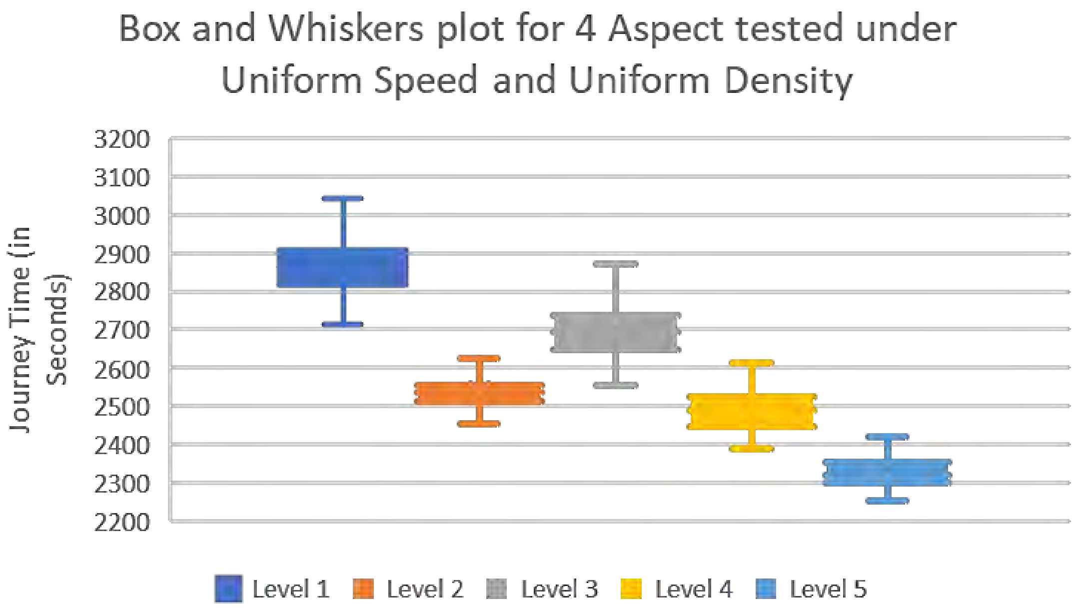
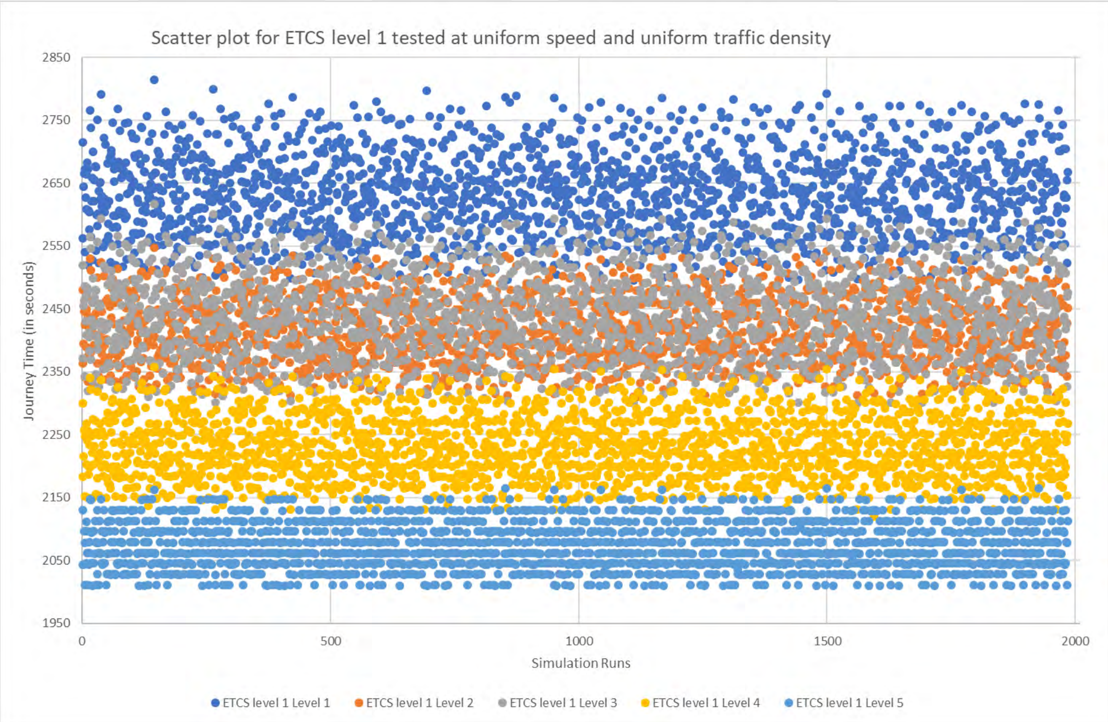


Figure 36: Results from framework testing Level 1 under uniform speed and uniform density (author, 2018)



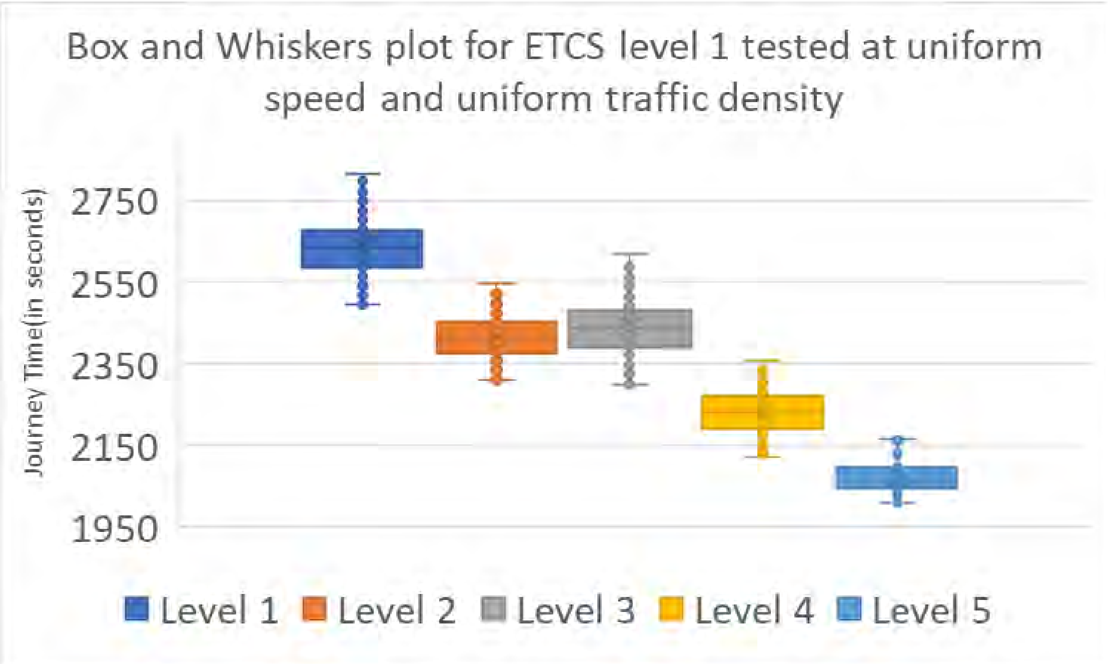
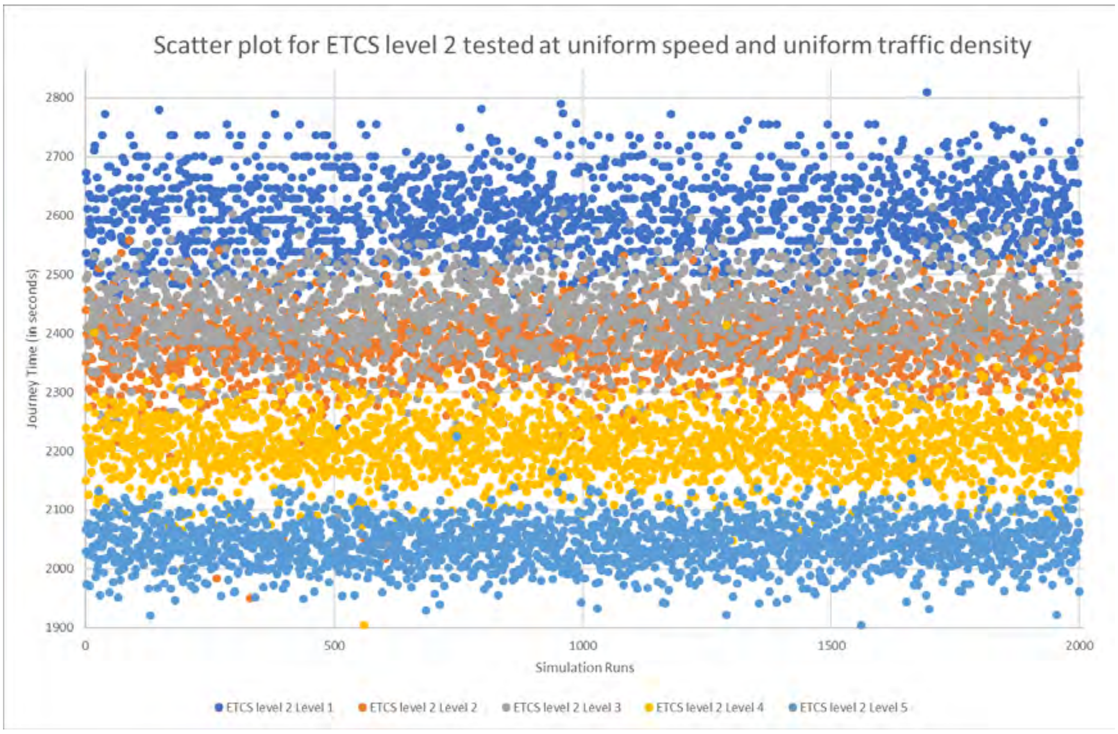


Figure 37: Simulation results for framework testing Level 2 at uniform speed and density (author, 2018)



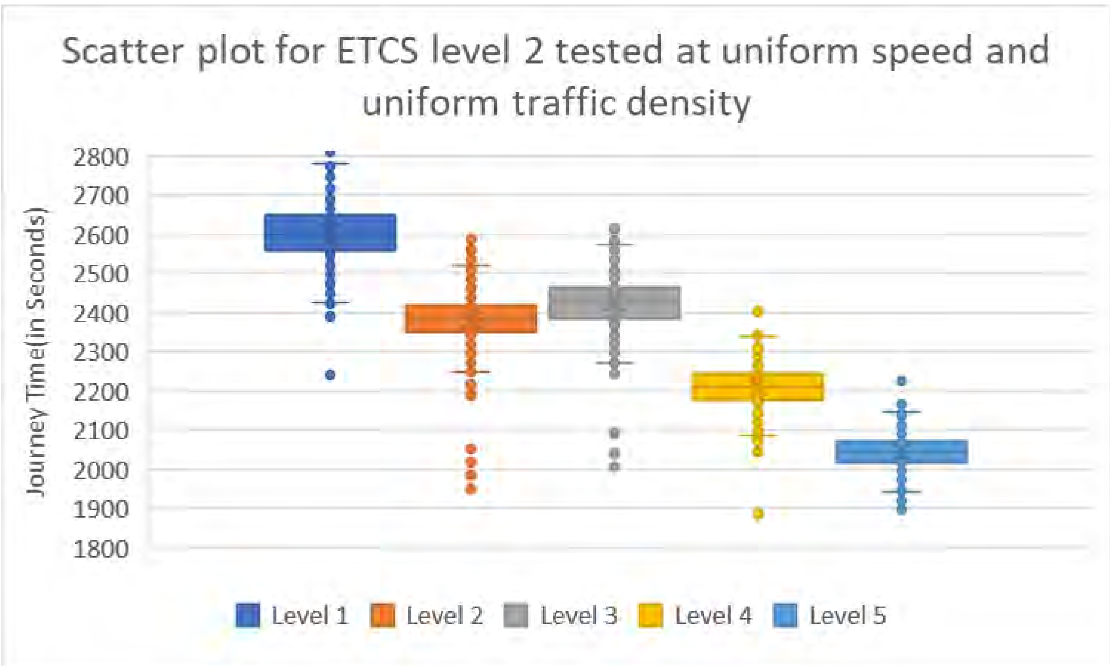


Figure 38: Simulation results for framework testing Level 3 at uniform speed and density (author, 2018)

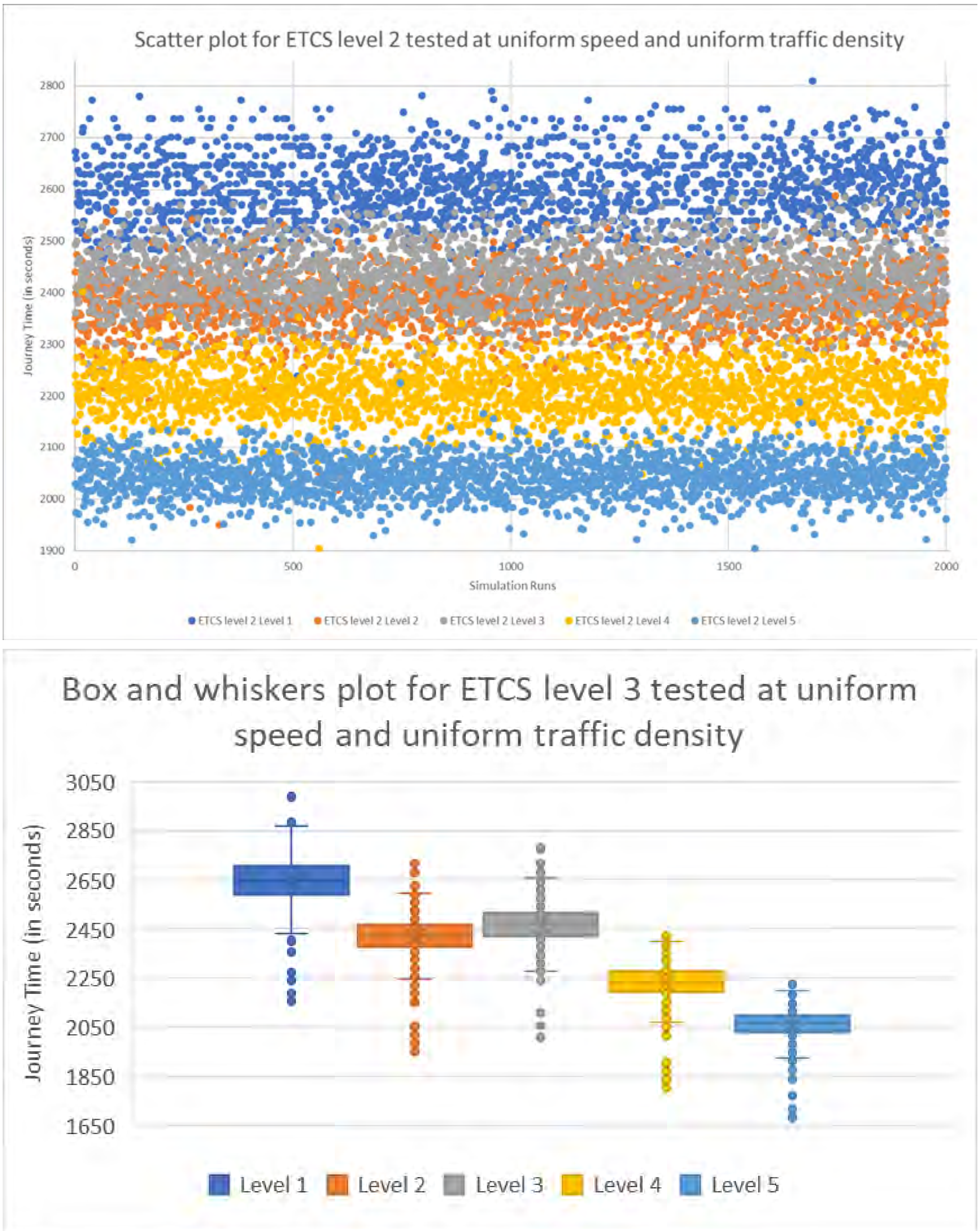


Figure 39: Simulation results for framework testing Level 4 at uniform speed and density (author, 2018)

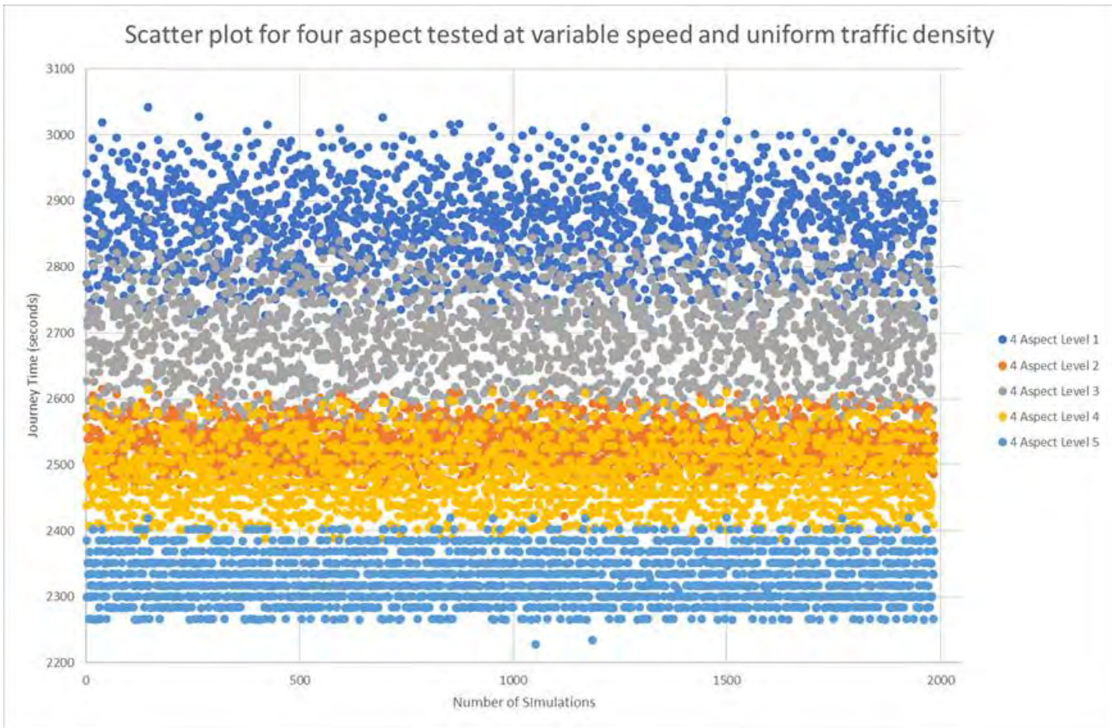
The overall spectrum of the journey times for this test case i between 2300 seconds and 3000 seconds with the overall sample mean around 2650 seconds. As expected, the level with high level of automation (level 5) is at the bottom of the spectrum (shortest journey times) and level 1 with no automation for driving or platform supervision is closer to the upper limit of journey times of 3000 seconds. The intermediate levels move show movement with change in each framework testing level (called as FT level from now).

