

DITTO Project Deliverable 4.1

Milestone 3

Interim Report and Prototype

April 2016

Version 2

Professor John Preston, Dr John Armstrong and Dr Attila Kovacs, University of Southampton.

Dr Ronghui Liu and Dr Hongbo Ye, University of Leeds.

Dr Phillip James, Prof Faron Moller, and Prof Markus Roggenbach, Swansea University.

Table of Contents

| | |
|--|----|
| Abstract | 7 |
| 1. Introduction and Project Outline | 8 |
| 1.1 Background and objectives | 8 |
| 1.2 Outline of the deliverable | 11 |
| 2. Test Environments | 13 |
| 2.1 The Four Station Model and TrackULA | 13 |
| 2.1.1 Background and objectives | 14 |
| 2.1.2 The train dynamics model | 15 |
| 2.1.3 Stability analysis of the controlled train-following model | 20 |
| 2.1.4 A TrackULA simulation test environment | 24 |
| 2.1.5 A multi-train trajectory optimisation model | 29 |
| 2.1.6 Conclusions | 36 |
| 2.2 Stochastic Optimisation | 37 |
| 2.2.1 Model | 37 |
| 2.2.2 Solution approaches | 41 |
| 2.2.3 Trade-off between reliability and utilisation | 44 |
| 2.2.4 Experiments | 44 |
| 2.2.5 Conclusions and further work | 53 |
| 3. Capacity Utilisation Indices Calculations | 54 |
| 3.1 Introduction | 54 |
| 3.2 Methodology | 54 |
| 3.2.1 Data and network modelling | 54 |
| 3.2.2 Data processing | 57 |
| 3.3 Ongoing and further work | 58 |
| 3.4 Conclusions | 59 |
| 4. Prototype Development | 60 |
| 4.1 BRaVE and OnTrack | 60 |
| 4.1.1 Data sets: East Coast main line | 61 |
| 4.2 A First Integration | 61 |
| 4.2.1 Challenges in Abstraction Levels | 63 |

| | | |
|-------|---|----|
| 4.2.2 | Challenges in Verification due to Differences in Modelling | 64 |
| 4.3 | Verification of the Translated Models | 65 |
| 4.4 | Conclusions and Further Work | 66 |
| 5 | Interaction with Other T1071 Projects (DEDOTS and Safecap Plus) | 68 |
| 6 | Implementation Plan | 69 |
| 7 | Conclusions and Next Steps | 70 |
| 8 | References | 71 |

List of Figures

| | |
|---|----|
| Figure 1.1: Safety and Capacity Validation | 9 |
| Figure 1.2: Static Optimisation | 10 |
| Figure 1.3: Dynamic Rescheduling | 11 |
| Figure 2.1: Definition of a basic train following situation | 14 |
| Figure 2.2: A continuous profiles of acceleration and jerk vs speed | 15 |
| Figure 2.3: Trajectories of acceleration- and speed- with time (a) and with distance (b) | 16 |
| Figure 2.4: A trajectory for approaching a station or signal, with a given braking speed V_b | 17 |
| Figure 2.5: Trajectory for approaching a station or signal from a distance s to stop: (a) earlier coasting; (b) later coasting | 19 |
| Figure 2.6: Stability conditions in the space of (α, S) for different values of: (a) safe following time T , and (b) response time t_d | 22 |
| Figure 2.7: Space-time evolution of the following headways for: (a) $\beta = 0.05$, and (b) $\beta = 0.1$ | 23 |
| Figure 2.8: The test network for two different designs for the platforms at stations B and C: (a) platforms on the mainline track; and (b) platforms on passing loops off the mainline | 24 |
| Figure 2.9: Space-time trajectories in Scenario I. The two dashed lines are those of the two delayed departure trains number 2 and 3, while the solid lines represent the trains who departed on time | 26 |
| Figure 2.10: Space-time trajectories for Scenario II where train number 2 (dashed) had to make an emergency stop mid-way between stations B and C | 27 |

| | |
|---|----|
| Figure 2.11: Space-time trajectories of a group of 16 fast and slow trains traversing the network Figure 2.8a. | 28 |
| Figure 2.12: Relative speed and space distance between the slow train 2 and its following fast train 3, for test scenario based on network design of Figure 2.8a. | 28 |
| Figure 2.13: Space-time trajectories of a group of 16 fast and slow trains traversing the network Figure 2.8b. | 29 |
| Figure 2.14: Energy-efficient train trajectory with level tracks and constant speed limit | 30 |
| Figure 2.15: Illustration of Phases | 34 |
| Figure 2.16: Optimal train trajectories and operations | 35 |
| Figure 2.17: Example of a network layout | 38 |
| Figure 2.18: Example of an event-activity graph | 39 |
| Figure 2.19: Potential conflicts at a junction of two double-track lines | 40 |
| Figure 2.20: No-wait activities are incorporated to avoid sequence 2-3-1 | 41 |
| Figure 2.21: Transformation from event activity-model into job-shop model | 43 |
| Figure 2.22: Differences in the probability of being delayed | 47 |
| Figure 2.23: Delay distribution of Train 0 | 48 |
| Figure 2.24: QQ-plot for different distributions for Train 0 | 48 |
| Figure 2.25: Delay distribution of Train 6 | 49 |
| Figure 2.26: QQ-plot for different distributions for Train 6 | 49 |
| Figure 2.27: Example rail network | 50 |
| Figure 2.28: Set of trains to be scheduled, given in a timetable and illustrated on a time-space graph | 51 |
| Figure 2.29: Solution timetable that satisfies all constraints | 52 |
| Figure 2.30: Junction in solution timetable | 53 |

| | |
|---|----|
| Figure 3.1: Initial ECML Model Area | 56 |
| Figure 3.2: Current Peterborough station layout | 57 |
| Figure 4.1: Barkston South Junction Werrington Junction as displayed in BRaVE | 62 |
| Figure 4.2: Examples of BRaVE and OnTrack Data | 62 |
| Figure 4.3 Algorithm to Construct Topological Route | 64 |
| Figure 6.1: Rail Technical Strategy | 69 |

List of Tables

| | |
|---|----|
| Table 2.1: Categorisation of train status in a phase | 31 |
| Table 2.2: Track gradient in the form: gradient (‰)/section (m) | 33 |
| Table 2.3: Original schedule | 33 |

Abstract

This milestone constitutes the approximate halfway point in the DITTO project. It demonstrates the progress we have made in developing a stochastic optimisation scheduling model to develop robust timetables and shows that this method is feasible using a test environment. It also demonstrates the progress we have made in enhancing the dynamic simulation model, Trackula, to work with ERTMS Level 3 and in determining optimal train trajectories. Using another test environment, we show how this modelling work can lead to stable operations. We also detail how we have assembled data for a study of the East Coast Main Line, with an emphasis on the section between Retford and Huntingdon and a particular focus on Peterborough. We outline the progress we have made in developing nodal Capacity Utilisation Indices and in examining their relationship with Congestion Related Reactionary Delays. This will feed into the stochastic optimisation model of the East Coast Main Line. We outline the latest developments with OnTrack, an open toolset for railway verification, which will be our core interfacing tool. We demonstrate the interface between OnTrack and BRaVE (Birmingham Railway Virtual Environment) with reference to the East Coast Main Line.

1. Introduction and Project Outline

1.1 Background and objectives

Previous research, including the OCCASION project, has shown that rail capacity is constrained by nodes – particularly stations (see, for example, Armstrong et al., 2015). For example, Peterborough station’s ‘practical’ capacity (morning peak) is around 4.4 trains per track per hour¹, whilst Peterborough station’s ‘maximum’ capacity was estimated at around 5.6 trains per track per hour. However to a lay-person, a train every 10 minutes or so hardly seems busy but the capacity limitations are apparent when additional high speed paths for open access operators are sought (Network Rail, 2014). Hence DITTO is looking to investigate whether improved static and dynamic scheduling, in conjunction with advances in signalling technology, can lead to a step change in capacity limits. In particular, DITTO will examine whether the theoretical increases in capacity offered by ERTMS/ETCS Level 3 (Wendler, 2007) can be achieved in practice.