FT Level 1: Level 4 and 2 are well mixed for this level but the interesting observation is that level 3 contains higher journey times than level 4 and level 2. We can remind ourselves that level 3 is a combination of an autonomous train on an infrastructure with train staff supervising the door. This produces a standard deviation for this level is high.

FT Level 2: Level 4 is almost clear of level and level 2 and level 3 not producing mixed results. With ETCS Level 1, the run time for the train would not include the regular loses due to speed changes as observed in FT level 1.

FT Level 3 and Level 4: There are no significant deviations from previous and the system sees a net no change in run times. This is because FT level 3 and FT level 4 are moving block systems and with constraints on speed no clear advantage appears to have been made at these levels.

6.2.4 Test Case 2 - Variable Speed and Uniform Traffic Density



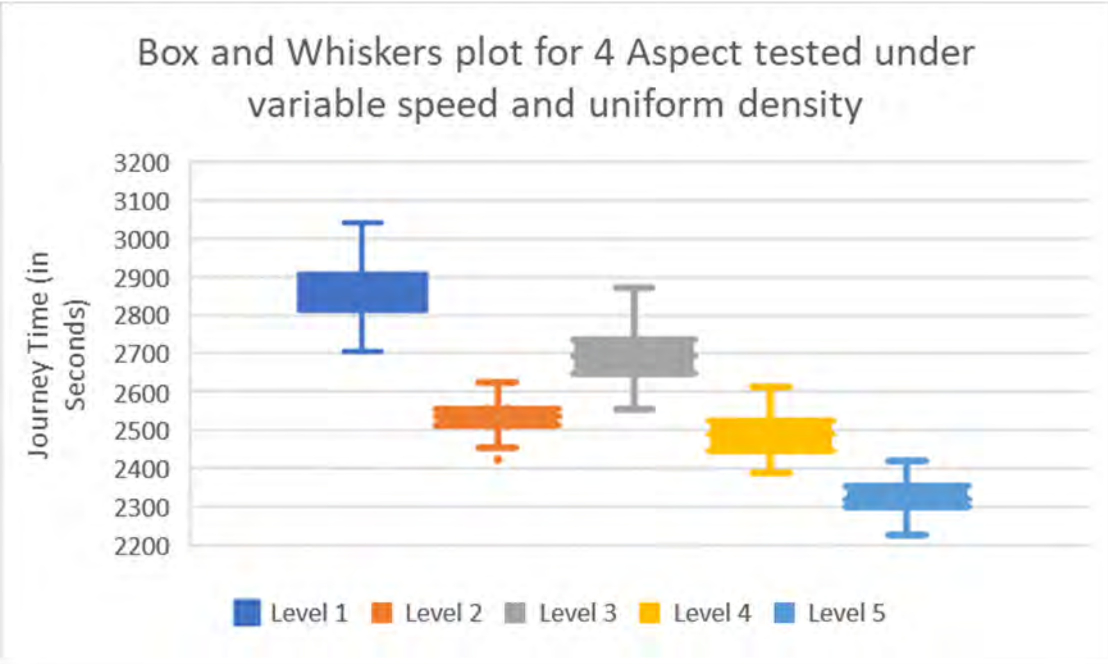
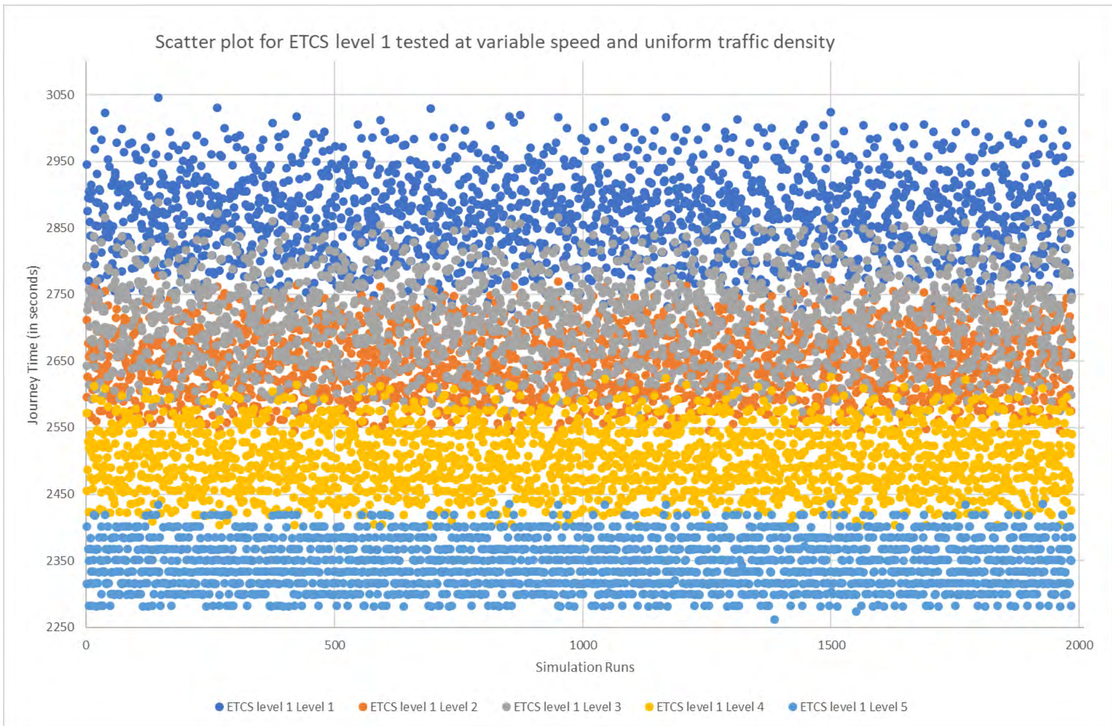


Figure 40: Simulation results for framework testing Level 1 at variable speed & uniform density (author, 2018)



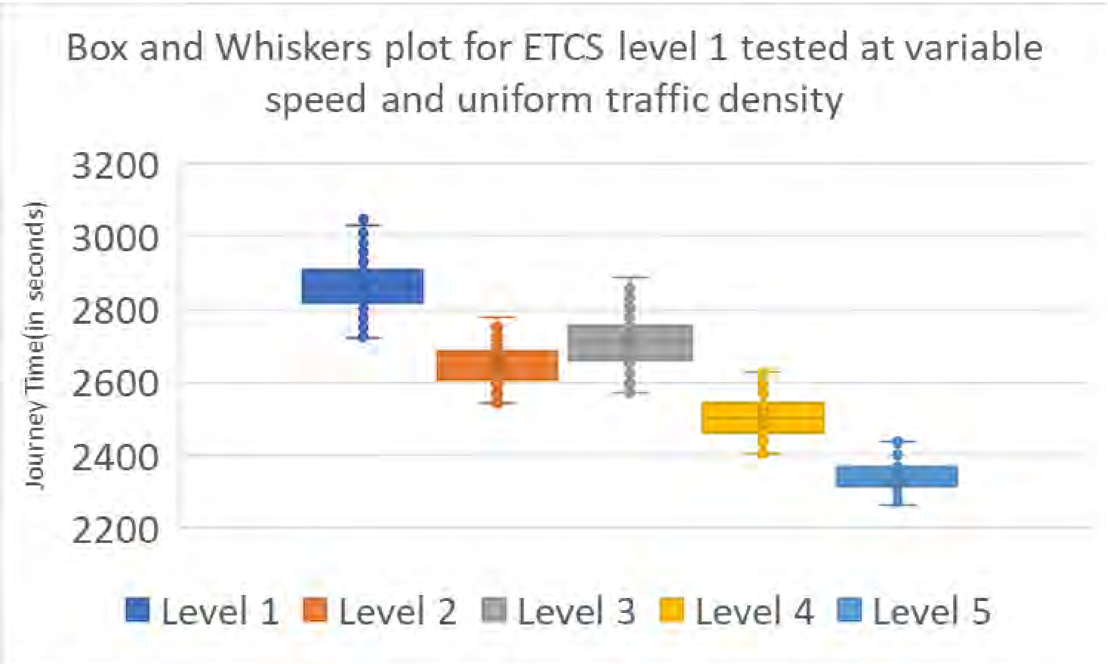
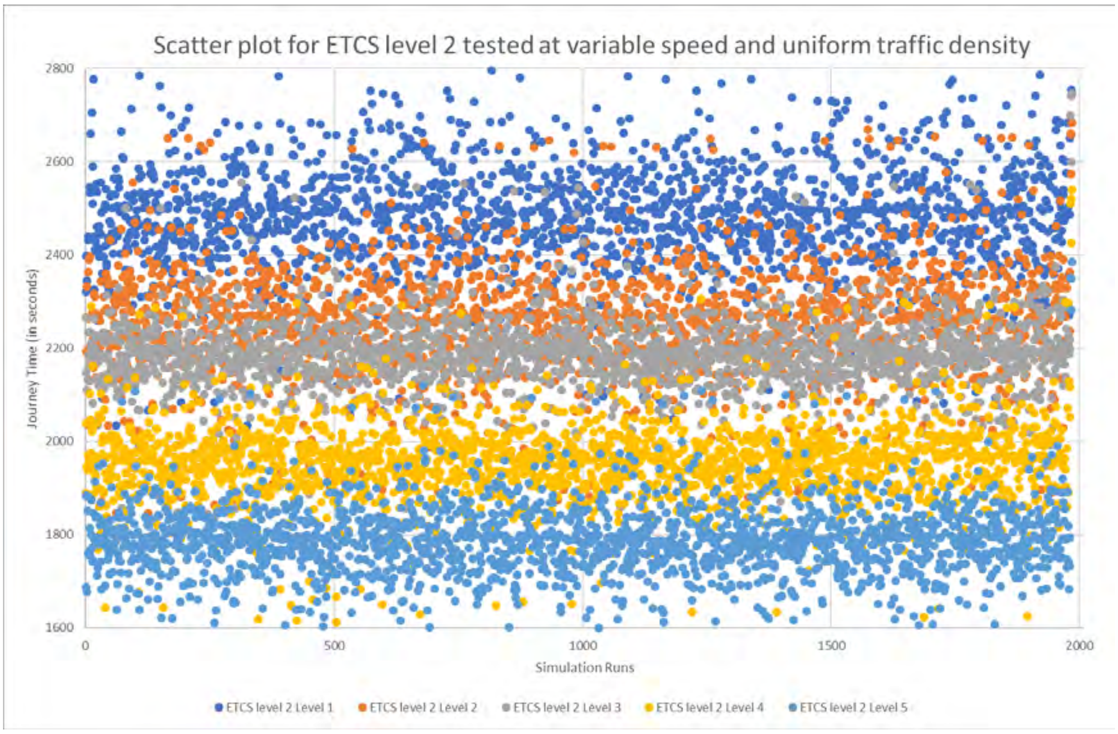


Figure 41: Simulation results for framework testing Level 2 at variable speed & uniform density (author, 2018)



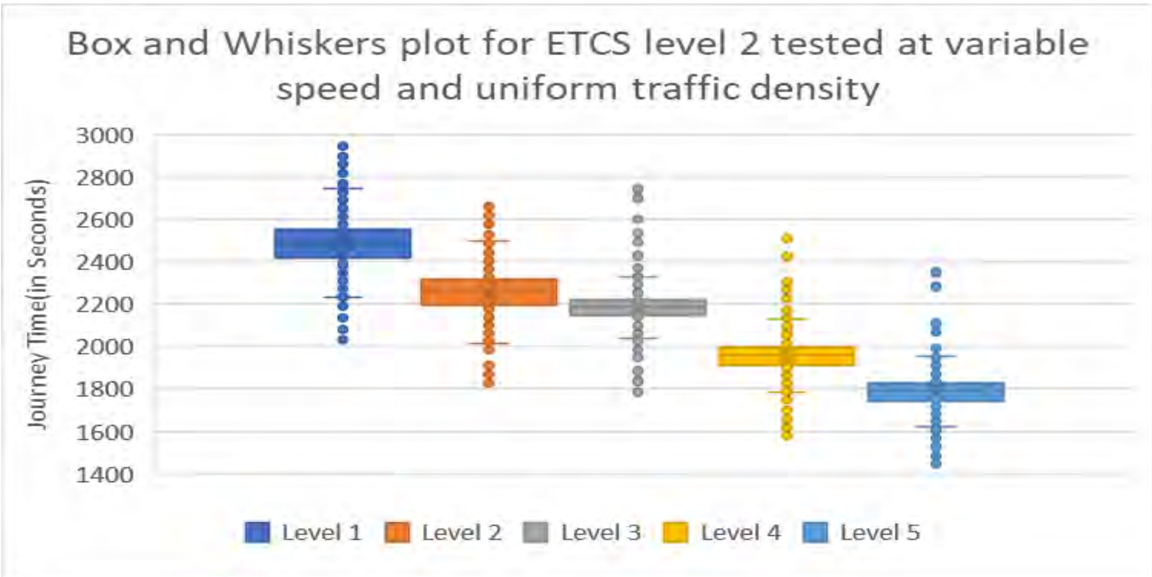


Figure 42: Simulation results for framework testing Level 3 at variable speed & uniform density (author, 2018)

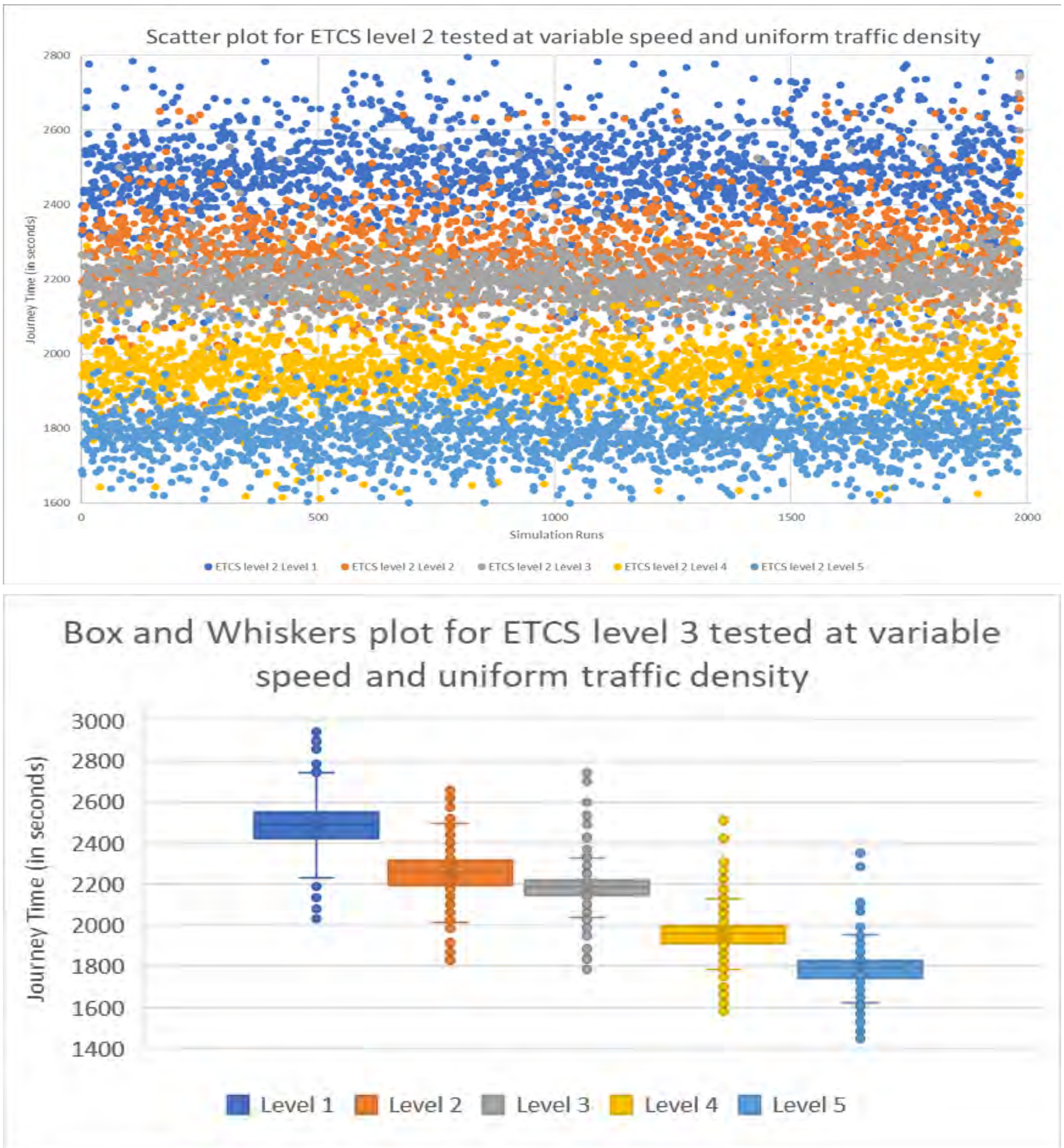


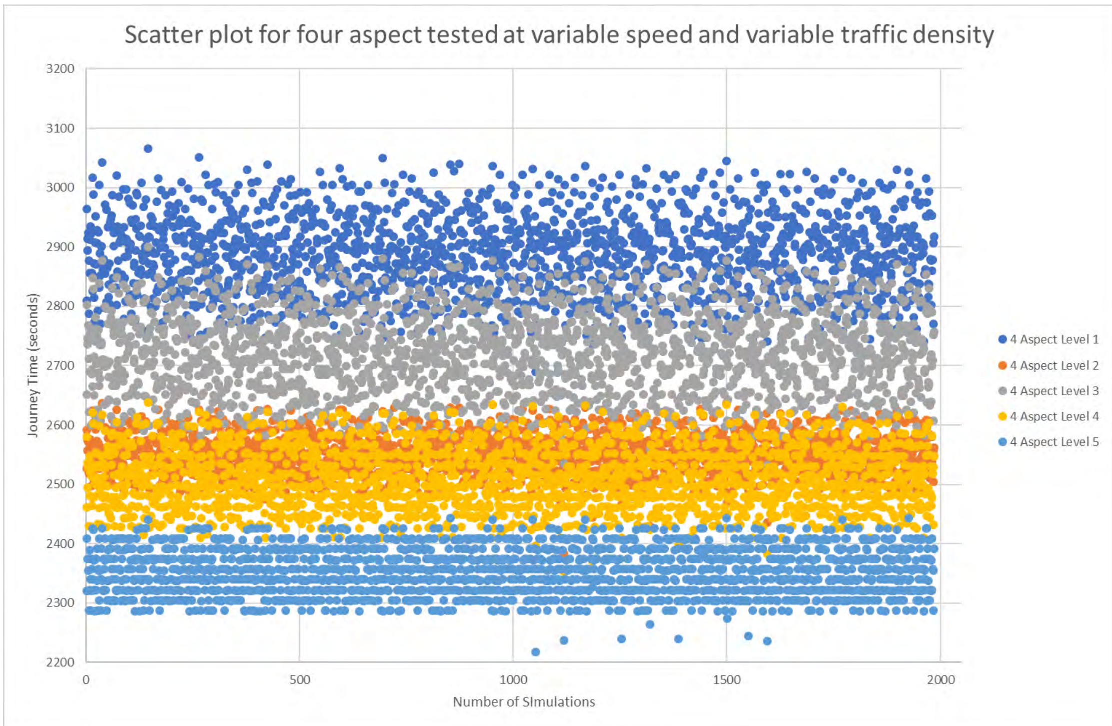
Figure 43: Simulation results for framework testing Level 4 at variable speed & uniform density (author, 2018)

The overall spectrum of the journey times for this test case is between 2300 seconds and 3000 seconds with the overall sample mean around 2650 seconds. As expected, the level with high level of automation (level 5) is at the bottom of the spectrum (shortest journey times) and level 1 with no automation for driving or platform supervision is closer to the upper limit of journey times of 3000 seconds. The intermediate levels show movement with change in each framework testing level (called as FT level from now) as was previously observed under section 6.2.3 above.

The most interesting observation is that for this test case FT level 3 shows distinct banding. The test case applies a variable speed which allows for trains behind catching up to trains ahead always maintaining a set block distance. Hence there is an observable reduction in journey times for the level of automation 5.

Another point of note is that FT level 4 should have seen the same journey time reductions as in FT level 3 but this is not the case. A possible explanation could be that between FT level 3 and 4 the only change is that of static moving block and dynamic moving block. As a result of this the system can be showing sensitivity to the speed of the trains ahead. This observation can be checked against the observations from Test case 3 mentioned in section 6.2.5 below.

6.2.5 Test Case 3 - Variable Speed and Variable Traffic Density



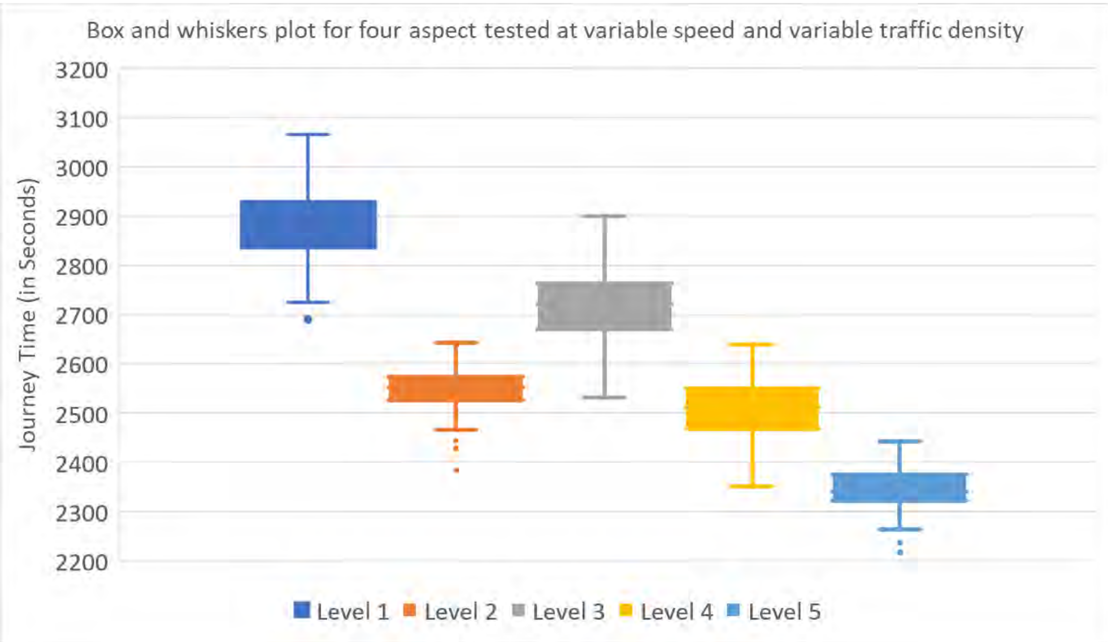
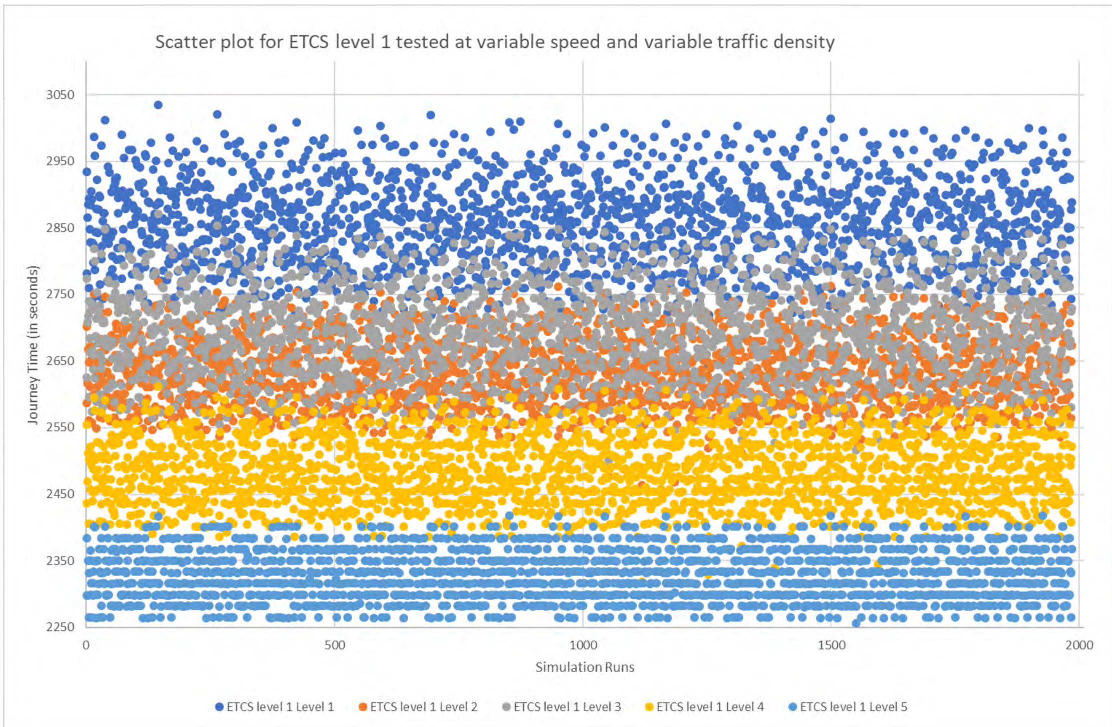


Figure 44: Simulation results for framework testing Level 1 at variable speed & traffic density (author, 2018)



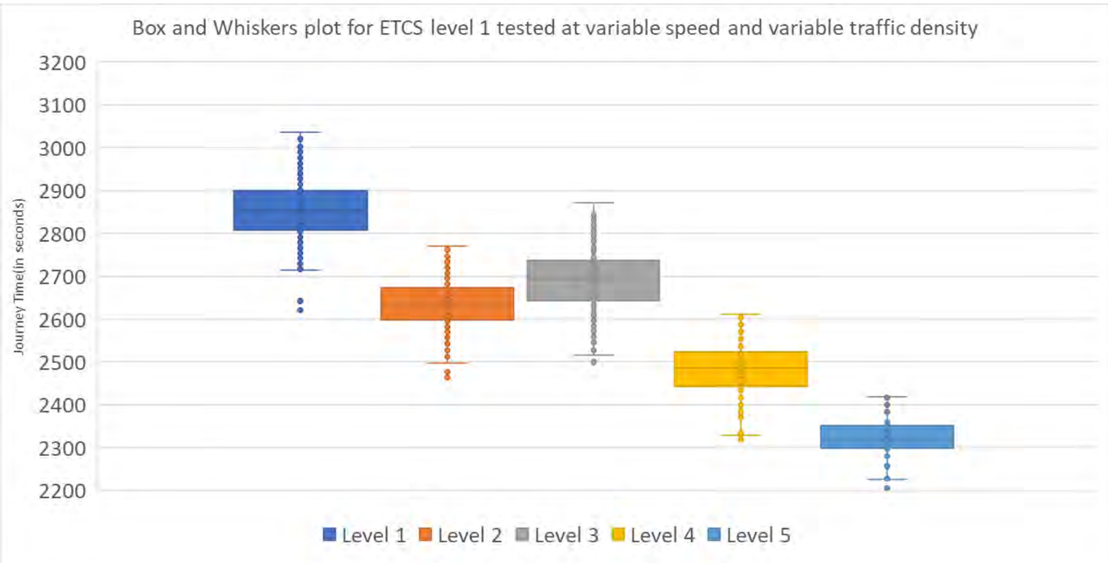


Figure 45: Simulation results for framework testing Level 2 at variable speed and traffic density (author, 2018)

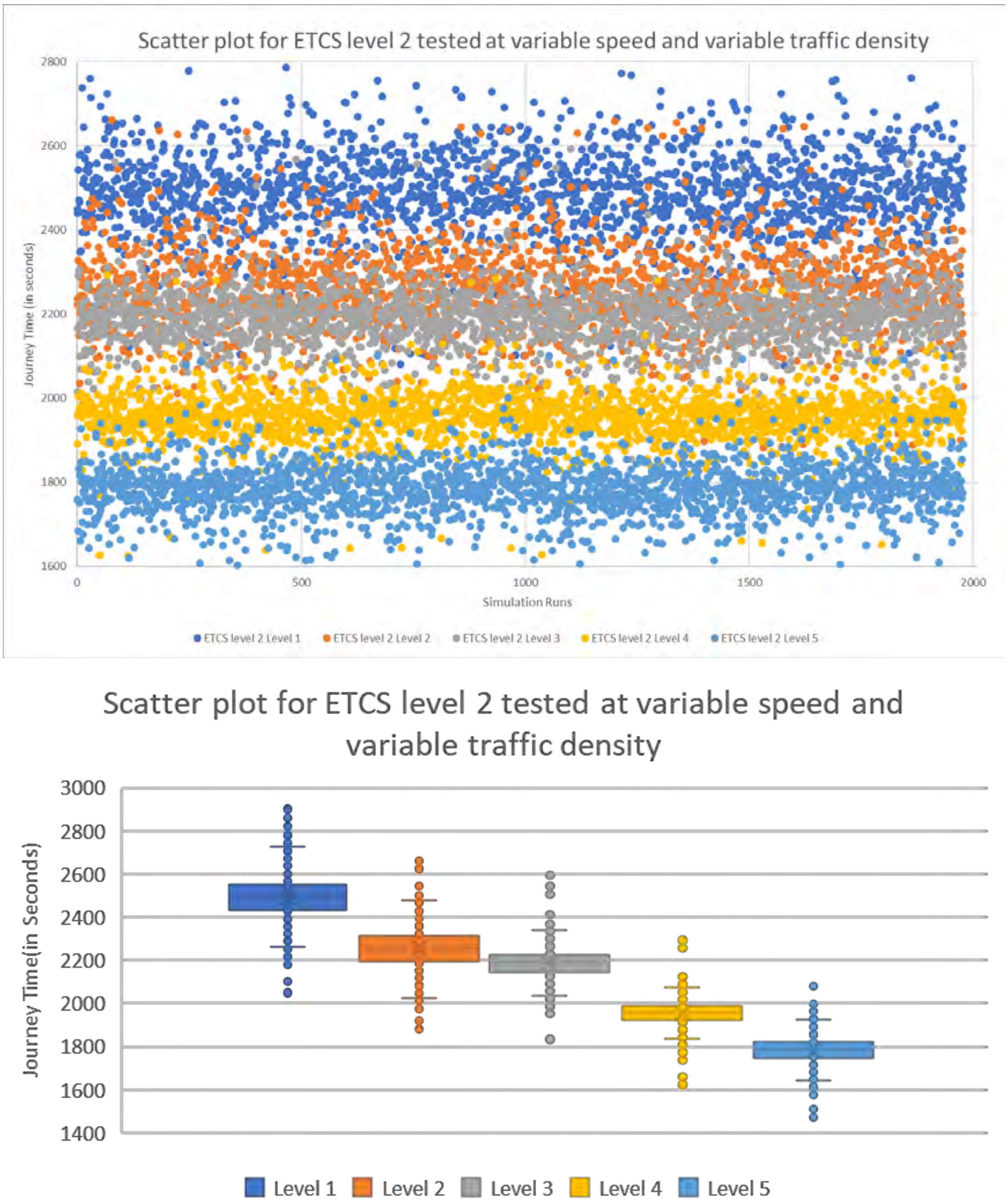
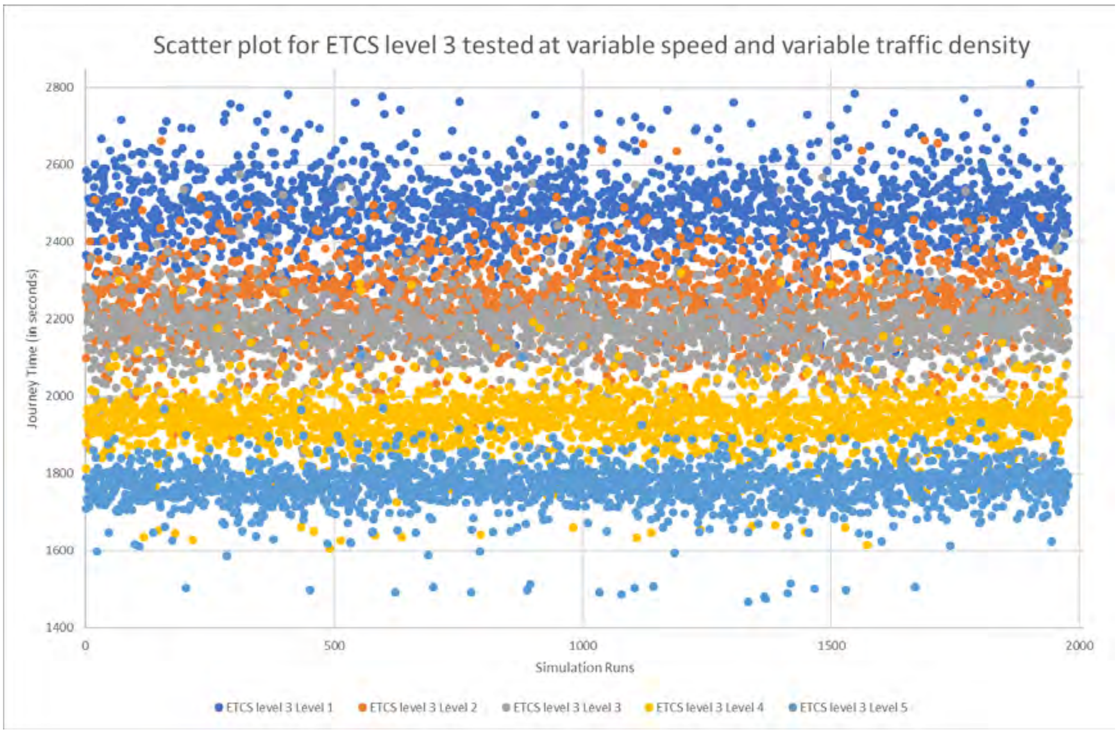


Figure 46: Simulation results for framework testing Level 3 at variable speed and traffic density (author, 2018)



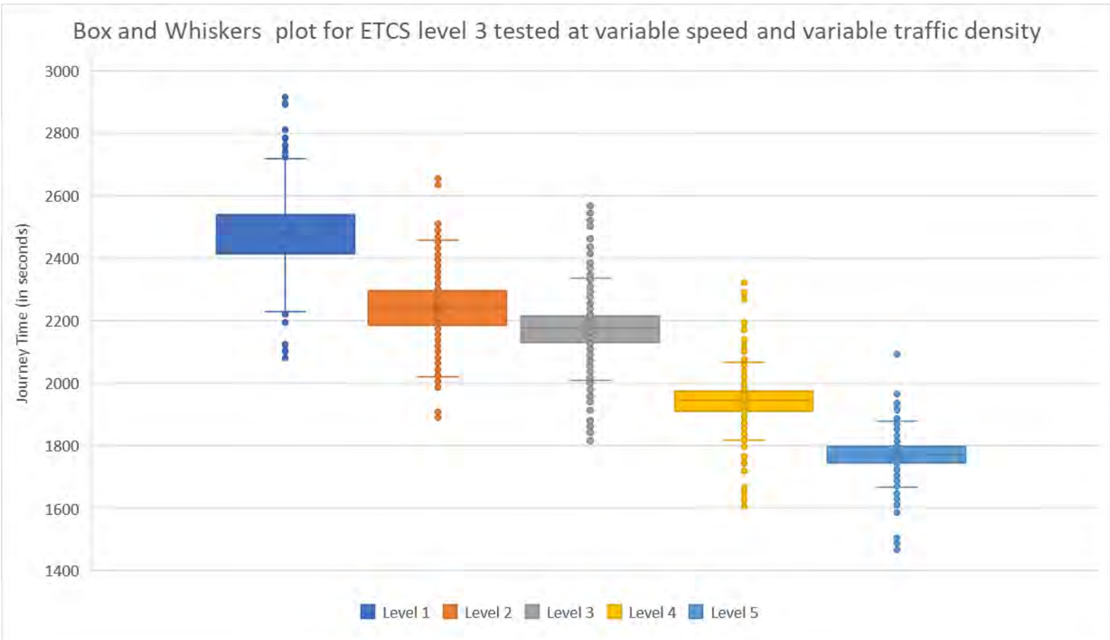


Figure 47: Simulation results for framework testing Level 4 at variable speed and traffic density (author, 2018)

The overall spectrum of the journey times for this test case is between 2300 seconds and 3000 seconds with the overall sample mean around 2650 seconds for FT levels 1 and 2. The journey times for the 2000 points is from 1600 seconds to 2900 for FT level 3 and 1400 to 2600 for FT level 3. There are no major deviations for each FT level other than the ones made under section 6.2.3 above.

As proposed in section 6.2.4, there is an overall reduction in journey times with speed also becoming variable. ETCS Level 3 can make use of this change in order to reduce the inconsistencies in speed. This results in the overall reduction in journey time.

6.3 Overall Inference from Experiments

For all the above-mentioned test cases, we observe that instead of a smooth reduction of journey time from level to level, the second level is almost always lower than the third level when a fixed block system is applied (4 aspect and ETCS 1).

Recall that level 2 specifically is a case where the rolling stock doors are operated manually but the platform is supervised by staff. Level 3 on the other hand is a train with automatic opening and closing of doors without any supervision on the platform. The significance is that in a fixed block system the sensitivity of the overall journey time to the dwell time is higher than in the case of moving block systems. This is because there is no flexibility involved when a train is occupying a platform, dispatch time becomes important and any anomalies to timely dispatch would mount a stochastic delay on the trains behind. Specifically, a supervised platform ensures that trains depart without any need for the driver/train manager to perform multiple duties of ensuring the platform is clear and/or also manage a train.