In order to investigate this issue DITTO will develop integrated tools to optimise the use of rail capacity and this will involve four main objectives.

1. Develop optimisation procedures that maintain safe operating conditions and do not exceed theoretical capacity limits. This is designated Work Area 1, Safety.
2. Develop timetables that optimise capacity utilisation without compromising service reliability. This is designated Work Area 2, Reliability.
3. Combine dynamic data on the status of individual trains to produce an optimal system-wide outcome. This is designated Work Area 3, Dynamics.
4. Use Artificial Intelligence to produce tractable solutions to real-time traffic control. This is designated Work Area 4, Integration.

The DITTO approach is illustrated by Figures 1.1 to 1.3 (see also Milestone 2).

¹ If we simplify the Peterborough layout, as it existed at the time of the OCCASION study, to consist of six tracks. In practice, Peterborough was modelled as an assemblage of 190 links and nodes.

Figure 1.1: Safety and Capacity Validation.

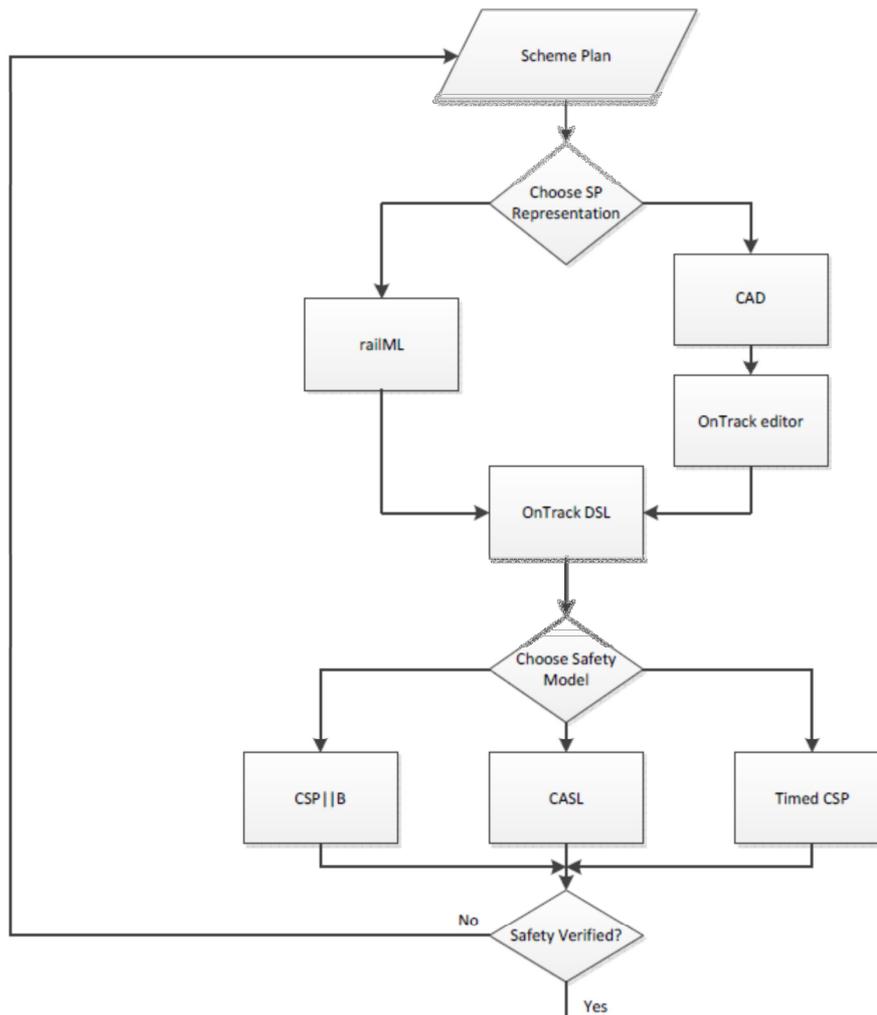
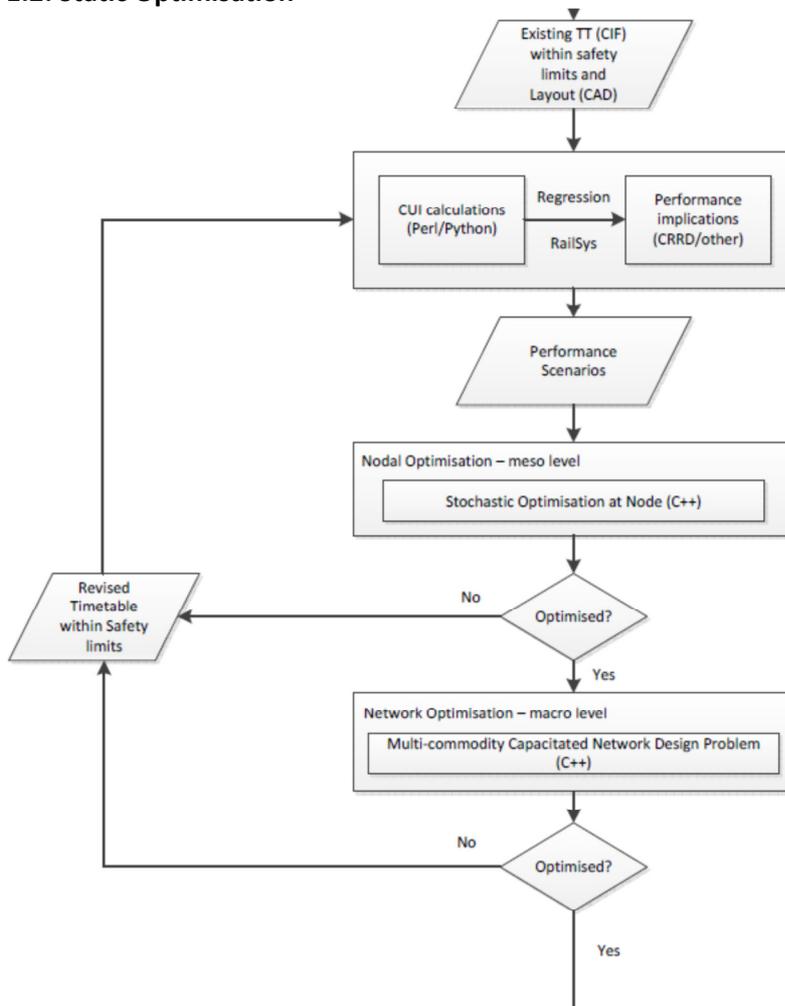


Figure 1.1 shows that the starting point is to put a Scheme Plan (SP) through a safety and capacity verification process.. This might use either RailML or output from Computer Assisted Design (CAD) and the OnTrack editor. These approaches are brought together by the OnTrack Domain Specific Language (DSL) detailed in Milestones 1. Safety verification can then be performed using a variety of languages such as: CSP, Communicating Sequential Processes; a specification language for concurrent systems defined by Sir Tony Hoare in the early 1980s; CSP Parallel B, a combination of CSP and the specification language B, defined by Swansea's research partners at Surrey University around 2000; and CASL, Common Algebraic Specification Language. Future work will develop the safety validation based on CSP and the capacity validation based on Timed CSP building on successful experiments at Swansea.

Figure 1.2 shows that the next stage is to undertake the static optimisation, the early development of which was outlined in Milestones 4 and 5. This takes the existing Timetable (TT) in CIF (Common Interface Format) and the safety limits established by the verification and using Capacity Utilisation Indices or other related approaches assesses

the likely performance in terms of Congestion Related Reactionary Delay (CRRD). Performance scenarios are then developed to feed into a stochastic optimization based on a variant of job shop scheduling, in which tracks are treated as machine shops and train movements are treated as jobs (Liu and Kozan, 2009). This involves a two stage stochastic program. In the first stage, new trains are inserted into the timetable. The second stage involves optimising for reliability for various random scenarios. This is undertaken at the meso-level, for example for a node such as Peterborough on the East Coast Main Line (ECML). The implications are assessed at a macro-level, for example for the ECML between Retford and Huntingdon using a variant of the Multi-Commodity Network Design Problems (MCNDP). Constraints ensure that the revised timetable is within safety limits. Once the optimisation is confirmed at the meso- and macro-levels, it is fed into the final stage.

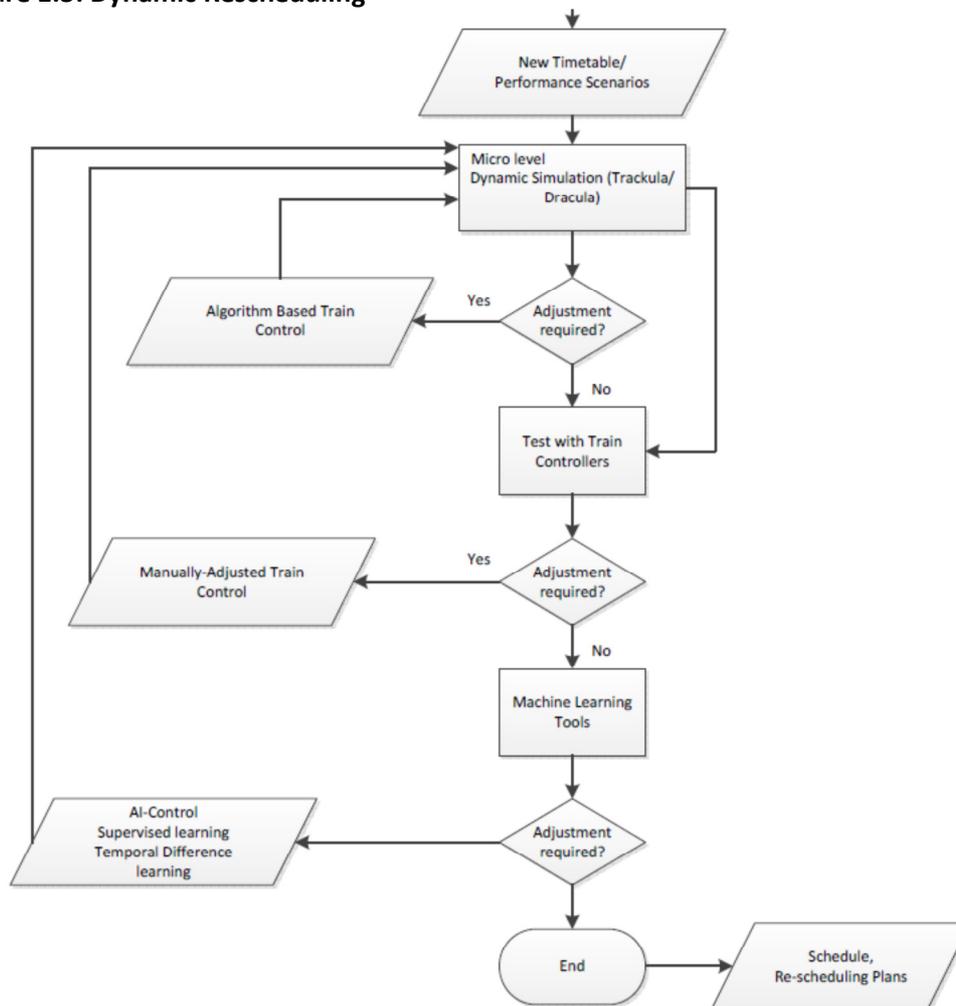
Figure 1.2: Static Optimisation



The final stage involves examining the scope for dynamic rescheduling and this is done by using the Trackula train simulator, along with examining traditional algorithms, alongside human control and artificial intelligence based on machine learning. This is

informed by historic data on performance (in terms of delays) that has also informed the static optimisation but may consider a wider range of scenarios. The early developments of this work were described in Milestone 6. The final output is an optimised table, along with a series of rescheduling plans, if needed.

Figure 1.3: Dynamic Rescheduling



1.2 Outline of the deliverable

This interim report will outline some of the progress we have made in operationalising Figures 1.1 to 1.3. In section 2, we will outline the progress we have made in developing the dynamic and static optimisations. In section 3, we give some details of the ECML Case Study and of the work on capacity assessment. In section 4, we outline some of the progress we have made in developing a prototype. In section 5, we further examine the inter-relationships with other T1071 projects. In section 6, we consider the

implementation plan, whilst in section 7 we draw some brief conclusions and implications for further work.

2. Test Environments

2.1 The Four Station Model and TrackULA

2.1.1 Background and objectives

The moving-block (MB) ERTMS Level 3 system represents a continuous detection of train positions, and continuous train-to-train/train-to-infrastructure data transmission. Trains can follow one another in smaller (space and time) headways than possible in fixed-block (FB) systems. MB allows the formation of platoons of trains with undisturbed headways and uniform speeds, thus improved safety and capacity. Trains travelling in a platoon (or 'virtually coupled' mode) also have the potential for energy savings due to reduced running resistance.

However, the dynamics of such train-following and platooning are likely to be critical, as this can give rise to several practical effects, including unrealistic performance demand on braking and traction forces, unstable following and incessant speed fluctuations. Dynamic simulation models can be used to design and test train control algorithms and design solutions to these problems.

DITTO Deliverable 3.1/Milestone 6 (Liu et al., 2015) reported on the development of the TrackULA microsimulation model to represent the MB system under ERTMS Level 3 control. A train-following model was developed to represent the 'movement command' on the following train's speed in response to a desired headway and to the speed of the train in front. The train-following model was tested in a corridor network of four-stations and with different traffic mixes. The results show that the model is capable of reproducing the acceleration of trains when the gap in front is large, the slowing down of trains when the gap reduces, and the deceleration and acceleration to/from stations.

This report summarises the work following from Deliverable 3.1 in September 2015 on modelling the ERTMS Level 3 control systems. Sub-section 2.1.2 presents a new train acceleration and a new mathematical formulation of the deceleration model, for a free-running train (without the constraint of a lead train on the same track). A detailed controlled train-following model, based on the concept of an optimal velocity and desired following headway is presented, and the practical relevance of the model variables and parameters explained.

Sub-section 2.1.3 analyses the performance of the proposed train-following model, in terms of safety and flow stability. Results are presented to show the sensitivity of the control mechanism to the choice of model parameter values. The dynamic performance of the control mechanism is illustrated in a simulation test, showing stop-and-go congestion and delay propagation. We discuss the practical implications of the train-following formulation and the choice of model parameter values. The analysis

demonstrates the role simulation models can play in designing good train control algorithms.

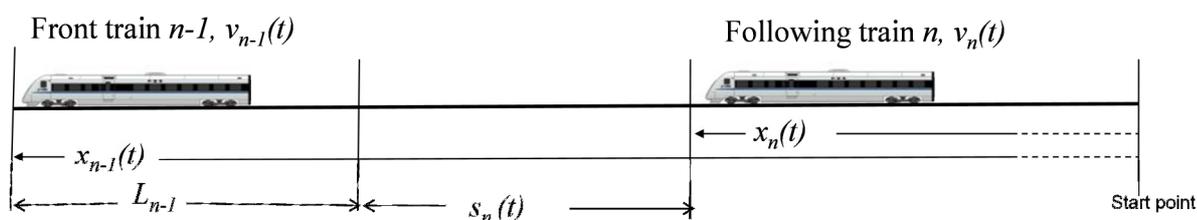
Sub-section 2.1.4 of this report presents a new development of the TrackULA model to include platforms off the main track (i.e. on passing loops) so as to allow overtaking at stations. We present example tests to illustrate the workings of this new model.