In a moving block system (ETCS 2 and ETCS 3), trains can be stacked flexibly behind each other and if there is any time being lost on the platform this time can be recovered in the section. Also when the speed is allowed to vary in test case 2, fixed block systems perform much better. This is directly because of the section time becoming more important for improving journey times over dwell time.

By observing the results of where same speed and same traffic density are considered for all automation levels, we can see that the signalling type seems to be a minor contributor to the decrease of journey times (and hence to the increase of capacity). A more noticeable role is played by the automation of driving and platform management.

However, it is possible to envisage that different types of signalling are coupled with different technologies which bring the increase of the allowed speed. Hence, the figures below (Figure 48,Figure 49 and Figure 50) show how journey time changes if the speed allowed increases from 125 to 180, 220 and 300 km per hour when passing from 4 Aspect to ETCS1, ETCS2 and ETCS3. Here, the complimentary technology groups show a clear decrease of journey times, especially when passing to ETCS2 or ETCS3. Finally, allows the verification of the fact that such complimentary technology groups allow maintaining a somehow constant journey time even in case of increase of traffic density, from 21 to 24, 30 and 31 trains per hour when passing from 4 Aspect to ETCS1, ETCS2 and ETCS3.

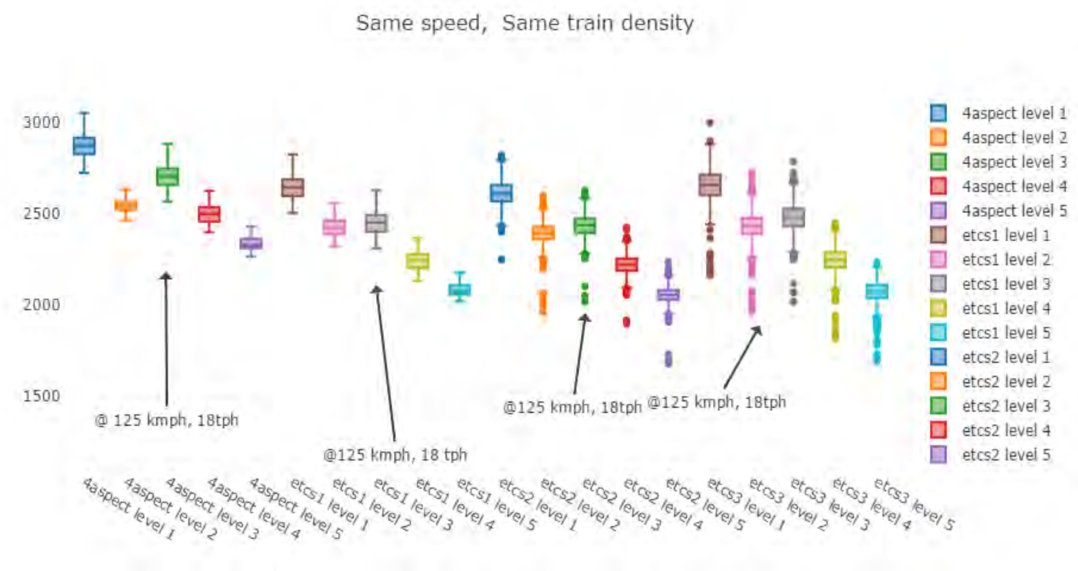


Figure 48: Simulations results Test Case 1

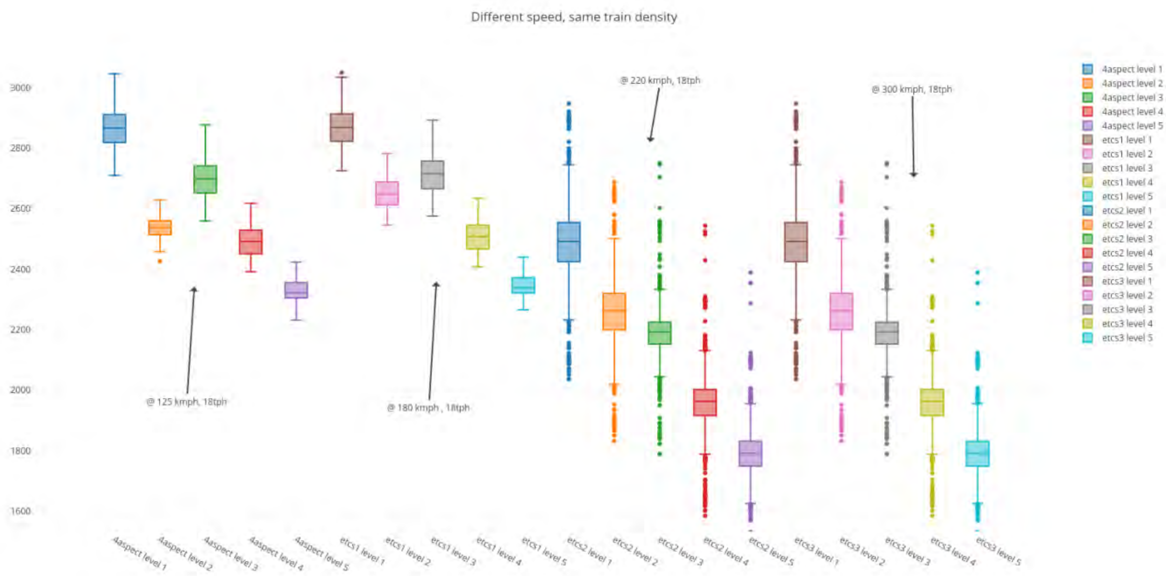


Figure 49: Simulations results Test Case 2

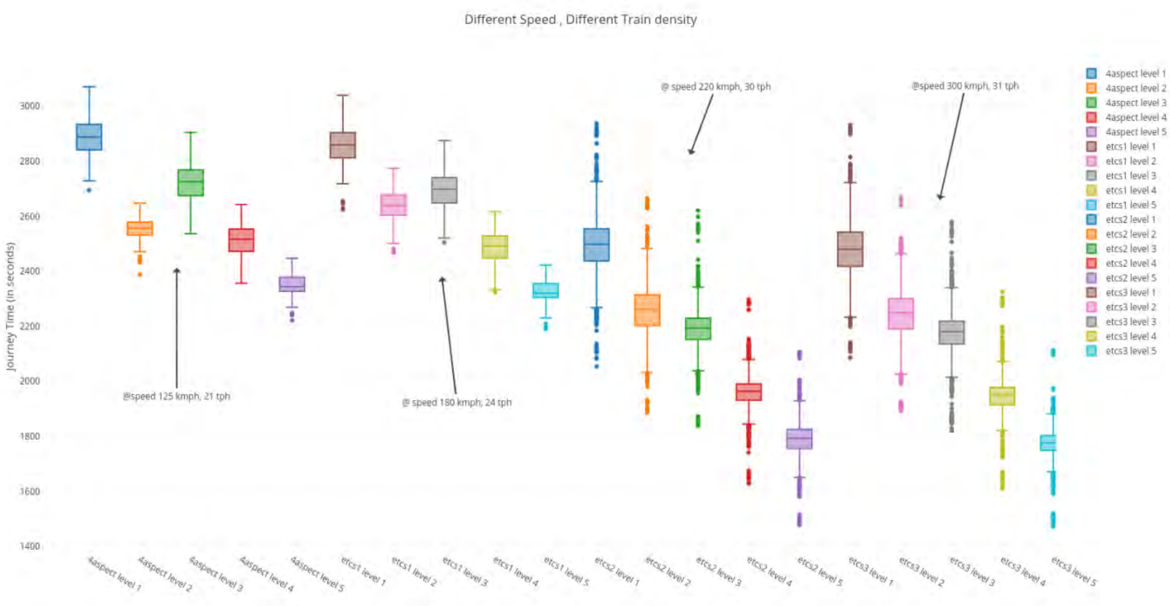


Figure 50: Simulations results Test Case 3

6.4 Summary

This chapter presented the design of experiments for testing the framework discussed in chapter 5. Four levels of automation were chosen and using the framework in the previous chapter a levels of automation roadmap was specified. The experiment was broken down into Test cases, framework testing levels and base level simulation factors.

The results from the experiment were discussed in the next section. The results were shown as a combination of scatter plot, to show the separation of the layers of automation levels, and a box and whiskers plot to show the spread the results of the simulation.

The next chapter presents a case study based on the framework developed in the previous chapters. The line between Kings Cross and Welwyn Garden City is discussed and a framework for its upgrade to higher technology levels are presented.

7 Case Study: Line between Kings Cross and Welwyn Garden City Station

7.1 Introduction

The case study has the objective of mapping the findings from the theoretical work presented in the previous chapter to that of a real life railway. For the case study, the section of railway from Kings Cross to Welwyn Garden City is selected. Using the framework for automation previously introduced in chapter 5, a similar yet simpler automation level increase is proposed. An experiment to test the automation increase is laid out for conducting Monte Carlo based simulation. Finally, the chapter concludes with a discussion of the results.

7.2 Current Scenario

The line from Kings Cross (KXW) to Welwyn Garden City (WGC) on the East Coast Main Line has been selected for this case study. This is because it is one of the busiest stretches of railway in GB largely due to the extensive number of commuter service into London. East Coast Main Line currently provides a capacity of 3400 passenger trains every week, carrying close to 200 million passenger journeys every year (Network Rail, 2018a)

The current public performance measure for the section is 87.2 %. This measure indicates that only 87 % of trains on the line arrive within their scheduled arrival time. The route reports from Network Rail (2018a) highlight the need for renewed investment on the line, as the last major signalling renewal was in the 1970s, track renewal was conducted in 1980s and electrification of the line was completed in 1991 before privatisation.

Over the last few years the demand for services on the line has grown by 30 % between 2011/12 and 2017/18. Consequently, major line upgrades are required in order to meet this demand. The network rail report suggests that the PPM will drop to 86% if no intervention is done on the line.

The report highlighted several key challenges for improving performance on the line. The most significant of the ones being the bottleneck area near Kings Cross station which includes the station throat and the large number of switches and crossings in the area.

Currently the KXW line is serviced by several train operating companies including virgin trains East Coast, northern, great northern and a relatively recently the Thameslink railway. The service mixture on the line comprises of both high speed intercity and suburban trains with freight trains being included at off-peak times.

7.2.1 Infrastructure

The Figure 51 below shows the geographical position of the line around London. A much more detailed line layout is provided in *appendix C* (Network Rail, 2006). It’s a complex track corridor from Kings Cross with a variety of crossing traffic arriving and departing onto the line from Moorgate, Camden junction and Hertford junction.

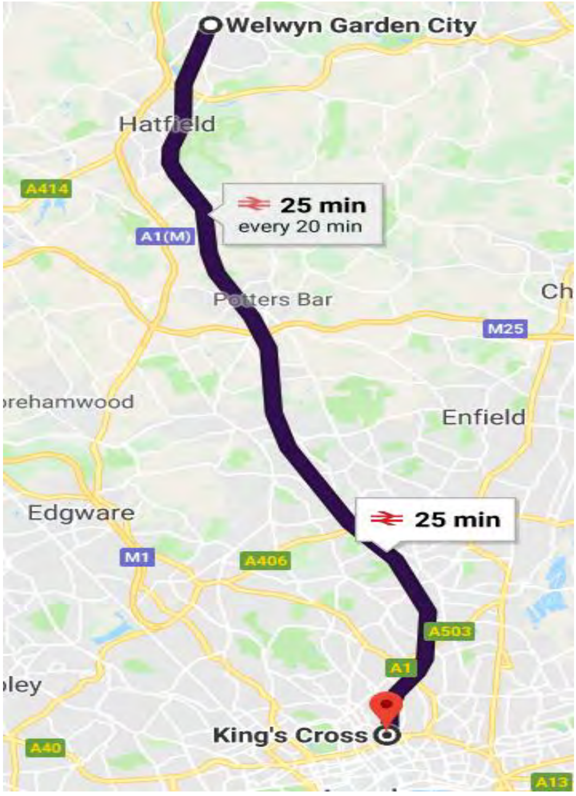


Figure 51: Geographical map of Kings Cross to Welwyn Garden City line (google maps)

The infrastructure is a four-track corridor containing up and down fast lines between Kings Cross and Welwyn station. Thereafter the line converts into a two-track corridor. The area just outside Kings Cross contains complex junctions that allow trains to enter from Camden junction and Moorgate stations and exit onto grade separated infrastructure towards Hertford.

7.2.2 Rolling Stock

The route between Kings Cross and Welwyn the rolling stock is currently operated by Great northern and Thameslink with a combination of older generation of EMUs (class 313 and class 365) and newer EMUs (class 387/1 and 700) (Network Rail, 2006).

Table 17: Top Speeds of the rolling stock currently available on the line (Pritchard, 2017)

Class of train	Maximum speed (km/h)
313	120
365	161
387/1	177
700	161

7.2.3 Signalling

Currently signalled for four aspect with TPWS and AWS systems installed as part of the upgrades in 1970s. The line has seen a rollout of the GSM-R communication system for enabling driver-signaller voice-based communication which currently places it in the ETCS Level 1 equivalent. The existing signalling on the route between Kings Cross and Peterborough is being refitted with ETCS Level 2 with signals which is eventually expected to be transitioned over to commissioned ETCS Level 2 system without lineside signals post 2020. (Department for Transport, 2011)

Thameslink line has been fitted with ERTMS equipment for providing interoperability with the main line network. Thameslink has an ATP overlay towards the centre of the network. All trains currently operational and future orders are expected to have ERTMS driver machine interface (DMI) and GSM-R DMI screens fitted on them. The Hertford loop was experimentally converted to ERTMS with GSM-R for experimental testing for before the Thameslink upgrade.(Kessell, 2015, Kessell, 2017)

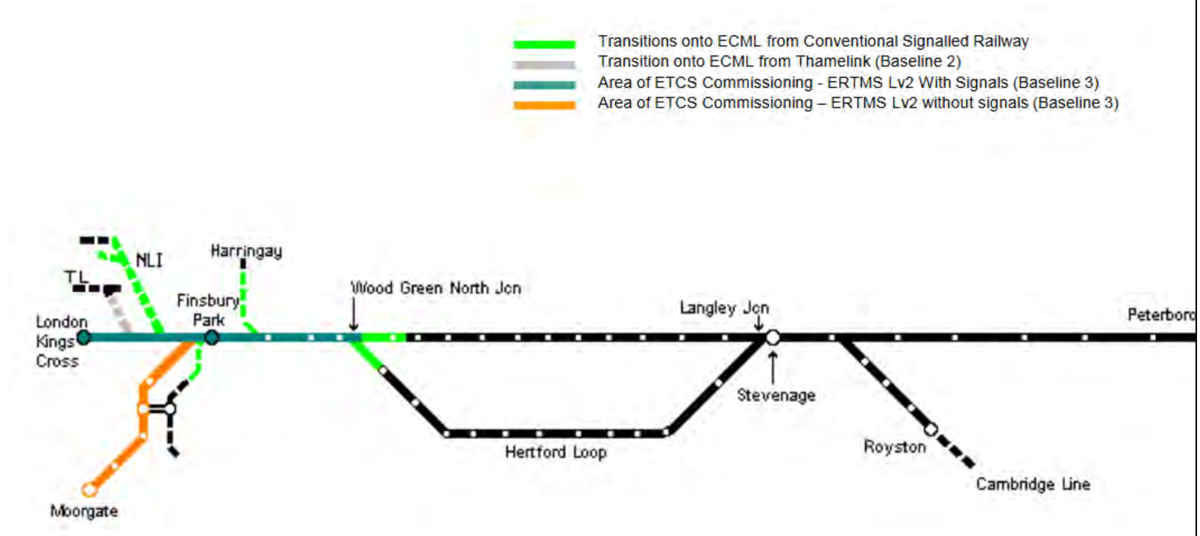


Figure 52: The Kings Cross to Wood green north showing ETCS Level 2 by commissioning plan by December 2018 (MacLennan, 2013)

7.2.4 Timetable

The timetable is currently operated by two train operating companies on the KXW line. Services originating at Kings Cross to Welwyn Garden City is available every 15 mins during the peak hours and every 30 mins during the off-peak hours. This service is provided currently by Thameslink railway (Network Rail, 2018b).

Services originating at Moorgate are timetabled for trains departing towards Welwyn or Hertford branch line. These services join the line after Finsbury Park arrive every 5 mins during the peak hours with alternating trains towards Welwyn or Hertford. During the off-peak this service drops to once every 10 and 20 mins respectively. This service is provided by Great Northern railway (Network Rail, 2018c).

7.3 Experimental Setup

7.3.1 Automation Framework

The chief testing for the experiment is investigating the transition from the current ETCS Level 1 to ETCS Level 3 without line side signalling. The secondary automation focus would be on driving and dwell times at platforms.

For the purpose of this case study, the assumption is made for pure levels of signalling increases. Although it is possible to test overlays of different signalling system for the purposes of this case study, the results would be more conclusive if only pure levels are tested. This is in keeping with the end goal design for the next control period to introduce complete ETCS Level 2 on the line.

Therefore, the signalling levels for testing for performance increases with automation are

1. ETCS Level 1 with line side signals,
2. ETCS Level 2 without line side signals,
3. ETCS Level 3 with moving block signalling.

The levels for evaluating increases in performance train driving automation are

1. Manual driving by a train driver,
2. Automated driving through an ATC (ATP +ATO) system.

The levels for platform supervision to test performance increases with automation are

1. Manual supervision by the driver or station staff,
2. Supervision by station staff only,
3. Supervision by an automated system.

The above can automation levels can be factored into an automation framework as shown in Table 18 below.

Table 18: Automation framework for testing with the KXW line

Automation Level	Signalling	Driving	Platform
0	ETCS Level 1	Manual	Driver
1	ETCS Level 2	Manual	Staff
2	ETCS Level 3	Computer	Computer

For the purposes of this case study the existing timetable on the line is used for analysis. The time period selected is the time between 9:00 am and 12:00 am where the traffic contains 70 services. The national railway timetable made available by network rail is used as a reference for creating the timetable (Network Rail, 2018b, Network Rail, 2018c) for the simulation.

The timetable consists of 60 services, reduced from the 70 to increase clarity of results in the simulation phase.

1. Eight suburban services up and down from Camden road branch travelling to Welwyn Garden City,
2. Eight suburban services from Moorgate travelling in the up direction to Welwyn,
3. Eight suburban services from Welwyn travelling in the down direction to Moorgate,
4. Eight suburban services from Moorgate travelling in the up direction to Hertford,
5. Eight suburban services from Hertford branch travelling in the down direction to Moorgate,

6. Six intercity services travelling from Kings Cross towards Welwyn in the up direction. These services do not call at any station in the route,
7. Six intercity services travelling from Kings Cross towards Welwyn in the up direction. These services do not call at any station in the route,
8. Four interurban services calling at stops between Welwyn and Kings Cross in the down direction,
9. Four interurban services calling at stops between Kings Cross and Welwyn in the up direction.