In simulation models, the trains' speeds and trajectories are determined by the train-following mechanism. They respond to the instantaneous local conditions; their overall trajectories are the results of the dynamical interactions with other trains and with network conditions. In sub-section 2.1.5, we present an optimal control formulation to obtain optimised trains trajectories which minimise energy consumption. The method proposed takes account of track topology and simultaneously optimises the trajectories of coupled (or closely-following) trains. We demonstrate the method in solving a two-train trajectory optimisation problem in a network of four stations.

2.1.2 The train dynamics model

Consider a simple train-following situation as illustrated in Figure 2.1, where train n follows train $n-1$ on a single track. The variable $x_n(t)$ denotes the position of train n as measured from an arbitrary starting point, whilst $v_n(t)$ is its velocity at time t . L_{n-1} is the 'effective' size of train $n-1$, which includes the physical length of the train plus a safe margin.

Figure 2.1: Definition of a basic train following situation.



The train dynamics model is composed of three phases: (1) an acceleration model for when a train accelerates freely away from stations; (2) a controlled train-following model for when a train's movement is constrained by that of the train in front; and (3) a deceleration model for when a train approaches and aims to stop at a station.

(1) A continuous acceleration model

In Deliverable 3.1, we described a 'free-flow' train driving cycle (Fig. 6a of Deliverable 3.1) to represent the dynamics of the lead train on the line. In that model, the acceleration of the train is set according to a constant jerk and a constant maximum acceleration. As a result, the acceleration profile of the train is discontinuous (Fig. 6b of Deliverable 3.1). In practice, train acceleration characteristics vary with the train's

speed, power to weight ratio, and resistance force. Vehicles' acceleration rates are also shown to decrease with speed (John and Kobett, 1978). When a train starts from stationary, it takes a large maximum accelerate rate. When the speed of the train approaches the speed limit, its acceleration decreases until the traction force is just so that the train can maintain a constant speed.

To describe the above acceleration behaviour, we use the following equation to represent the concave relationship between the acceleration and the speed of a train:

$$a(v) = \gamma_1 A (1 - v/V) \sqrt{\gamma_2 + v/V} \quad (1)$$

where $a(v)$ is the acceleration at velocity v , A is the maximum acceleration of the train, V is the maximum speed of the train (or the speed limit on the track, whichever is smaller), and γ_1 and γ_2 are two constants. With $\gamma_1 = 2.5$ and $\gamma_2 = 0.025$, eq. (1) becomes the Gipps' free-flow car-following model (Gipps, 1981).

Suppose we set $A = 1.0 \text{ m/s}^2$, $V = 200 \text{ km/hr}$ (or 55.56 m/s), $\gamma_1 = 2.5$ and $\gamma_2 = 0.025$, Figure 2.2 shows the relationships between acceleration and jerk with speed. It can be seen that the acceleration increases firstly with speed to the maximum at speed around 18 m/s ($\sim 65 \text{ km/hr}$), before decreases with speed to zero when the speed reaches its maximum at 55.5m/s. The rates of change in acceleration (jerk) also vary continuously with speed and are in a comfortable range between -0.03 and 0.06 m/s^3 . The corresponding trajectories of acceleration and speed vs time (and distance) are shown in Figures 2.3a and 2.3b.

Figure 2.2: A continuous profiles of acceleration and jerk vs speed.

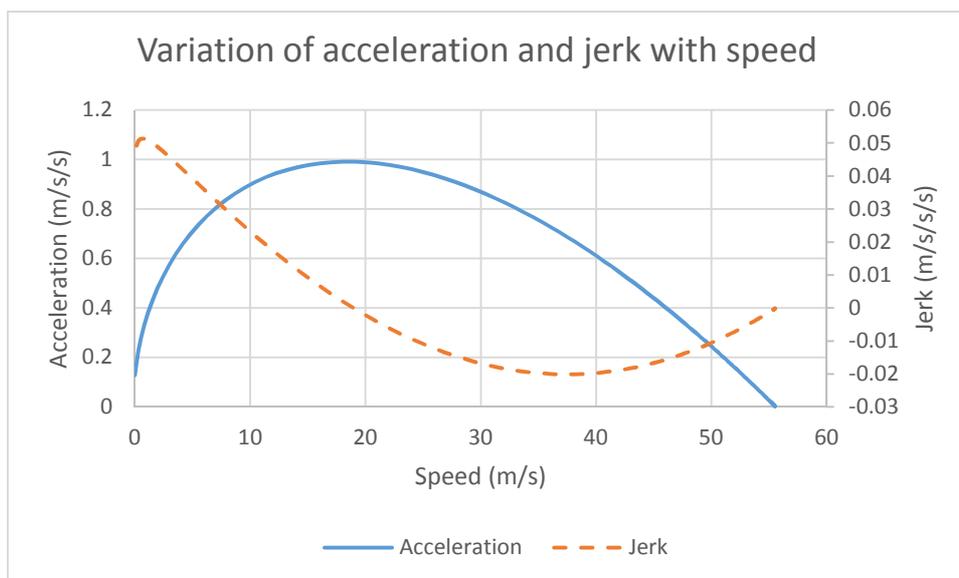
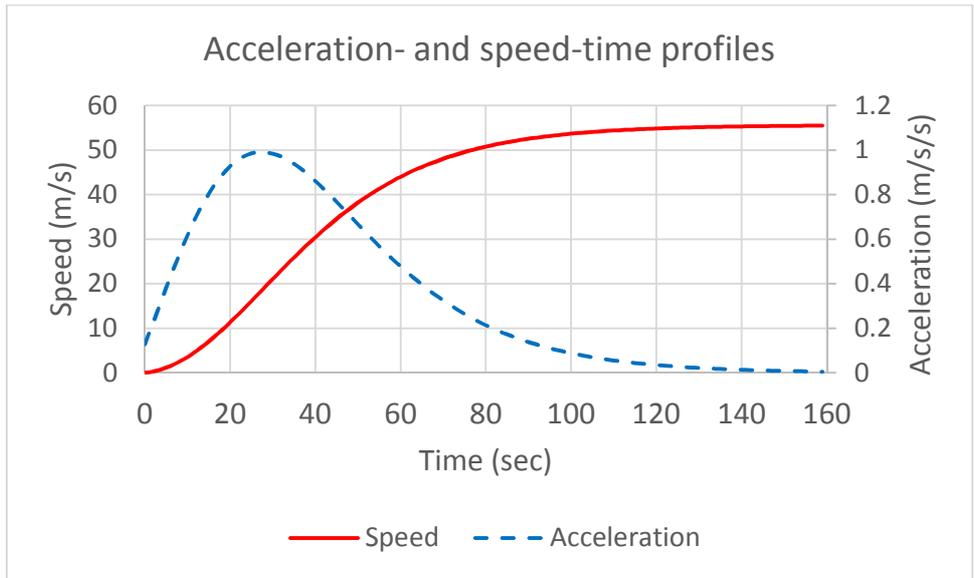
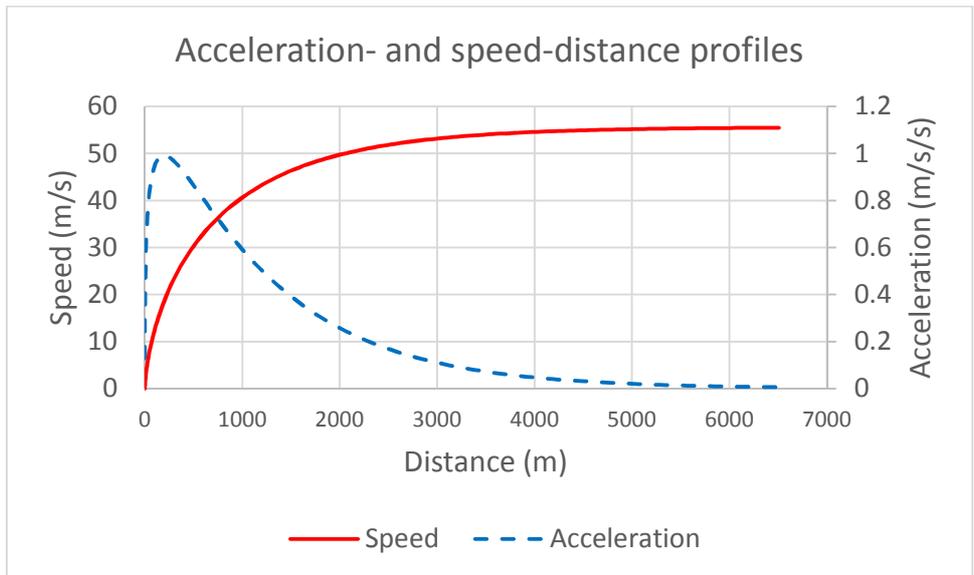


Figure 2.3: Trajectories of acceleration- and speed- with time (a) and with distance (b).



(a)

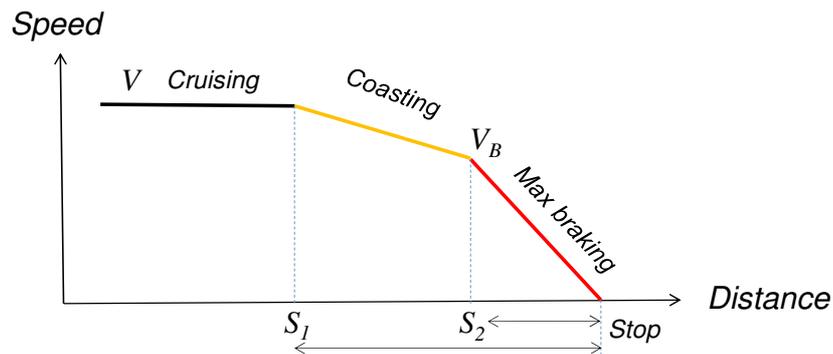


(b)

A coasting-maximum braking model

When approaching a station or a signal point where it is required to stop, a train may first let the resistance force (a combination of track resistance and train running resistance) reduce its speed to a lower ‘braking speed’, before applying a braking force to stop by the station/signal point. We call this driving state unconstrained (or free) coasting and braking, as opposed to deceleration in a train-following state (see the controlled-following state below). Figure 2.4 illustrates an approaching trajectory, starting from a cruising state, undergoing a coasting state before a maximum braking state. The distance to stop at the start of the coasting state is S_1 , and at the start of the maximum braking state is S_2 .

Figure 2.4: A trajectory for approaching a station or signal, with a given braking speed V_b .



We construct the following analytic equation to capture the full braking curve which encompasses a coasting and a maximum braking state, with equations (2) and (3).

$$v(t) = \begin{cases} V - d_c t & 0 \leq t < t_B \\ V_B - d_{\max}(t - t_B) & t \geq t_B \end{cases} \quad (2)$$

with

$$t_B = \frac{V - V_B}{d_c} \quad (3)$$

where V is the maximum (or cruise) speed as that in eq. (1), V_b is the braking speed, d_c and d_{\max} are the decelerations applied at the coasting state and during maximum braking respectively and are modelled as constants.

With a given braking speed V_b , a train needs to start coasting at a distance S_1 away from the stopping point and start to apply its maximum braking force from a distance S_2

away from the stopping point. The deceleration points S_1 and S_2 can be derived from the cruising speed and the braking speed as:

$$S_1 = \frac{V^2 - V_B^2}{2d_c} + \frac{V_B^2}{2d_{\max}} \quad \text{and} \quad S_2 = \frac{V_B^2}{2d_{\max}}. \quad (4)$$

There may be situations where it is necessary for the train to start coasting earlier, or to brake later. For example, if a train is running earlier than schedule and the platform in the station is not clear, it is more advantageous to command the train start coasting earlier to save energy. The train trajectory for this case is illustrated in Figure 2.5a, where it starts to coast from a distance s_1 to a stop line (where $s_1 < S_1$), until its speed is reduced to v_b when it applies maximum brake to stop at the stop line.

Figure 2.5b illustrates a scenario where a train starts coasting later, say at a distance s from the stop line, where $S_1 < s_1 < S_0$. S_0 is the closed distance to the stop where the train can travel with its maximum speed and be able to stop at its maximum braking power at the stop line.

$$S_0 = \frac{V^2}{2d_{\max}}.$$

Thus, the analytic equations (2) and (3) for calculating the reference speed for the train during its coasting-braking states can be generalised as follows:

$$v(t) = \begin{cases} V - d_c t & 0 \leq t < t_b \\ v_b - d_{\max} (t - t_b) & t \geq t_b \end{cases} \quad (5)$$

and equation (4) can be generalised as:

$$s_1 = \frac{V^2 - v_b^2}{2d_c} + \frac{v_b^2}{2d_{\max}} \quad \text{and} \quad s_2 = \frac{v_b^2}{2d_{\max}}. \quad (6)$$

where s_1 is the distance to stop where the train starts coasting, subject to the constraint $S_1 < s_1 < S_0$. s_2 is the distance to stop where the train must apply its maximum deceleration d_{\max} in order to stop in time.

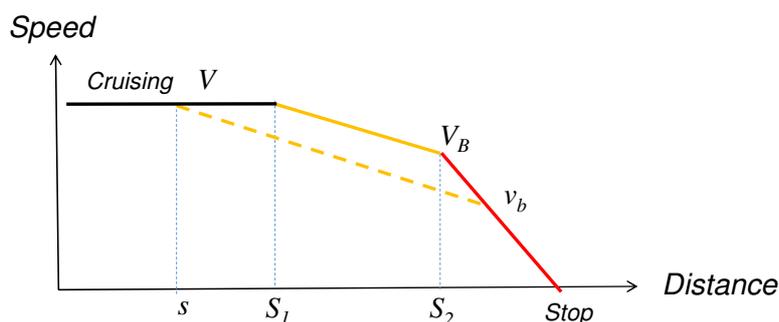
From equation (6), we can derive also the speed at the end of the coasting phase:

$$v_b = \sqrt{\left(\frac{V^2}{d_c} - 2s\right)\left(\frac{d_c d_{\max}}{d_{\max} - d_c}\right)}$$

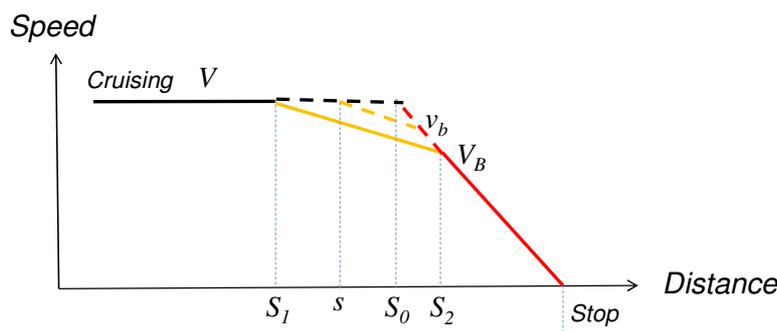
Putting (7) to (5), we get the coasting time period :

$$t_b = \frac{V - v_b}{d_c}$$

Figure 2.5: Trajectory for approaching a station or signal from a distance s to stop: (a) earlier coasting; (b) later coasting.



(a)



(b)

A controlled following model

When several trains are moving on the same track following one another in a moving block (MB) system, their desired movements will be constrained by the movement state of the train(s) in front. We formulate such constrained movements as a controlled train-following problem, with the control mechanisms designed to mimic the movement command in a MB ERTMS Level 3 system.

Let $v_n(t)$ denotes the velocity of a train n at time t , and $s_n(t)$ the distance gap behind the train in front. We formulate a controlled train-following model to predict the acceleration of the train at the next time instance time $t + \Delta t$:

$$\frac{dv_n(t + \Delta t)}{dt} = \alpha[V^*(s_n(t - t_d)) - v_n(t)] + \beta[s_n(t) - s_n^{des}] \quad (7)$$

where $V(\cdot)$ is an optimal velocity for a train n following at a distance gap s_n behind the train in front, and s_n^{des} is a desired headway for train n .