Also, there are no speed restrictions applied to the timetable. Each service uses rolling stock capable of running up to the designated line speed. The up and down fast are designated at 180 km/h whereas the slower lines on the outside are designated with a speed of 120 km/h. Stopper services use the routes available on the up and down slow with the faster through services using the up and down fast.

The factors that are controlled within the simulation are shown in the table below. They are constructed using the distributions selected from the modelling and regression investigations as conducted by Higgins, Kozan and Yuan et. al. (Higgins and Kozan, 1998) (Yuan, 2008b).

Table 19: Table showing the factors that impact the subsystem during the simulation and their respective distributions (Yuan, Yuan and Hansen)

Delay Factors	Distribution
Interlock time	Exponential
Door open & close times	Normal
Dispatch time	Uniform
Route setting times	Normal

7.3.2 Experiment Setup and Simulation

In order to test the framework proposed in the previous section an experiment was designed to check all possible combinations. The signalling levels proposed in the previous section are designated as S1, S2 and S3 respectively. The driving levels proposed in the previous section are designated as D1 and D2 respectively. Finally, the platform levels as per the previous levels are designated as P1, P2 and P3 respectively.

The experiment set is shown in the table below. Three combinations were rejected because it is improbable for manually (train staff) operated doors being present with completely automated driving. Also, the experimental case consisting of ETCS Level 3 with manual driving and automated platform supervision can also be removed.

Table 20: Experiment set for the case study. The shaded cases are not considered for the simulation

Results
Case Study: Line between Kings Cross and Welwyn Garden City Station

Experiment	Signalling	Driving	Platform	Description
E1	S1	D1	P1	
E2	S1	D1	P2	
E3	S1	D1	P3	
E4	S1	D2	P1	Manual doors with automated driving
E5	S1	D2	P2	
E6	S1	D2	P3	
E7	S2	D1	P1	
E8	S2	D1	P2	
E9	S2	D1	P3	
E10	S2	D2	P1	Manual doors with automated driving
E11	S2	D2	P2	
E12	S2	D2	P3	
E13	S3	D1	P1	
E14	S3	D1	P2	
E15	S3	D1	P3	ETCS Level 3 with manual driving
E16	S3	D2	P1	Manual doors with automated driving
E17	S3	D2	P2	
E18	S3	D2	P3	

This experiment set contains 14 cases that require checking over the 3 signalling cases. Each case returns the journey time (in the form of running time and dwell time) for each train in the timetable. Each case is individually iterated for 50 runs. This is to show convergence of the data points around a mean for each individual case. A net total of 700 simulation runs were conducted resulting in a total of 42,000 data points for all trains in the network.

The simulation was carried out using BRaVE as the microsimulation environment and Matlab as the macro simulation environment. The factors and the input variables were inserted from Matlab using the same architecture as that which was used in the previous chapter. The main difference from the theoretical simulation conducted in chapter 5&6 to the case study presented here is the absence of control layers that changed the density of traffic and speed of traffic in the microsimulation. No artificial delays were injected to test the robustness of the system.

7.4 Results

The results from the simulation are shown in the Figure 55. A simple scatter plot is used to represent the average journey times for each service in the timetable. The journey time is plotted on the y axis against each experiment case plotted on the x axis. The legend on the graph indicates the individual service type for the timetable.

Individual service types can be categorised into suburban and intercity services. Services numbered between 700 and beyond are intercity services. Figure 54, shows the results of the intercity services across the network. The services with times less than 600 seconds on the network are crossing the network onto a branch line. Services numbered between 100 and 700 are suburban services. Figure 53 shows the results of the suburban services across the network.

In the graph the automation levels increase from the left to the right. The results from the simulation support the theoretical conclusions in that there are improvements to journey time. However, the improvements are not as significant as the results seen in the theoretical simulation and analysis conducted in the previous chapters.

This lack of clarity in the outcome is because of the high traffic density and the lack of homogeneity between the trains' speed during the simulation. However, it supports, albeit indirectly, the conclusion that automation improvements need to be made on a whole system basis with changes also being made to supporting subsystems.

Figure 53: Simulation results for suburban trains only. Automation levels increase left to right (author, 2018)

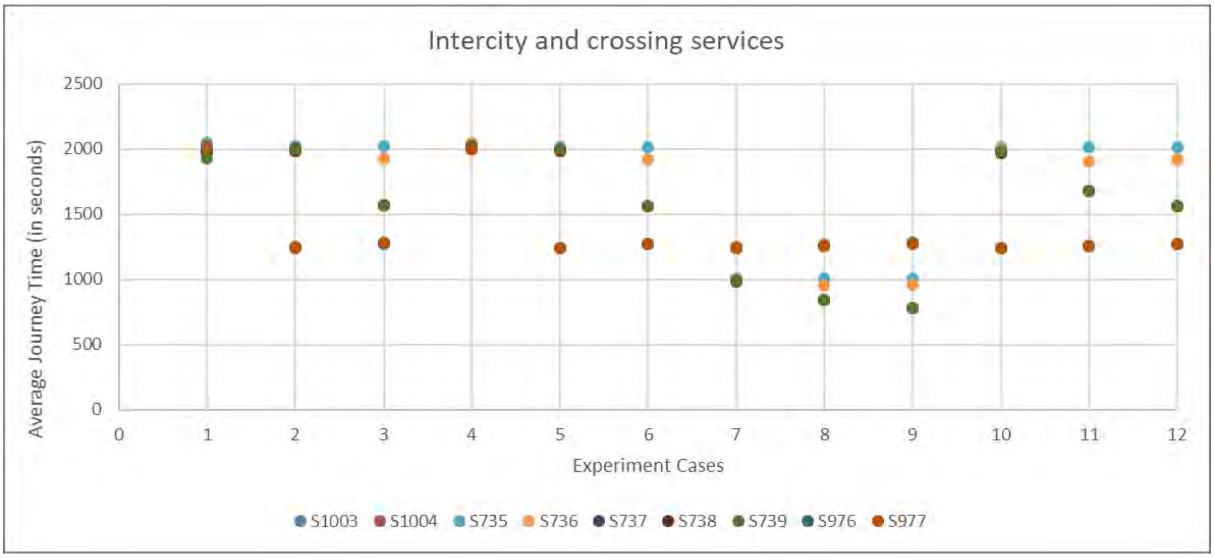


Figure 54: Simulation results for suburban trains only. Automation levels increase left to right (author, 2018)

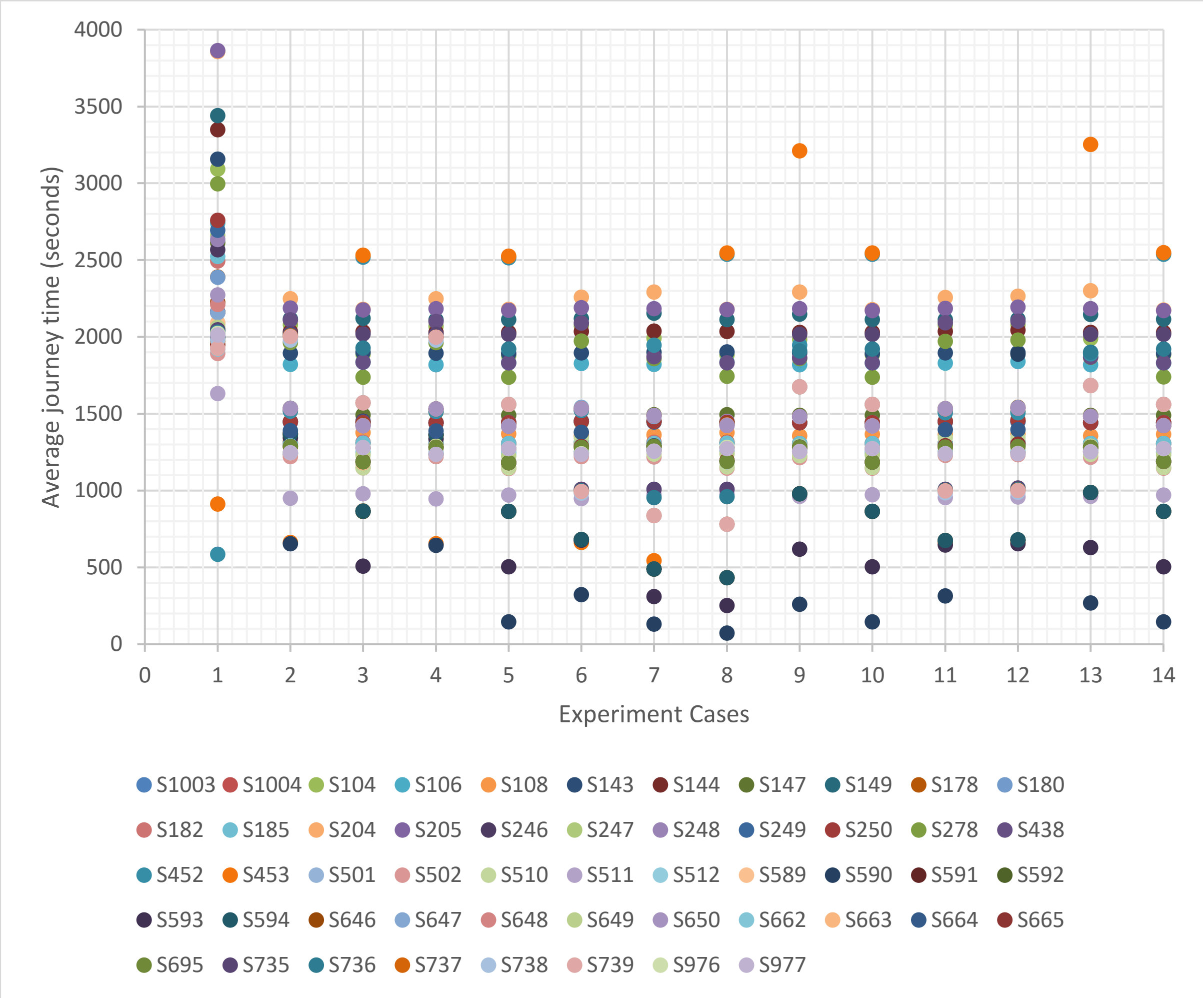


Figure 55: Overall Results from the simulation of the automation framework (author, 2018)

7.5 Conclusion

In this case study, a real-world railway line was chosen. The line chosen was the section of the East Coast Main Line around the bottle-neck area of Kings Cross station. I then propose a suitable automation framework for testing performance increases. The automation framework consisted of conducting signalling upgrades, the introduction of driverless trains and changing the supervision level of the platform from manual to automatic. The factors for the simulation were specified. Finally, the simulation of a real-world railway was carried out.

The simulation did not yield very clear results but supported the trend of increasing levels of automation improving journey time performance. Improvements that could have been seen if with the increase in automation the timetable was changed as well. Also, another factor could have improvements could be clearer if the permissible speed of services was increased which would improve the real journey time improvements could have been made.

The simulation also indicates the requirement for analysing mixed modes of operation with islands of different signalling system. This was unfortunately, beyond the scope of the objectives of this thesis.

Conclusion

8 Conclusion

The overall objective of this thesis and the motivation are presented in chapter one. The objective of the thesis is to analyse the impact of automation on railway system performance using simulation. The null hypothesis for the thesis is that automation has no impact on system performance when applied to a railway system. The alternative hypothesis is that system performance will see improvement when automation is applied. Over the course of this thesis, it is shown that the null hypothesis can be rejected, but the conclusion does not entirely support the alternative hypothesis either. An overview of the chapter layout and publication outcomes was presented in this chapter as well.

The second chapter of this thesis began with an overview of basic thoughts and progress in the concept of automation. I examined the different technological progressions in automation from the initial paper written on automation by (Rasmussen). This involved the initial organisation of an automated system in the form of a Master and slave system where the human places a demand on the system. The next generation refined the same into a supervisory system where the machine was allowed to analyse tasks, but the human was expected to approve the machine selected actions. Parasuraman et al. (McGee et al., 1998, Parasuraman et al., 2012, Parasuraman and Riley, 1997, Parasuraman et al., 2000, Wickens et al., 2010)propose a systematic approach to understand and describe the system using two major axes. The primary axis described the type of activity being performed (detect, analyse, select, and action) and the secondary axis described the level of automation involved in executing the activity (from low to high).

The third chapter provides an overview of the developments in the railway system. The chapter introduces railways as a system with several subcomponents. The subcomponents being infrastructure, rolling stock, signalling, and train control components. Through the literature the I have tried to present an evolution of philosophies used to model the railway. From the early time block signalling to the present day distance block signalling and flowing into the future with speed block signalling. The chapter also presents a hierarchy of delays and how each of the delay patterns have been identified with specific distribution patterns.

The fourth chapter presents an overview of simulation methods both hardware and software. Literature is presented of various simulation architectures that have been followed elsewhere. This chapter also contains a discussion on Monte Carlo based simulation and the development of a numerical expression to develop an expression for the framework in the proceeding chapters.

The fifth chapter provides an initial framework development and presents an ideal roadmap using the various signalling and infrastructure-based technologies described in chapter two. Using the levels of automation technique presented in chapter one, the author has aimed to discuss the roadmap from the perspective of increasing levels of automation. Further to this, a numerical expression is developed for the framework so that a Monte Carlo based simulation can be proposed to test the changes in performance with increasing levels of automation.

The sixth chapter presents an experiment definition for testing the framework developed in the previous chapter. The overall experiment was divided into three tiers. The first tier contains three test cases where the framework was tested at three different speeds with uniform or variable traffic demands. The second tier contains four framework testing levels where the key variables of speed and traffic density are split into four testing levels. The final base tier contains five base cases where variables for driving, platform supervision and junction set reset times are specified. The chapter ends with a discussion on the results of the simulation. A summary of the outcomes from the simulation is presented later in this chapter.

The seventh chapter presents a case study on the line between Kings Cross and Welwyn Garden City line on the East Coast Main Line. The study presents a simplified experimental setup due to the complexity of junctions present on the mainline of the network. The experiment was limited to two tiers of simulation, the first one manipulating the base tier variables such as door operation and driving. The upper tier changes the overall signalling parameters only. The outcome from the case study shadows the theoretical experiments conducted in the previous chapters, albeit with a lot more noise.

8.1 Summary of Outcomes

The thesis proposes a roadmap for automation increase on the railway system. Each subsystem is characterised by an individual roadmap with a view of arranging the stages to have a step by step increase in automation between each level.

A framework for evaluating increasing levels of automation proposed previously was developed. A Monte Carlo method was implemented for use within the thesis. A formal numerical expression for the delays accrued within a block section of the network was introduced.

A simulation model was created with a theoretical network in BRAVE. The model was developed to reflect different signalling levels, traffic densities, speed changes. Also, the elements for driving, dispatching, door operation and platform supervision were provided as controls within the system.

The results from the simulation show that with changing levels of automation, the performance of the system does vary. However, the change need not always be positive for the system. This is shown from comparing the scenario of an automated train with platform supervision being conducted by a human versus the scenario of an automated train with no platform supervision. The simulation shows that the performance of the system is relatively better for the former case rather than the latter. This observation holds true if the system is in a fixed block system.

As the railway system moves to a moving block configuration, at a lower level of automation, the benefits are seen when the trains are moving in the block. Previously automated trains in the fixed block system were spending more time waiting in the blocks between trains. However, in a moving block system at a higher speed, the trains cope with an unattended platform at a lower level of automation.

The work presented in this thesis was included as part of chapter 3, sections 3.2 of the Capacity for rail project. The chapter highlighted the use of automation roadmaps and a testing framework for analysing the impact on the railway.

The numerical expression developed was published as a conference paper for the IEEE intelligent transportation systems 2015 conference titled “Impact of Automation on the Capacity of a Mainline Railway: A Preliminary Hypothesis and Methodology”. As the title suggests, the paper was a preliminary hypothesis, the content of this thesis is the actual experiment to verify the stated hypothesis in that paper. (Venkateswaran et al., 2015)

8.2 Significant Conclusions

Automation is beneficial at a group level. This means that when the automation level of an individual subsystem is increased it is necessary to make upgrades to the interfaces of that system. For example, if the train is driven by a computer then the platform supervision and the route setting will also need to be automated. This ensures that the system is free of delays accrued due to human inaccuracy.

Suggestions to make door operation supervised by the driver will increase the risk of delays within the system. This is because the drivers' workload will have increased to include monitoring platforms. Human reaction delays will play a major role in the dispatch of a train from the station into the railway sections ahead. The following figures, Figure 56, Figure 57 and Figure 58, summarise the results of the simulations and support the conclusions stated here.