Equation (7) is adapted from a multi-anticipatory model proposed by Chen and Liu (2016) to model car following behaviour. It is an extension of a strand of well-studied car-following models based on the concept of an ‘optimal velocity’. The concept was first proposed by Bando et al. (1995); the difference between the optimal velocity and the velocity of the considered vehicle is assumed to be a stimulus for driver’s actions. Chen and Liu (2016) consider the desired following distance as another explicit stimulus and formulate the following vehicle’s acceleration as a linear function of the optimal velocity and the desired distance.

The first term on the Right Hand Side of (7) represents a control mechanism for train n to reach its optimal speed $V(\cdot)$, while the second term models a control mechanism for train n to keep to its desired separation s_n^{des} to the train in front. α and β are sensitivity parameters for the speed difference term and the space gap term respectively. A higher value of α (or β) means that the train movement control is more sensitive to the speed difference (or to the space difference), than with a lower parameter value.

We adopt also from Chen and Liu (2016) the following functions to model the optimal velocity as a function of the space gap:

$$V^*(s) = V_1 + V_2 \tanh(C_1 s - C_2) \quad (8)$$

and the desired following distance a function of the current running speed and a desired following time T :

$$s^{des}(t) = s_0 + T v(t) \quad (9)$$

where the constants in (8) are set as: $V_1 = 13$ m/s, $V_2 = 20$ m/s, $C_1 = 0.005$ m^{-1} and $C_2 = 0.1$. The parameter s_0 in (9) is the minimum space gap between trains and T represents a desired following time headway.

Equation (8) models the optimal velocity as a function of the space gap, while Equation (9) represents that the desired following gap varies with the current running speed of the train with a constant time headway. In the next sub-section, we analyse the stability property of this control mechanism for the ERTMS Level 3 system.

2.1.3 Stability analysis of the controlled train-following model

The controlled train-following model provides a mechanism for the MB ERTMS Level 3 system to control and give movement commands to trains following one another in the

same direction on a track. At any given point in time, if a train's speed and its space gap to the train in front in the previous time instance are known, the acceleration of the following trains can be calculated using Equations (7) to (9).

However, the choice of the parameter values used in these equations can have a significant impact on the performance of the control mechanism in terms of safety and the resulting capacities. In this section, we discuss the train-following parameter values and their effect on stable headways.

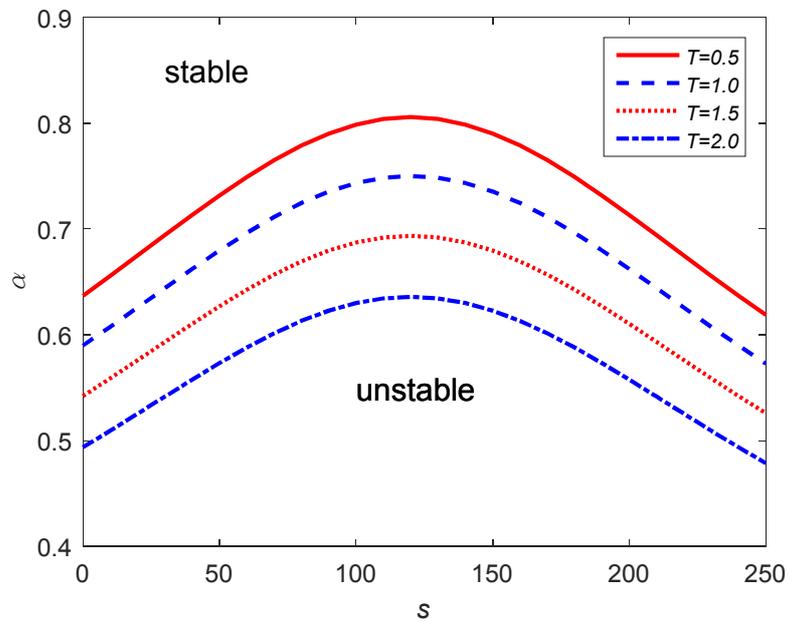
Chen and Liu (2016) provided the theoretical derivation of the linear stability condition of the model (7) – (9). We illustrate below the sensitivity of some of the model parameter values on the stability of the traffic flow systems. If we set the default parameter values as: $\beta = 0.05$, $V_1 = 13$ m/s, $V_2 = 20$ m/s, $C_1 = 0.005$ m⁻¹ and $C_2 = 0.1$, Figure 2.6 shows the neutral stability lines in the space (α, s) for different values of the desired following time heading T in equation (9) and the system response time t_d in equation (7).

The area above each of the stability lines in Figure 2.6 is the 'stable region': under those parameter values, the control mechanism will result in a stable train following at a safe following distance between two trains and smooth transition in speeds (with small acceleration and deceleration rates). The area below the stability lines represent the 'unstable region': controls with such parameter settings would result in trains over-responding with significantly large acceleration or deceleration rates, causing unstable traffic flows.

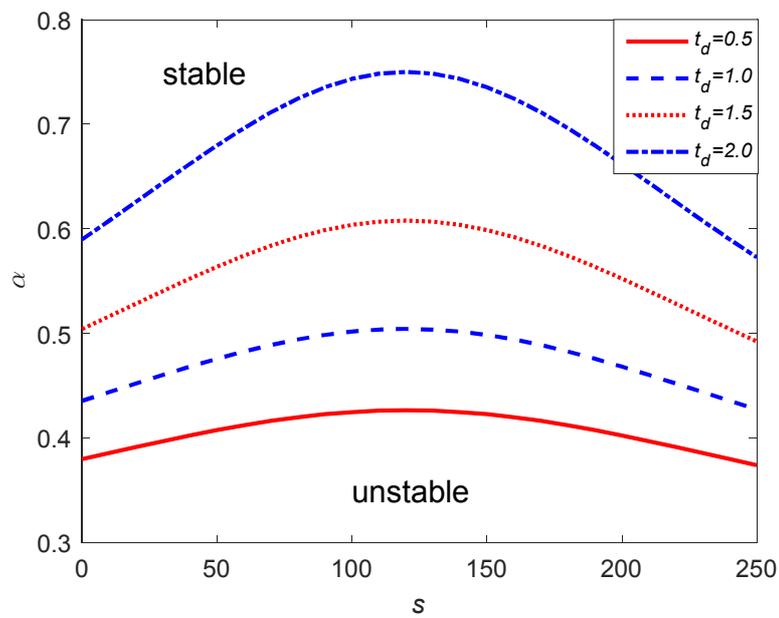
The results in Figure 2.6a show that the stable region increases with increasing desired following time gap. For example, with a desired following time gap $T = 0.5$ sec, a control based on a speed sensitivity parameter $\alpha > 0.8$ is required to yield a stable train following, whilst, with a longer following time gap, say $T = 2.0$ sec, most of the area above $\alpha > 0.6$ is stable.

The model parameter t_d represents a delay time for the system (train-driver unit) to respond (to an incident, a speed change, etc.). Figure 2.6b shows that the stable following regions are larger with smaller t_d values, implying that the traffic flow will be more stable when the system responds faster.

Figure 2.6: Stability conditions in the space of (α, s) for different values of: (a) safe following time T , and (b) response time t_d .