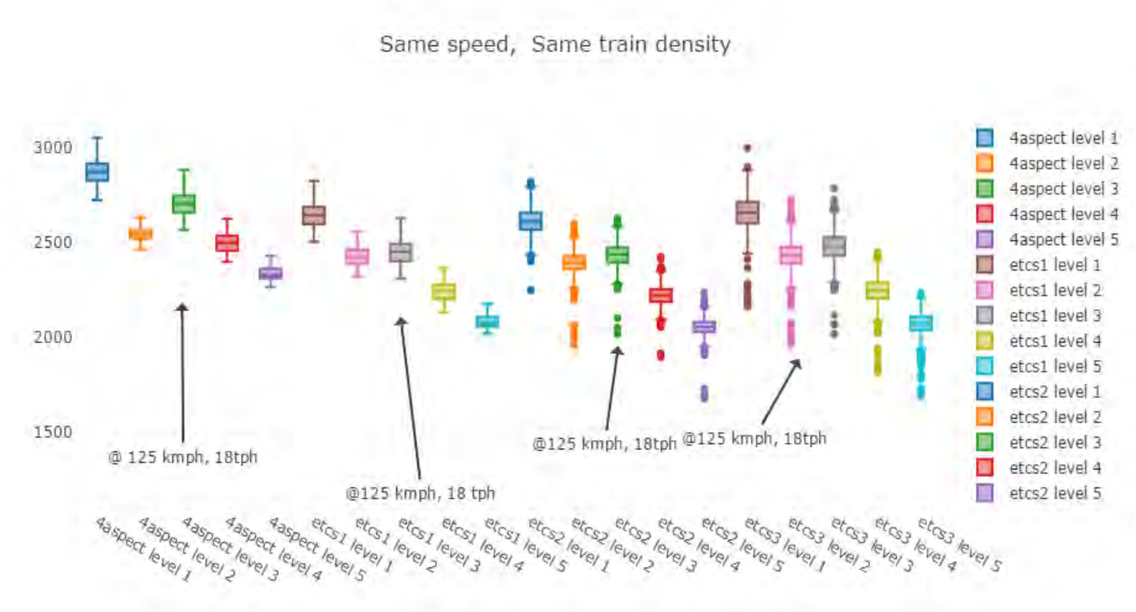


Figure 56: Summary of outcomes from simulation when system is operated at a uniform speed and traffic density. Automation increases from left to right. (author, 2018)

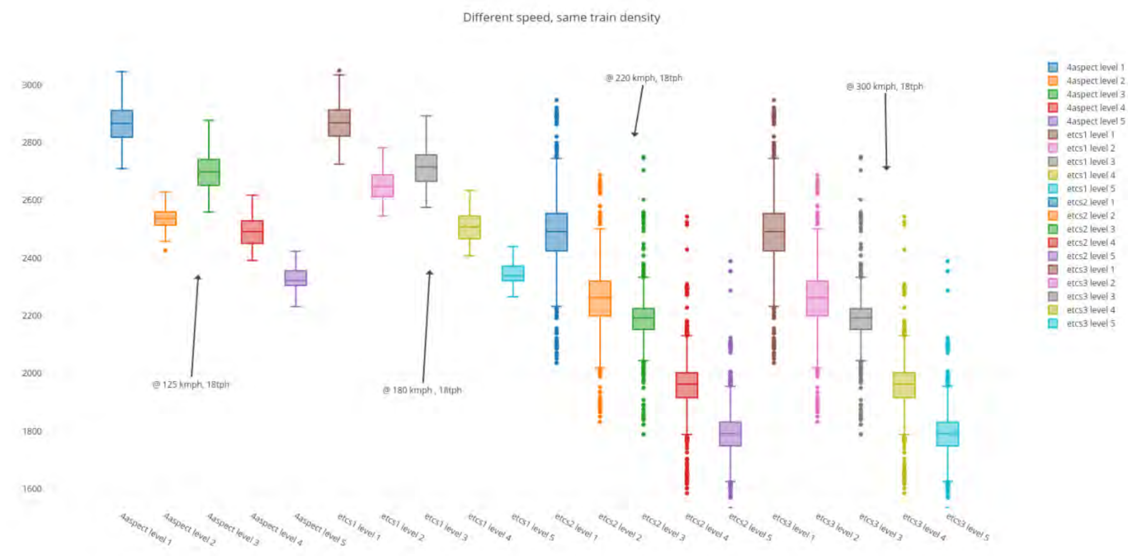


Figure 57: Summary of outcomes from simulation when system is operated at a variable speed and uniform traffic density. Automation increases from left to right. (author, 2018)

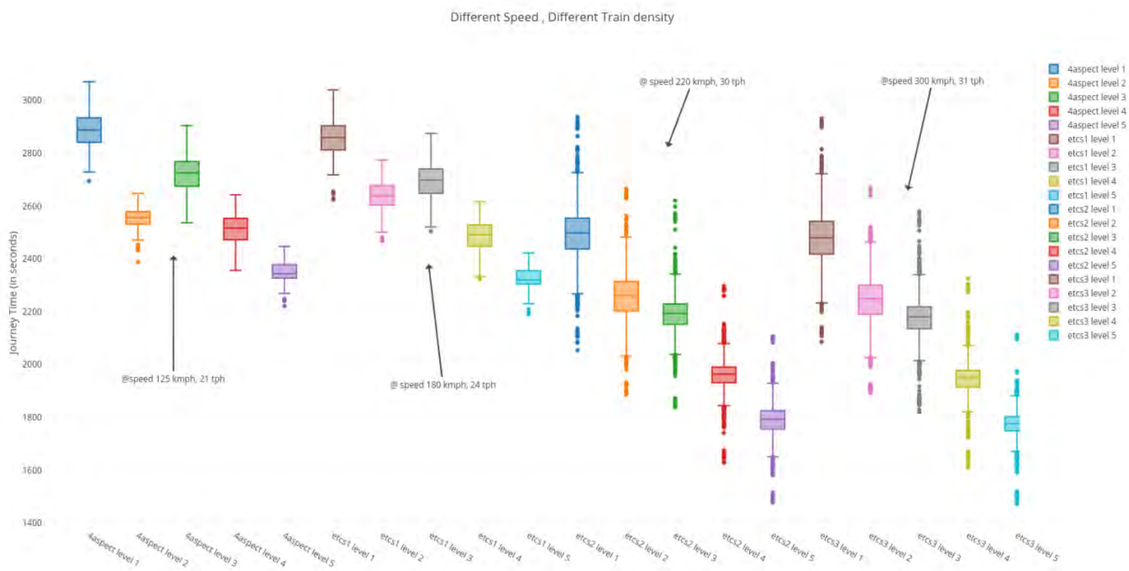


Figure 58 Summary of outcomes from simulation when system is operated at a variable speed and traffic density. Automation increases from left to right. (author, 2018)

8.3 Recommendations for Future Work

The case study conducted does not fully resolve the noise in the system due to the complex interactions between subsystems. To better characterise the system, future work will probably merit from working on a real timetable.

The experiment considers only pure automation levels which might not be the reality on a railway line. The diversity of infrastructure may mean that there are isolated areas of high automation in the middle of a medium level automated system. Future projects could focus on a mixed infrastructure railway where some stretches could be at a high or low level of automation. The latest railway lines in Britain such as Crossrail and Thameslink could serve as excellent baseline models for a mainline railway.

Most of the simulation work conducted in the thesis ignores freight service on the corridor. It should be noted that freight services are the slowest moving objects on the railway and will have a detrimental effect on the performance after moving to a higher level of automation. This is due to the relatively high axle loads on freight combined with the length of the services. A mainline railway case should consider this as part of the equation. This thesis did not include a freight service because the discussion was on improvements in the performance of a passenger service limited within a limited timetable window.

Future work could also focus on finding an optimum point between dwell times factors with increasing levels of automation. During the research period, the experiments showed that designing dwell times is a significant factor for achieving higher performance over the railway line. A microanalysis may be required to show the relationship between the alighting and boarding times to the door width, coach occupancy level, platform occupancy level, etc. Such a study would bring out optimum designs for coaches so that the planned dwell time can be achieved given a loading rate. This is similar to the study proposed in Chapter 4 of this thesis.

9 References and Bibliography

1889. Regulation of Railways Act of 1889. *57.* United Kingdom: legislation.gov.uk.

ABRIL, M., BARBER, F., INGOLOTTI, L., SALIDO, M. A., TORMOS, P. & LOVA, A. 2008. An assessment of railway capacity. *Transportation Research Part E: Logistics and Transportation Review,* 44**,** 774-806.

BAINBRIDGE, L. 1983. Ironies of automation. *Automatica,* 19**,** 775-779.

BSI 2006. Railway applications — Urban guided transport management and command/control systems. *Part 1: System principles and fundamental concepts.* London: British Standards Institure.

BSI 2014. Railway applications — Urban guided transport management and command/control systems. *Part 2: Functional requirements specification.* London: British Standards Institure.

CAREY, M. 1998. Optimizing scheduled times, allowing for behavioural response. *Transportation Research Part B: Methodological,* 32**,** 329-342.

CAREY, M. & KWIECIŃSKI, A. 1994. Stochastic approximation to the effects of headways on knock-on delays of trains. *Transportation Research Part B: Methodological,* 28**,** 251-267.

CHAHAR, K., CHENG, C. & PRANOTO, Y. Strategic crew planning tool in railroad: A discrete event simulation. Simulation Conference (WSC), Proceedings of the 2011 Winter, 11-14 Dec. 2011 2011. 3688-3698.

CONFESSORE, G., LIOTTA, G., CICINI, P., RONDINONE, F. & DE LUCA, P. A simulation-based approach for estimating the commercial capacity of railways. Simulation Conference (WSC), Proceedings of the 2009 Winter, 13-16 Dec. 2009 2009. 2542-2552.

CULLEN, W. D. B. 2000. *The Ladbroke Grove Rail Inquiry. Pt.1, Report,* Sudbury, Sudbury : HSE Books.

DE FABRIS, S., LONGO, G., MEDEOSSI, G. & PESENTI, R. 2014. Automatic generation of railway timetables based on a mesoscopic infrastructure model. *Journal of Rail Transport Planning & Management,* 4**,** 2-13.

DE WINTER, J. & DODOU, D. 2014. Why the Fitts list has persisted throughout the history of function allocation. *Cognition, Technology & Work***,** 1-11.

DENTON, H. R. 1987. The causes and consequences of the Chernobyl nuclear accident and implications for the regulation of U.S. nuclear power plants. *Annals of Nuclear Energy,* 14**,** 295-315.

DEPARTMENT FOR TRANSPORT 2011. ERTMS national implementation plan London: Department for Transport

DOUGLAS, H., WESTON, P., KIRKWOOD, D., HILLMANSEN, S. & ROBERTS, C. 2017. Method for validating the train motion equations used for passenger rail vehicle simulation. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit,* 231**,** 455-469.

DUNTON, C. E., KELL, J. & MORGAN, H. D. Victoria line: Experimentation, Design, Programming, And Early Progress.(includes plates.). ICE Proceedings, 1965. Thomas Telford, 1-24.

FITTS, P. M. 1951. Human engineering for an effective air navigation and traction control system. Columbus, OH: Ohio state University Foundation Report.

GIBSON, S. 2003. Allocation of capacity in the rail industry. *Utilities Policy,* 11**,** 39-42.

GIBSON, S., COOPER, G. & BALL, B. 2002. Developments in Transport Policy: The Evolution of Capacity Charges on the UK Rail Network. *Journal of Transport Economics and Policy (JTEP),* 36**,** 341-354.

GREEN, S. R., HICKSON, D., WARD, D., ROBERTS, C., WESTON, P. & STEWART, E. Monitoring the DC third rail interface using an in-service train. Railway Condition Monitoring and Non-Destructive Testing (RCM 2011), 5th IET Conference on, 29-30 Nov. 2011 2011. 1-5.

GRUBE, P. & CIPRIANO, A. 2010. Comparison of simple and model predictive control strategies for the holding problem in a metro train system. *Intelligent Transport Systems, IET,* 4**,** 161-175.

HAITH, J., JOHNSON, D. & NASH, C. 2013. The case for space: the measurement of capacity utilisation, its relationship with reactionary delay and the calculation of the capacity charge for the British rail network. *Transportation Planning and Technology,* 37**,** 20-37.

HALL, C. 2016. *Modern Signalling Handbook,* London, Ian Allen Publishing.

HANSEN, I. A. 2008. *Railway timetable & traffic: analysis, modelling, simulation*, Eurailpress.

HARRIS, N. G. 2006. Train boarding and alighting rates at high passenger loads. *Journal of Advanced Transportation,* 40**,** 249-263.

HEWISON, L. R. DESIGN AND CONSTRUCTION OF INTERSECTING IN SITU TUNNEL OPENINGS ON THE MURL PROJECT. 1981. 177-186.

HIGGINS, A. & KOZAN, E. 1998. Modeling Train Delays in Urban Networks[1]. *Transportation Science,* 32**,** 346.

HYUN-KYU, J. & SUNG-KYOU, C. Development of a Multi-train Operation Simulator with Interactive Human Computer Interfaces. Hybrid Information Technology, 2006. ICHIT '06. International Conference on, 9-11 Nov. 2006 2006. 237-243.

KAAKAI, F., HAYAT, S. & EL MOUDNI, A. 2007. A hybrid Petri nets-based simulation model for evaluating the design of railway transit stations. *Simulation Modelling Practice and Theory,* 15**,** 935-969.

KESSELL, C. 2015. The transition to ERTMS in the UK. *Rail Engineer* [Online]. Available: https://www.railengineer.uk/2015/03/31/the-transition-to-ertms-in-the-uk/.

KESSELL, C. 2017. UK signalling - A 2017 update. *Rail Engineer* [Online]. Available: https://www.railengineer.uk/2017/02/23/uk-signalling-a-2017-update/.

KRIVKA, J., ELIS, L., VLASEK, J., ZAHOUR, J., STETKA, P. & KOSTURIK, K. Hardware for rail traffic simulator. Telecommunications Forum (TELFOR), 2013 21st, 26-28 Nov. 2013 2013. 590-593.

KRUEGER, H. Parametric modeling in rail capacity planning. Simulation Conference Proceedings, 1999 Winter, 1999 1999. 1194-1200 vol.2.

KWANGSOO, S., YOUNGBUM, L., YONGSOO, S., SUNGHO, H. & JONG-HO, L. Ride Comfort Analysis based on Heart Rate and Blood Pressure Variability according to Changed Tilting

Angle at Tilting Train Simulator. Smart Manufacturing Application, 2008. ICSMA 2008. International Conference on, 9-11 April 2008 2008. 570-574.

LANDEX, A. & NIELSEN, O. A. Simulation of disturbances and modelling of expected train passenger delays. 2006. 521-529.

LINDFELDT, O. 2006. Influences of station length and inter-station distance on delays and delay propagation on single-track lines with regional rail traffic. *Computers in Railways X : Computer System Design and Operation in the Railway and Other Transit Systems.* Southampton: WIT Press.

LO, J. C. & MEIJER, S. A. Situation awareness measurement techniques for gaming simulations: An overview and application for railway traffic controllers. Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2013 IEEE International Multi-Disciplinary Conference on, 25-28 Feb. 2013 2013. 238-245.

MACLENNAN, D. B., GRAEME 2013. ETCS on East Coast Main Line. IRSE.

MALAVASI, G., MOLKOVÁ, T., RICCI, S. & ROTOLI, F. 2014. A synthetic approach to the evaluation of the carrying capacity of complex railway nodes. *Journal of Rail Transport Planning & Management,* 4**,** 28-42.

MALAVASI, G. & RICCI, S. 2001. Simulation of stochastic elements in railway systems using self-learning processes. *European Journal of Operational Research,* 131**,** 262-272.

MATTSSON, L.-G. 2007. Railway Capacity and Train Delay Relationships. *In:* MURRAY, A. & GRUBESIC, T. (eds.) *Critical Infrastructure.* Springer Berlin Heidelberg.

MCGEE, J. P., PARASURAMAN, R., MAVOR, A. S. & WICKENS, C. D. 1998. *The Future of Air Traffic Control:: Human Operators and Automation*, National Academies Press.

MELI, E., MALVEZZI, M., PAPINI, S., PUGI, L., RINCHI, M. & RINDI, A. 2008. A railway vehicle multibody model for real-time applications. *Vehicle System Dynamics,* 46**,** 1083-1105.

MESHKATI, N. 1991. Human factors in large-scale technological systems' accidents: Three Mile Island, Bhopal, Chernobyl. *Organization & Environment,* 5**,** 133-154.

METROS, O. O. A. 2016. Statistics Brief - world report on automation *In:* (UITP), I. A. O. P. T. (ed.). Brussels: UITP.

MUNIPOV, V. M. 1992. Chernobyl operators: criminals or victims? *Applied Ergonomics,* 23**,** 337-342.

NETWORK RAIL 2006. London North Eastern Sectional Appendix. *Sectional Appendix* 03 March 2017 ed. London: Network Rail.

NETWORK RAIL 2018a. London North Eastern & East Midlands Route. *Route Strategic Plan.* London: Network Rail.

NETWORK RAIL 2018b. National Railway Timetable - Table 24. *National Railway Timetable.* London: Network Rail.

NETWORK RAIL 2018c. National Railway Timetable - Table 25. *National Railway Timetable.* London: Network Rail.

NICHOLSON, G. L., KIRKWOOD, D., ROBERTS, C. & SCHMID, F. 2015. Benchmarking and evaluation of railway operations performance. *Journal of Rail Transport Planning & Management,* 5**,** 274-293.

NUNEZ, F., REYES, F., GRUBE, P. & CIPRIANO, A. 2010. Simulating Railway and Metropolitan Rail Networks: From Planning to On-line Control. *Intelligent Transportation Systems Magazine, IEEE,* 2**,** 18-30.

ONNASCH, L., WICKENS, C. D., LI, H. & MANZEY, D. 2014. Human Performance Consequences of Stages and Levels of Automation: An Integrated Meta-Analysis. *Human Factors: The Journal of the Human Factors and Ergonomics Society,* 56**,** 476-488.

PARASURAMAN, R., ISAAC, M. & FISHER, D. L. 2012. Reducing Major Rule Violations in Commuter Rail Operations: The Role of Distraction and Attentional Errors. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting,* 56**,** 2331-2334.

PARASURAMAN, R. & RILEY, V. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society,* 39**,** 230-253.

PARASURAMAN, R., SHERIDAN, T. B. & WICKENS, C. D. 2000. A Model for Types and Levels of Human Interaction with Automation. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS,* 30.

PENDER, B., CURRIE, G., DELBOSC, A. & SHIWAKOTI, N. 2013. Disruption Recovery in Passenger Railways. *Transportation Research Record: Journal of the Transportation Research Board,* 2353**,** 22-32.

PRITCHARD, R. 2017. *British railways locomotives & coaching stock 2017* London, Platform 5 publishing ltd.

RASMUSSEN, J. 1983. Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *Systems, Man and Cybernetics, IEEE Transactions on,* SMC-13**,** 257-266.

ROLT, L. T. C. 1966. *Red for danger*, David & Charles.

ROSENBROCK, H. H. 1984. Designing automated systems - need skill be lost? *Trends in Analytical Chemistry,* 3**,** 1-5.

ROTH, E. M., MALSCH, N., MULTER, J. & COPLEN, M. 1999. Understanding how Train Dispatchers Manage and Control Trains: A Cognitive Task Analysis of a Distributed Team Planning Task. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting,* 43**,** 218-222.

RSSB 2011. Braking Principles for Rail Vehicles. *Appendix C.* London: Rail Safety Standards Boards Limited.

SAMENI, M. & PRESTON, J. 2012. Value for Railway Capacity. *Transportation Research Record: Journal of the Transportation Research Board,* 2289**,** 134-144.

SCHLAKE, B., BARKAN, C. & EDWARDS, J. 2011. Train Delay and Economic Impact of In-Service Failures of Railroad Rolling Stock. *Transportation Research Record: Journal of the Transportation Research Board,* 2261**,** 124-133.

SCHLECHTE, T., BORNDÖRFER, R., EROL, B., GRAFFAGNINO, T. & SWARAT, E. 2011. Micro–macro transformation of railway networks. *Journal of Rail Transport Planning & Management,* 1**,** 38-48.

SCHMID, F. 2001. Kempe's engineers year-book. *journal*.

SEIFER, T. 2008. Simulation. *In:* HANSEN, I. A., PACHL, J. & ALBRECHT, T. (eds.) *Railway timetable and traffic: analysis, modelling, simulation.* Hamburg: Hamburg : Eurailpress.

SHARPLES, S., MILLEN, L., GOLIGHTLY, D. & BALFE, N. 2011. The impact of automation in rail signalling operations. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit,* 225**,** 179-191.

SHERIDAN, T. B. 1992. *Telerobotics, automation, and human supervisory control*, MIT press.

SHERIDAN, T. B. 2001. Rumination on automation, 1998. *Annual Reviews in Control,* 25**,** 89-97.

SHERIDAN, T. B. & VERPLANK, W. L. 1978. Human and computer control of undersea teleoperators. DTIC Document.

SOBOL, I. M. 2001. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and computers in simulation,* 55**,** 271-280.

TOMOEDA, A., KOMATSU, M., IL YUN, Y., UCHIDA, M., TAKAYAMA, R., RUI, J. & NISHINARI, K. Real-time railway network simulation and alleviating congestion of crowded trains. ICCAS-SICE, 2009, 18-21 Aug. 2009 2009. 4932-4936.

VANLANDEGEN, L. D. & CHEN, X. 2012. Microsimulation of large-scale evacuations utilizing metrorail transit. *Applied Geography,* 32**,** 787-797.

VENKATESWARAN, K., NICHOLSON, G., CHEN, L. & PELLIGRINI, P. 2017. D3.3.2– Analysis of European best practices and levels of
automation for traffic management under large disruptions *In:* IFFSTAR (ed.) *Capacity for Rail.* UIC.

VENKATESWARAN, K. G., NICHOLSON, G. L., ROBERTS, C. & STONE, R. Impact of Automation on the Capacity of a Mainline Railway: A Preliminary Hypothesis and Methodology. 2015 IEEE 18th International Conference on Intelligent Transportation Systems, 15-18 Sept. 2015 2015. 2097-2102.

VROMANS, M. 2005. *Reliability of Railway Systems.* Ph.D, Erasmus University.

VROMANS, M. J. C. M., DEKKER, R. & KROON, L. G. 2006. Reliability and heterogeneity of railway services. *European Journal of Operational Research,* 172**,** 647-665.

WARD, D. P. 1982. *Untimetabled traffic in a timetabled railway : simulation, control and capacity.* Thesis (Ph.D.) - University of Birmingham, Dept of Electrical and Electronic Engineering, 1983.

WICKENS, C. D., LI, H., SANTAMARIA, A., SEBOK, A. & SARTER, N. B. 2010. Stages and Levels of Automation: An Integrated Meta-analysis. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting,* 54**,** 389-393.

WILSON, G. 1954. III. SAFETY ON THE RAILWAYS. *Journal of the Royal Society of Arts***,** 534-549.

WOLMAR, C. 2005. *On the wrong line: How ideology and incompetence wrecked Britain's railways*, Kemsing Publishing Limited.

WONG, K. K. & HO, T. K. 2004. Coast control for mass rapid transit railways with searching methods. *Electric Power Applications, IEE Proceedings -,* 151**,** 365-376.

WOODS, D. D. & HOLLNAGEL, E. 2006. *Joint cognitive systems: Patterns in cognitive systems engineering*, CRC Press.

YUAN, J. 2008a. *Railway timetable & traffic : analysis, modelling, simulation / editors: Ingo A. Hansen, Jörn Pachl
authors: Thomas Albrecht ... [et al.],* Hamburg, Hamburg : Eurailpress.

YUAN, J. 2008b. Statistical Analysis of Train Delays. *In:* HANSEN, I. A., PACHL, J. & ALBRECHT, T. (eds.) *Railway timetable and traffic: analysis, modelling, simulation.* Hamburg: Hamburg : Eurailpress.

YUAN, J. & HANSEN, I. A. 2007. Optimizing capacity utilization of stations by estimating knock-on train delays. *Transportation Research Part B: Methodological,* 41**,** 202-217.

10 Appendix: Simulation Code

Table 21: Code for running Monte Carlo in JAVA

```
package org.uob.brave;

import java.awt.List;
import java.io.*;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Calendar;
import java.util.Map;
import org.uob.brave.laf.BCRRELookAndFeel;
import org.uob.brave.toolkit.application.events.ShutdownListener;
import org.uob.braveapi.BraveDispatcherContext;
import org.uob.braveapi.dispatcher.EngineRandomiser;
import org.uob.braveapi.dispatcher.KrishnanDispatcher;
import org.uob.braveapi.driver.KrishnanDriver;
import org.uob.braveapi.signaller.FCFS;
import org.uob.braveapi.signaller.kFCFS;

@SuppressWarnings("unused")
public class RunEngine implements EngineRandomiser
{
        public class rdata
        {       public int[] delay;
                public int index;
                public int[] Rtime;
                public int[] Dtime;
                public rdata()
                {
                        Rtime=new int[50];
                        Dtime=new int[50];
                }
                public void adddata(int Rdata,int Ddata, int in)
                {       index=in;
                        Rtime[index]=Rdata;
                        Dtime[index]=Ddata;
                }
                public int getRdata(int index)
                {
                        return Rtime[index];
                }
                public int getDdata(int index)
                {
                        return Dtime[index];
                }
                public int getindex()
                {
                        return index;
                }
        }
```

```
        public static String fname;
        public static int driveflag;
        public static int pfdoors_setflag;
        public static String[] counter;
        public static ArrayList<String> tname;
        public static int[] delay;
        public static double[] dist_change;
        private Map<String, Double> stationmaplist;
        public static double set_time;
        public static double clear_time;
        public static double machine_time;
        public static double dwell_time_mean;
        public static double dwell_time_std;
        public static double dispatch_time;
        public static boolean syncObject=false;
        public rdata thisrun;
        public rdata thisrunroute2;
        private static int route_setflag;

        @SuppressWarnings("serial")
        public class RunBrave extends Brave
        {
                public RunBrave(String[] args)
                {super(args ,new ShutdownListener[] {new ShutdownListener(){
                        public void notifyShutdown() {
                                //                              delaywritetofile();
                                System.out.println("\n Closing Brave");
                        }}
                }, new BCRRELookAndFeel());
                }
        }
        public RunEngine(String[] args) {
                KrishnanDriver.t=this;
                KrishnanDispatcher.t=this;
                kFCFS.t=this;
                dwell_time_mean=Double.parseDouble(args[0]);
                dwell_time_std=Double.parseDouble(args[1]);
                machine_time=Double.parseDouble(args[2]);
                dispatch_time=Double.parseDouble(args[3]);
                set_time=Double.parseDouble(args[4]);
                clear_time=Double.parseDouble(args[5]);
                driveflag=(int) Double.parseDouble(args[6]);
                pfdoors_setflag=(int) Double.parseDouble(args[7]);
                route_setflag=(int) Double.parseDouble(args[8]);
                thisrun= new rdata();
                int j=0;
                for(int i=9;i<args.length;i++)
                        arg.add(args[i]);
                String[] arr = arg.toArray(new String[arg.size()]);
                RunBrave thisr = new RunBrave(arr);
                KrishnanDriver.t=this;
          fname="C:/Users/User/LogBrave/file"+Calendar.getInstance().getTimeInMillis();
```

```
	}
	public static void delaywritetofile() {
		//Writing output data to files
		try{
			BufferedWriter out =
			new BufferedWriter(new FileWriter(fname+"_Delay for
							trains"+".txt",true));
			for(int i=0; i<tname.size();i++)
			{
				out.write(tname.get(i)+":"+delay[i]);
				out.newLine();
			}
			out.close();
		}catch(IOException e)
		{
			System.out.println(e.toString());
		}
	}
	public static void main(String[] arg)
	{
		KrishnanDispatcher.t=new RunEngine(arg);
	}
// functions to fetch sssettings from the main
	public int fetchdriveflag() {
		return driveflag;
	}
	public void setstationlist(Map<String, Double> mapstval) {
		stationmaplist=mapstval;
	}
	public Map<String, Double> getstationlist() {
		return stationmaplist;
	}

	public int s1index=0;
	public int s2index=0;

	public void writedelay(String train, int rtime,int dtime, Double fl) {

		int index=0;
		if(tname.contains(train))
			index=tname.indexOf(train);
		else
		{
			tname.add(train);
			index=tname.indexOf(train);
		}
		thisrun.adddata(rtime, dtime, s2index++);
				}
	}
	public double fetchrandom(int i) {
		// return delay distribution
		if(i == 0)
			return dwell_time_mean;
```

```
		else if(i == 1)
			return dwell_time_std;
		else if(i==2)
			return machine_time;
		else if(i == 3)
			return dispatch_time;
		else if(i == 4)
			return set_time;
		else
			return clear_time;
	}
	public int fetchpfdoorstatus() {
		return pfdoors_setflag;
	}
}
```

Table 22: The code for the driver module

```
package org.uob.braveapi.driver;
import org.uob.braveapi.dispatcher.EngineRandomiser;

public class KrishnanDriver implements APIDriver {
	private BraveDriverContext context;
	private String trainName;
	private double clockTimeStep;
	public double prevSpeed;
	public int prevthrottle;
	private boolean callcount;
		private String[] station_names;
		private double[] station_distance;
	private Map<String, Double> mapstval;
	public static EngineRandomiser t;
	public void initialise(BraveDriverContext context, String
		trainName, double clockTimeStep)
	{
		this.context = context;
		this.trainName = trainName;
		this.clockTimeStep = clockTimeStep;
		this.callcount=false;
		this.prevthrottle=0;
		getStationlist();
	}

	private void getStationlist() {
		mapstval=context.getStationPosition();
		/*for(String key:mapstval.keySet())
			System.out.println(key+ " - "+mapstval.get(key));*/

	}

	public void preDrive(double currentSpeed, double currentDistance,double trainLength, boolean
preventEarlyRunning,int preventEarlyRunningSeconds, int runningTimeDeviation,
			double gradient, String spadNode, boolean trackPositionNull)
	{

		if(t.fetchdriveflag()==0)

	kdrive(currentSpeed,currentDistance,trainLength,preventEarlyRunning,preventEarlyRunningSecond
s, runningTimeDeviation,gradient,spadNode,trackPositionNull);
		else

	atpdrive(currentSpeed,currentDistance,trainLength,preventEarlyRunning,preventEarlyRunningSeco
nds, runningTimeDeviation,gradient,spadNode,trackPositionNull);
	}
	public void kdrive(double currentSpeed, double currentDistance,double trainLength, boolean
preventEarlyRunning,int preventEarlyRunningSeconds, int runningTimeDeviation,
			double gradient, String spadNode, boolean trackPositionNull)
	{
```

```
if(!context.isTrainDwelling())
{
    APISpeedProfileResult spr = context.getSpeedProfileSpeed(currentDistance,
currentSpeed * clockTimeStep, true, trainLength, true, true, !context.isTrainDispatched());
    APISpeedProfileResult spr20 =
context.getSpeedProfileSpeed(currentDistance+20, currentSpeed * clockTimeStep, true, trainLength, true,
true, !context.isTrainDispatched());
    double targetSpeed = spr.getSpeed();
    double lookaheadspeed=spr20.getSpeed();
    /**
     * Add coast boolean if the train driver is preventing early running.
     */
    boolean coast = false;
    if(targetSpeed >= currentSpeed &&
            preventEarlyRunning &&
            runningTimeDeviation < 0- preventEarlyRunningSeconds) {
        coast = currentSpeed > 1.0;
    }
    APIKinematicsResult akr = context.requestPower(currentSpeed, targetSpeed,
false, gradient, coast ? 0 : targetSpeed < currentSpeed ? -1 :1);
    switch (this.prevthrottle) {
    case 0: akr = context.requestPower(currentSpeed, targetSpeed, false, gradient,
coast ? 0 : targetSpeed < currentSpeed ? -1 :0.25);
    this.prevthrottle++;
    break;
    case 1: akr = context.requestPower(currentSpeed, targetSpeed, false, gradient,
coast ? 0 : targetSpeed < currentSpeed ? -1 :0.33);
    this.prevthrottle++;
    break;
    case 2: akr = context.requestPower(currentSpeed, targetSpeed, false, gradient,
coast ? 0 : targetSpeed < currentSpeed ? -1 :0.50);
    this.prevthrottle++;
    break;
    case 3: akr = context.requestPower(currentSpeed, targetSpeed, false, gradient,
coast ? 0 : targetSpeed < currentSpeed ? -1 :0.66);
    this.prevthrottle++;
    break;
    case 4: akr = context.requestPower(currentSpeed, targetSpeed, false, gradient,
coast ? 0 : targetSpeed < currentSpeed ? -1 :0.75);
    this.prevthrottle++;
    break;
    case 5: akr = context.requestPower(currentSpeed, targetSpeed, false, gradient,
coast ? 0 : targetSpeed < currentSpeed ? -1 :1);
    this.prevthrottle=0;
    break;
    }

    if(spadNode != null)
    {
        targetSpeed = 0;
        System.out.print("Look Ahead
Speed:"+Double.toString(lookaheadspeed)+"\n");
        double speed = akr.getSpeed();
        double distanceTravelled = akr.getDistanceTravelled();
```

```
                                        context.driveTrain(speed, distanceTravelled, currentSpeed, targetSpeed,
akr.getEnergyUsed(), akr.getThrottle());
                                }
                                else
                                {
                                        if(!trackPositionNull)
                                        { else train has left the network
                                                int steps=40;
                                                double speed = akr.getSpeed();
                                                double distanceTravelled = akr.getDistanceTravelled();
                                                if(!this.callcount)
                                                { this.prevSpeed=spr.getSpeed()*0.90; }
                                                /**
                                                 * SPAD or hit speed profile wall or discrete stepped driving?
                                                 */
                                                if(spr.getSpeed() < speed - APIDriver.SPAD_THRESHOLD)
                                                {
                                                        distanceTravelled = akr.getDistanceTravelled();
                                                        String spad = spr.getTargetNode();
                                                        context.setSpad(spad);
                                                }
                                                else if(spr.hasDistance() && spr.getSpeed() < currentSpeed)
                                                {
                                                        speed = spr.getSpeed()*0.85-(akr.getSpeed()/steps);
                                                        if(speed<0)
                                                                speed=0;
                                                        this.prevSpeed=currentSpeed;
                                                        distanceTravelled = spr.getDistance() -
currentDistance;
                                                }
                                                else if(speed > lookaheadspeed)
                                                {
                                                        if (currentSpeed>this.prevSpeed)
                                                        {
                                                                speed=this.prevSpeed;
                                                        }
                                                        else
                                                                speed=currentSpeed;
                                                        this.prevSpeed=currentSpeed;
                                                }
                                                /**
                                                 * End SPAD or hit speed profile wall
                                                 */
                                                checking why the train has stopped
                                                if(speed==0)
                                                {
                                                        if( mapstval.containsValue(currentDistance))
                                                        {
                                                                context.notifyDwell();
                                                        }
                                                }
```

```
                                        context.driveTrain(speed, distanceTravelled, currentSpeed,
targetSpeed, akr.getEnergyUsed(), akr.getThrottle());
                                        if(trainName.compareTo("S1")==0)
                                        t.writespeeddata(this.trainName, speed, currentDistance);

        System.out.println(this.trainName+","+speed+","+distanceTravelled+"\n");

                                }
                        }
                } else {
                        /*
                         * Must do this here so kinematics parameters are updated for graphs.
                         */
                        spr = null;
                        context.requestPower(0, 0, false, 0, 0, false);
                }
        }
        /**
         * Notified on train destruction.
         */
        public void notifyDestruction() {
        }

        public void atpdrive(double currentSpeed, double currentDistance,double trainLength, boolean
preventEarlyRunning,int preventEarlyRunningSeconds, int runningTimeDeviation,
                        double gradient, String spadNode, boolean trackPositionNull)
        {
                if(!context.isTrainDwelling()) {
                        APISpeedProfileResult spr = context.getSpeedProfileSpeed(currentDistance,
currentSpeed * clockTimeStep, true, trainLength, true, true, !context.isTrainDispatched());
                        double targetSpeed = spr.getSpeed();
                        /**
                         * Add coast boolean if the train driver is preventing early running.
                         */
                        boolean coast = false;
                        if(targetSpeed >= currentSpeed &&
                                        preventEarlyRunning &&
                                        runningTimeDeviation < 0- preventEarlyRunningSeconds) {
                                coast = currentSpeed > 1.0;
                        }
                        if(spadNode != null) {
                                targetSpeed = 0;
                                APIKinematicsResult akr = context.requestPower(currentSpeed,
targetSpeed, false, gradient, -1);
                                double speed = akr.getSpeed();
                                double distanceTravelled = akr.getDistanceTravelled();
                                context.driveTrain(speed, distanceTravelled, currentSpeed, targetSpeed,
akr.getEnergyUsed(), akr.getThrottle());
                        } else {
                                if(!trackPositionNull) { else train has left the network
                                        APIKinematicsResult akr = context.requestPower(currentSpeed,
targetSpeed, false, gradient, coast ? 0 : targetSpeed < currentSpeed ? -1 : 1);
                                        double speed = akr.getSpeed();
```

```
							double distanceTravelled = akr.getDistanceTravelled();
							/**
							 * SPAD or hit speed profile wall?
							 */
							if(spr.getSpeed() < speed - APIDriver.SPAD_THRESHOLD) {
								distanceTravelled = akr.getDistanceTravelled();
								String spad = spr.getTargetNode();
								context.setSpad(spad);
							} else if(spr.hasDistance() && spr.getSpeed() < currentSpeed) {
								speed = spr.getSpeed();
								distanceTravelled = spr.getDistance() -
currentDistance;
							}
							/**
							 * End SPAD or hit speed profile wall
							 */
							if(trainName.compareTo("S1")==0)
							t.writespeeddata(this.trainName, speed, currentDistance);
							context.driveTrain(speed, distanceTravelled, currentSpeed,
targetSpeed, akr.getEnergyUsed(), akr.getThrottle());
						}
					}
				}
		}

		@Override
		public void dwellUpdate() {

		}
}
```

Table 23: Code for the Signaller module

```
package org.uob.braveapi.dispatcher;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map;
import java.util.Random;
import org.uob.braveapi.BraveDispatcherContext;
import org.uob.braveapi.dispatcher.api.APIDispatcher;
import org.uob.braveapi.dispatcher.api.DispatchRecord;
import org.uob.braveapi.dispatcher.EngineRandomiser;
import org.uob.braveapi.util.APITime;
@SuppressWarnings("unused")
public class KrishnanDispatcher extends DefaultDispatcher
```

```
implements APIDispatcher {
        private Map<String, Double> mapstval;

        private Map<String, Double> arrtraintimes =new HashMap<String, Double>();;

        private Map<String, Double> deptraintimes = new HashMap<String, Double>();

        private Map<String, Double> RT= new HashMap<String,Double>();

        private Map<String, Double> DT= new HashMap<String,Double>();
        public int lastDRSize = 0;
        private BraveDispatcherContext context;
        public ArrayList<DispatchRecord> dispatchRecords = new ArrayList<DispatchRecord>();
        protected APITime startTime = null;
        protected APITime currentTime = null;
        protected int r ;
        public int randomstlistpick=(int)(Math.random()*8+1);
        public String stlist[]={"N2916",
                        "N2955",
                        "N2987",
                        "N2477",
                        "N2539",
                        "N2459",
                        "N2454",
                        "N1902"

        };
        public int temp_delay[]={0,0,0,0,0,0,0,0,0,0,0};
        public int[] stop_tag_array = {0,0,0,0,0,0,0,0,0,0,0};
        public double doors_opentime;
        public double doors_closetime;
        public double alighting_time;
        public double boarding_time;
        private Map<String, Double> fltrain= new HashMap<String, Double>();
        public static EngineRandomiser t;

        public static double at_curr;

        public static double dt_prev;

        public static double rt;

        public static double dwellt;

        @Override
        public void initialise(BraveDispatcherContext context, APITime startTime)
        {
                this.startTime = startTime;
                this.context = context;

                at_curr=0;
                dt_prev=0;
```