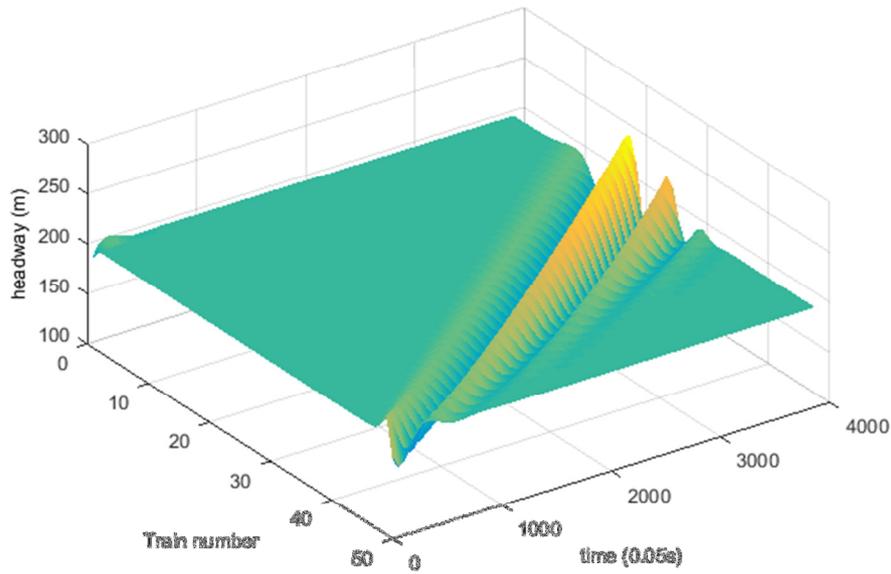


(a)

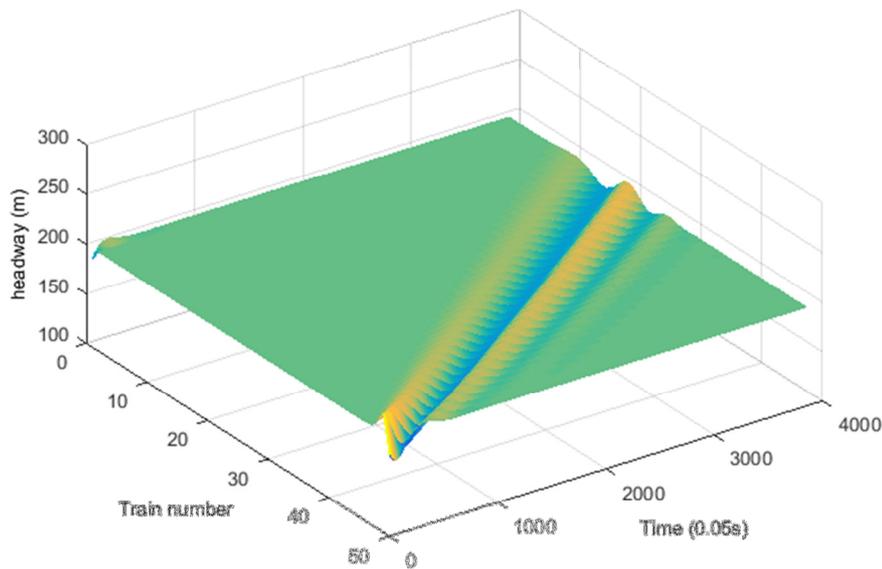


(b)

Figure 2.7: Space-time evolution of the following headways for: (a) $\beta = 0.05$, and (b) $\beta = 0.1$.



(a)



(b)

We perform a numerical simulation of the controlled following model (equations (7) to (9)) under a closed boundary condition. We consider 50 trains travelling on a single-track circular route of length 10 km. Initially, the trains are placed homogeneously on the track, with a uniform separation of 200m (including 100m train length) between

adjacent trains. A small disturbance was introduced to the system. We show below how the disturbance propagates through the stream of trains and how the system stability is affected by the choice of control parameter values.

Figure 2.7 shows how the headways between trains vary over trains and through time. In Figure 2.7a, we see a disturbance in headways started at around 50sec between train numbers 49 and 50. This disturbance propagates and is amplified through time and space (to other trains). The original homogeneous flow evolves into stop-and-go congestion during the period from 50 sec to 200 sec.

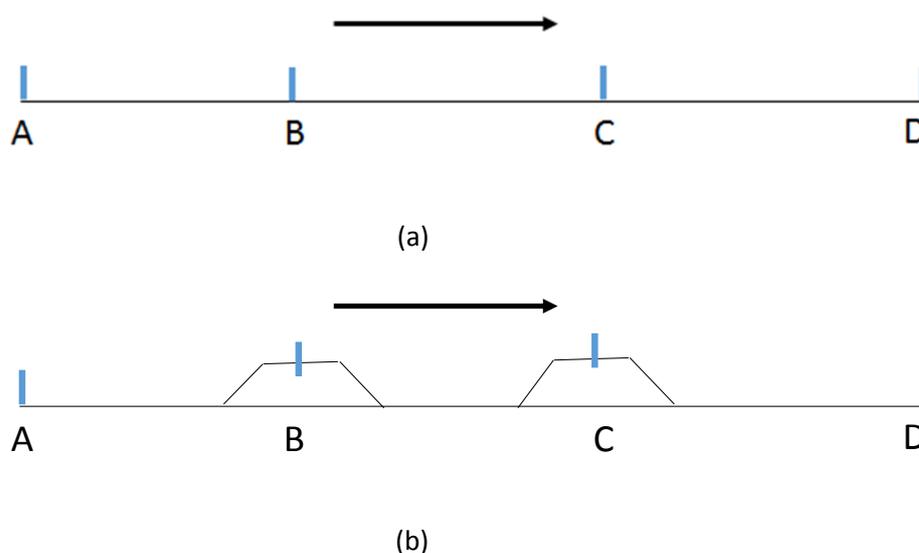
With a different choice of control parameter value $\beta = 0.1$, Figure 2.7b shows that the same initial disturbance can be better managed through the control mechanism.

The above results show the sensitivity of the control mechanism to the choice of model parameter values. The practical implication of which is that the train-following formulation and parameter values is critical to the success of the moving-block Level 3 control systems. Simulation models can be used to test theoretical control algorithms and evaluate their performances.

2.1.4 A TrackULA simulation test environment

The above train-following model was discretised and implemented in the TrackULA discrete-time simulation model. We apply this simulation model to evaluate the performance of a moving-block (MB) railway system under ERTMS Level 3 control. We set up simulation experiments to model the MB system on a single unidirectional track section with four stations as illustrated in Figure 2.8.

Figure 2.8: The test network for two different designs for the platforms at stations B and C: (a) platforms on the mainline track; and (b) platforms on passing loops off the mainline.



Trains enter at node A and exit at node D. Nodes B and C are station nodes where, depending on the schedule, trains may stop at one or both stations. The three sections of line are each 20km long.

Two network designs were set up. Figure 2.8a presents the same network set-up as used in Deliverable 3.1 whereby the platforms at stations B and C are on the mainline. When a train stops at such stations, no other trains can by pass them. Therefore no overtaking is possible on this network.

A new development has been made in TrackULA to represent platform loops on railway networks. The network in Figure 2.8b features platform loops at the stations B and C where the platforms for stopping trains are located. In this network layout, overtaking is then permitted at stations B and C. We then test the MB performance on these two different network designs.

We conduct two simulation tests on the performance of a MB system operating on a platoon of trains moving along this straight-line section of a network on level gradient. Firstly, we examine the congestion and delay propagation effects in a MB system. Secondly, we compare the performance of the two network designs of Figures 2.8a and 2.8b, without and with train overtakings respectively.

Congestion and delay propagation

In this test, we examine, when an incident occurs in the network of Figure 2.8a, how well our train speed-control algorithms respond to it and how far the delay propagates in the network in a MB system. We model 10 trains departing from station A with a scheduled headway of 4 minutes, with the first train departing at time 0. All trains are scheduled to stop at stations B and C for 1 minute each.

Two incident scenarios were set up:

Scenario I: an incident occurred at Station A which affected the departure times for trains number 2 and number 3. Instead of departing at 4 and 8 minutes as scheduled, their actual departures were at 9 and 10 minutes respectively. The incident was cleared quickly, trains number 4 onwards all departed on time.

Scenario II: an incident occurred on the track mid-way between Stations B and C. Train number 2 has to make an emergency stop for 2 minutes on the track.

The simulated outputs on the trains' space-time trajectories for Scenarios I and II are presented in Figures 2.9 and 2.10 respectively. Each line represents a train's trajectory.