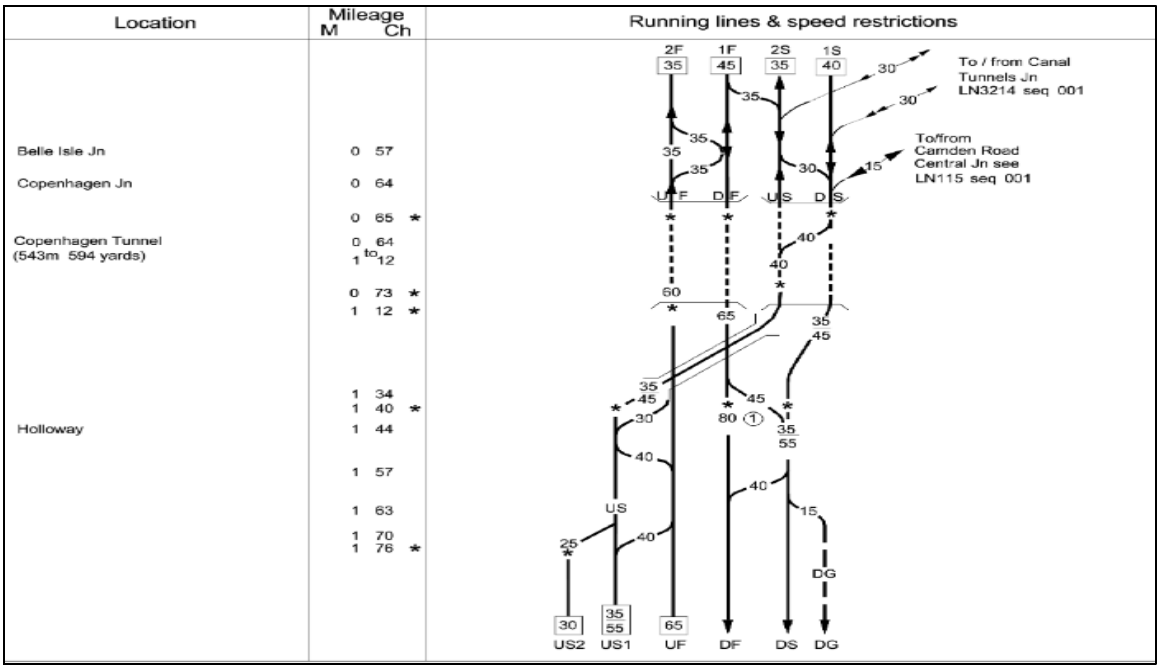
```
                propogate_delay(this.randomstlistpick);


        }
        @Override
        public void reinitialise()
        {
                dispatchRecords.clear();
                randomstlistpick=(int)(Math.random()*8+1);
                propogate_delay(this.randomstlistpick);
                reset_stop_tag_array();
        }

        private void getStationlist() {
                mapstval=context.getStationPosition("S1");

                for(String key:mapstval.keySet())
                        System.out.println(key+ " - "+mapstval.get(key));
        }

        private void reset_stop_tag_array() {
                for(int i=0;i<stop_tag_array.length;i++)
                        stop_tag_array[i]=0;
        }
        @Override
        public void addStoppedTrain(String train, APITime time, APITime scheduledDeparture, String
stopName, int minimumStop, boolean overnight)
        { int temp=0;
        int VC=200;
        int T=(int)0.6*VC;
        double DWF=1.2;
        APITime startclock=new APITime("09:00:00");
        Random rng= new Random(0);

        if(t.fetchpfdoorstatus()==0)
        {
                double dwell_time=rng.nextGaussian()*t.fetchrandome(1)
                        +t.fetchrandom(0);
                double machine_time=t.fetchrandom(2);
                double dispatch_time=t.fetchrandom(3);
                temp=(int)(dwell_time+machine_time+dispatch_time);
        }
        else if(t.fetchpfdoorstatus()==1)
        {
                double dwell_time=t.fetchrandom(0)+ rng.nextGaussian()
                        *t.fetchrandom(1);
                double machine_time=t.fetchrandom(2);
                double dispatch_time=t.fetchrandom(3);
                temp=(int)(dwell_time+machine_time+dispatch_time);
        }
        else if(t.fetchpfdoorstatus()==2)
        {
```

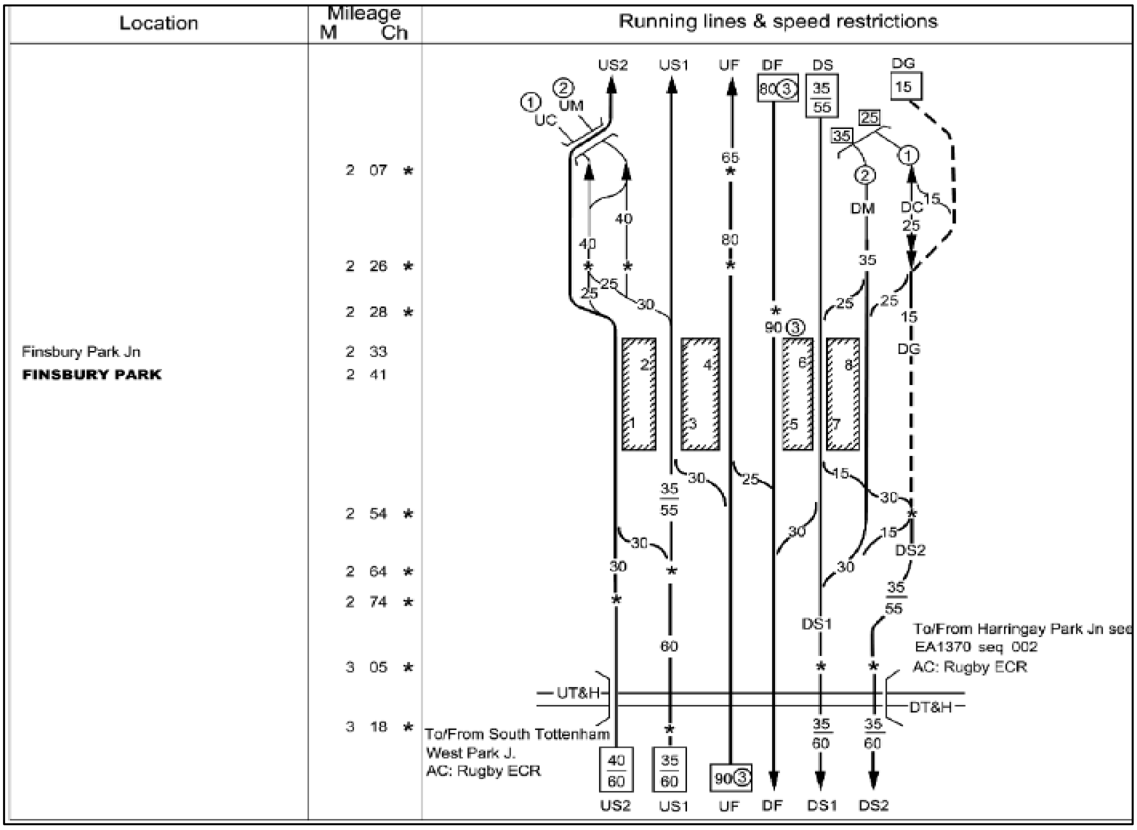
```
                double dwell_time=t.fetchrandom(0);
                double machine_time=t.fetchrandom(2);
                double dispatch_time=t.fetchrandom(3);
                temp=(int)(dwell_time+machine_time+dispatch_time);
        }

        this.r=temp;
        try
        {
                APITime minDwellDispatchTime = time.addSeconds(minimumStop);
                if(scheduledDeparture == null) {
                        scheduledDeparture = minDwellDispatchTime;
                }

        /*      if(stopName.compareTo(stlist[this.randomstlistpick])==0)
                {
                        minDwellDispatchTime
=time.addSeconds(temp+temp_delay[this.randomstlistpick]);
                }
                else*/
                        minDwellDispatchTime =time.addSeconds(temp);
//+temp_delay[station_node_id]);

                if(minDwellDispatchTime.compareTo(scheduledDeparture) > 0)
                {
                        dispatchRecords.add(new DispatchRecord(minDwellDispatchTime, train));
                        int delay = minDwellDispatchTime.getSecondsSinceMidnight()-
time.getSecondsSinceMidnight();

                        if(arrtraintimes.containsKey(train) == false)
                                {
                                        arrtraintimes.put(train, (double) time.getSecondsIntoDay());
                                        fltrain.put(train, 0.0);
                                        DT.put(train, (double) delay);
                                }
                        else
                                {
                                        deptraintimes.put(train, (double)time.getSecondsIntoDay());
                                        if( fltrain.get(train) == 0)
                                        {
                                                RT.put(train, (double) (deptraintimes.get(train)-
arrtraintimes.get(train)-delay));
                                                fltrain.put(train, 1.0);

                                        }
                                        else
                                                RT.put(train, (double) (deptraintimes.get(train)-
arrtraintimes.get(train)-delay+RT.get(train)));

                                arrtraintimes.put(train, deptraintimes.get(train));
                                DT.put(train, delay+this.DT.get(train));
                                }
```

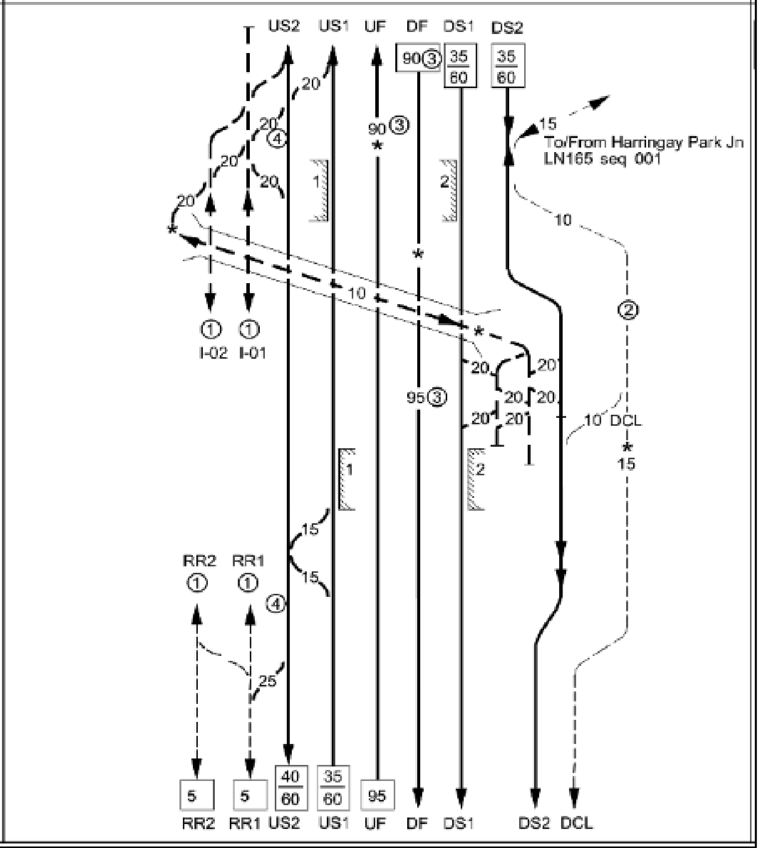
```
                } else
                {
                        dispatchRecords.add(new DispatchRecord(scheduledDeparture, train));
                        int delay = scheduledDeparture.getSecondsSinceMidnight()-
time.getSecondsSinceMidnight();

                        if(arrtraintimes.containsKey(train) == false)
                        {
                                arrtraintimes.put(train, (double) time.getSecondsIntoDay());
                                fltrain.put(train, 0.0);
                                DT.put(train, (double) delay);
                        }
                        else
                                {
                                        deptraintimes.put(train, (double)time.getSecondsIntoDay());
                                        if( fltrain.get(train) == 0)
                                        {
                                                RT.put(train, (double) (deptraintimes.get(train)-
arrtraintimes.get(train)-delay));
                                                fltrain.put(train, 1.0);

                                        }
                                        else
                                                RT.put(train, (double) (deptraintimes.get(train)-
arrtraintimes.get(train)-delay+RT.get(train)));

                                arrtraintimes.put(train, deptraintimes.get(train));
                                DT.put(train, delay+this.DT.get(train));
                                }


                }
        } catch(Exception e) {
                System.err.println(e);
                e.printStackTrace();
        }
        }

        @Override
        public void update(APITime time) {
                currentTime = time;
                if(lastDRSize != dispatchRecords.size()) {
                        Collections.sort(dispatchRecords);
                }
                while(dispatchRecords.size() > 0 && dispatchRecords.get(0).getTime().compareTo(time)
<=0) {
                        DispatchRecord dr = dispatchRecords.remove(0);
                        String train = dr.getTrain();
                        context.dispatchTrain(train);
                }
                lastDRSize = dispatchRecords.size();
        }
```

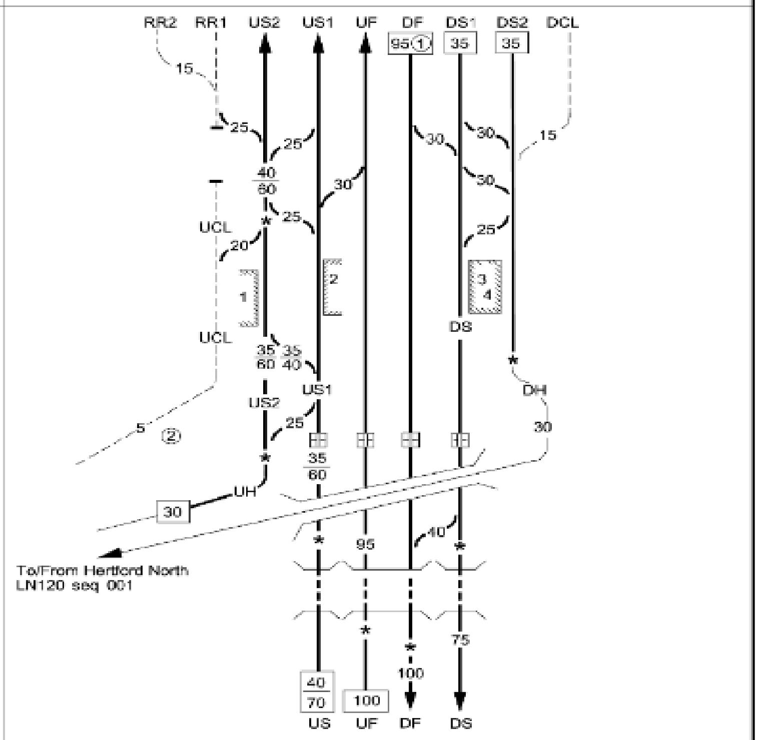
```
        @Override
        public void notifyTrainDestruction(String destroyedTrain)
        {
        try{
                t.writedelay(destroyedTrain,(int) Math.round(this.RT.get(destroyedTrain)),(int)
Math.round(this.DT.get(destroyedTrain)),fltrain.get(destroyedTrain));
        }catch(Exception e){e.printStackTrace();}
                }

}
```

11 Appendix: Detailed Route Information for the Section between Kings Cross and Welwyn Garden City



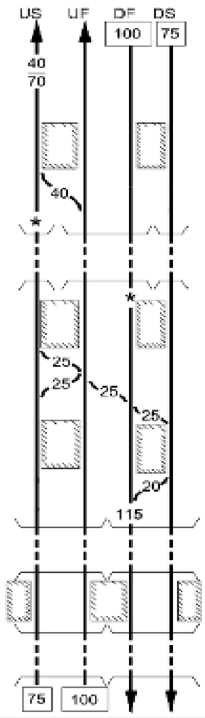


Location	Mileage M Ch	Running lines & speed restrictions
Harringay Jn	3 29 *	
HARRINGAY	3 32	
	3 34 *	
	3 37 *	
Harringay Viaduct	3 34 to 3 40	
	3 61 *	
Ferme Park Sidings (Down Side) Hornsey Depot (Up Side)		
	3 77 *	
HORNSEY	4 04	
	4 20	
	4 30	

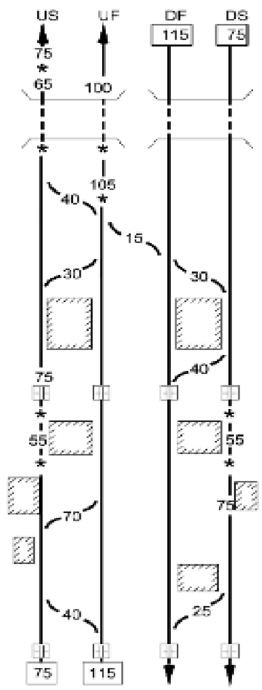
Location	Mileage M Ch	Running lines & speed restrictions
	4 38	
	4 60 to 4 63	
Wood Green South Jn	4 68	
	4 70 *	
	4 75	
ALEXANDRA PALACE	4 78	
	5 04 *	
Wood Green North Jn	5 07	
Wood Green F.S. OHNS	5 15	
	5 17 *	
	5 22	
	5 35 *	
Wood Green Tunnels (644m / 705 yards)	5 41 to 5 73	
	5 73 *	
	5 76 *	

Appendix: Detailed Route Information for the Section between Kings Cross and Welwyn Garden City

Location	Mileage M	Ch	Running lines & speed restrictions
NEW SOUTHGATE	6	35	
	7	40 *	
Barnet Tunnel (553m 605 yards)	7 to 7	42 to 70	
	7	73 *	
OAKLEIGH PARK	8	30	
Barnet South Crossovers	8 to 9	74 to 00	
NEW BARNET	9	12	
Barnet North Crossover	9	18	
Hadley Wood South Tunnel (351m 384 yards)	10 to 10	21 to 39	
HADLEY WOOD	10	46	
Hadley Wood North Tunnel (212m 232 yards)	10 to 10	60 to 70	



Location	Mileage M	Ch	Running lines & speed restrictions
	11	23 *	
Potters Bar Tunnel (1110m 1214 yards)	11 to 12	25 to 00	
	12	03 *	
	12	36	
	12	40 *	
	12	53	
POTTERS BAR	12	57	
Potters Bar TSC OHNS	13	21	
	14	25 *	
BROOKMANS PARK	14	37	
	14	47 *	
WELHAM GREEN	15	50	
Marshmoor	16	06	
HATFIELD	17	54	
Welwyn F.S. OHNS	19	29	



Location	Mileage M	Ch	Running lines & speed restrictions
	19	63 *	
	19	65 *	
Welwyn Garden City Up Yard	20	25	
WELWYN GARDEN CITY	20	25	
	21	07 *	
Digswell	21	18	
	21	24 *	
	21	36