We can see in Figure 2.9 that the delayed departures of train 2 and 3 began to show their impact on the system when train number 2 stopped for its scheduled 1-minute stop at station B, where the closely following train 3 caught up with it and had to wait outside station B for the platform/station to clear. The delay further propagates to later trains

number 4-6, though at a decreasing effect. Finally, by train number 7 the system appeared to be back on schedule.

Figure 2.9: Space-time trajectories in Scenario I. The two dashed lines are those of the two delayed departure trains number 2 and 3, while the solid lines represent the trains who departed on time.

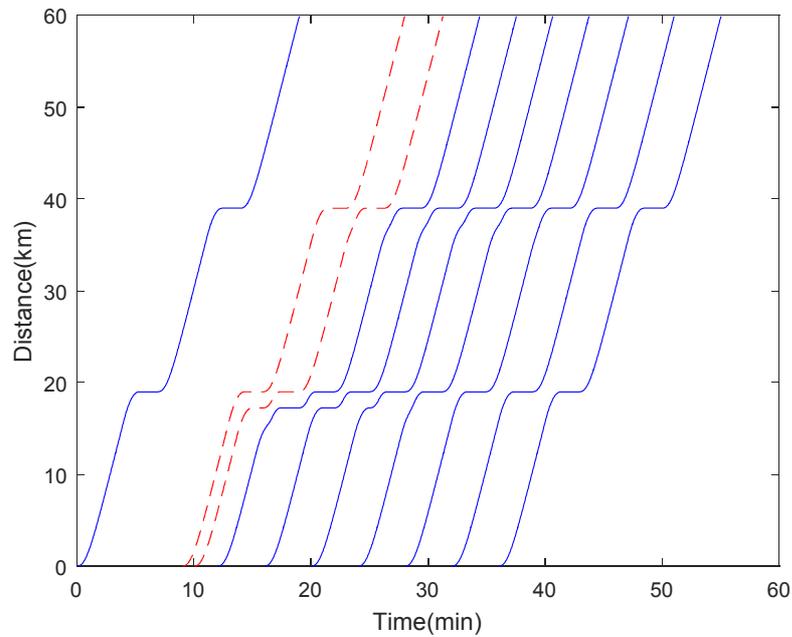
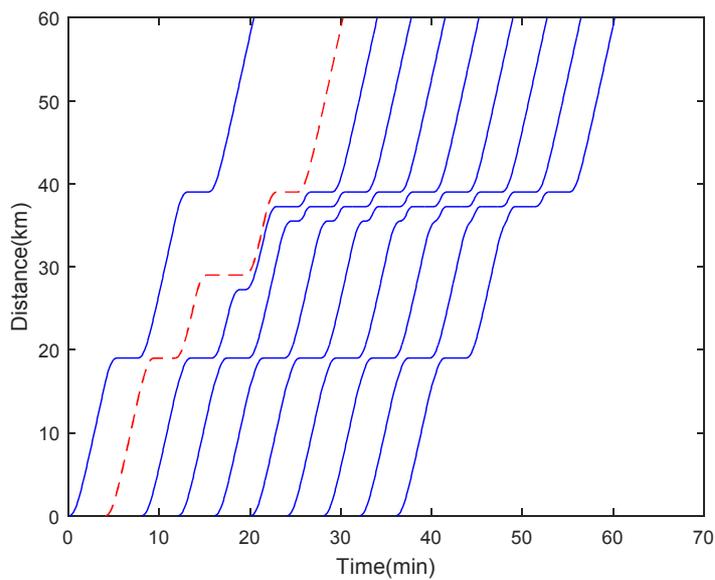


Figure 2.10: Space-time trajectories for Scenario II where train number 2 (dashed) had to make an emergency stop mid-way between stations B and C.



The emergency stop of train 2 had an immediate impact on the following train number 3 (Figure 2.10). Train number 3 had a stop-and-start on that section of the track and then followed train 2 closely while approaching station C. The major impact of the emergency stop was felt at station C, where the late arrival of trains 2 propagated to all the following eight trains.

The results show that the speed control algorithm for the ERTMS Level 3 MB control system is able to correctly capture the dynamic performance of the system and show congestion and delay propagation.

Network design - with and without platform loops for overtaking trains

In this test, we consider the effect of overtaking on network performance. The two configurations of the Figure 2.8 networks are modelled. In Figure 2.8a, once a train stops at the station B or C, trains behind also have to stop; whilst in configuration Figure 2.8b, fast trains which do not need to stop can overtake stopping or slower trains at the stations.

Two types of trains are modelled: a fast train and slow train. The fast trains have a maximum speed of 200 km/hr, are 250 metres long and with a 200 metres safety distance headway. This is the separation the trains need to keep when stationary and is a model parameter whose value can be changed to fit the purpose. The slow trains' maximum speed is 120 km/hr; they are 75 metres long and have a 100 metres safety distance headway.

A total of 16 trains are scheduled to traverse the line from A to D, with one fast train followed by one slow train to depart from A. The fast trains do not stop, while the slow trains are scheduled to stop at stations B and C for 1 minute each. The departure headway is 6 minutes for each of the two types of trains, with the first fast train departing at time 0 and the first slow train at time 1.5 minutes. Thus the departure times for all 16 trains are: {F0, S1.5, F6, S7.5, F12, S13.5, F18, S19.5, F24, S25.5, F30, S31.5} minutes, where letters F and S stands for fast and slow trains respectively.

Figure 2.11 shows the train trajectories under the two different network designs of Figure 2.8. We can see clearly that, except for the first fast train which traverses the entire network freely, all the other fast trains were obstructed by slow trains running in front of them. Figure 2.12 shows the close-following of train 3 (a fast train) with the slow train 2 ahead. The close-following headway between the two trains was around 350m when they were both travelling at the same speed.

Figure 2.13 shows the train trajectory with the network design of Figure 2.8b. We can see that, before the slow trains go to the siding to stop at station B, the fast trains following them firstly slow down on getting closer to the slow trains in front. Only when the slow trains have safely moved to the sidings do the fast trains behind then accelerate through the station.

Figure 2.11: Space-time trajectories of a group of 16 fast and slow trains traversing the network Figure 2.8a.

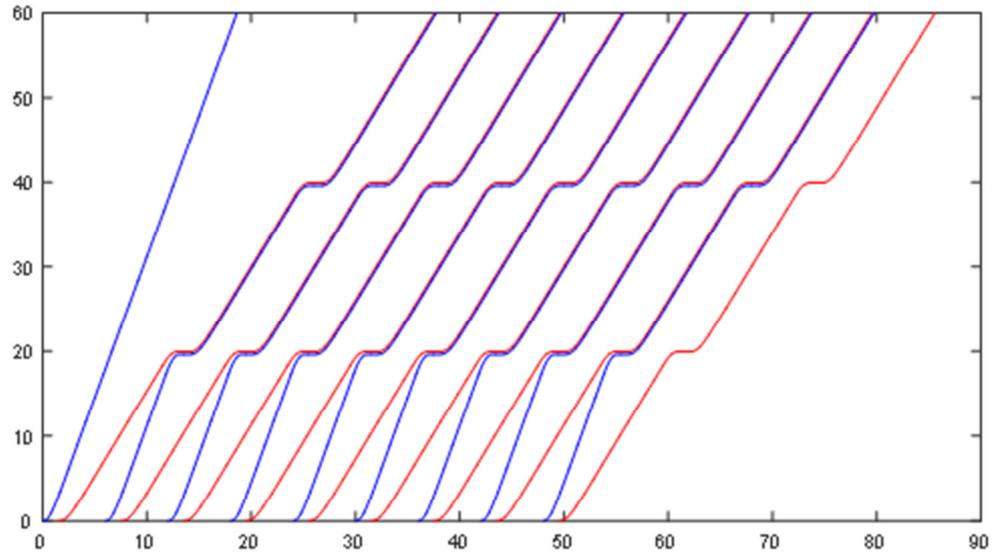


Figure 2.12: Relative speed and space distance between the slow train 2 and its following fast train 3, for test scenario based on network design of Figure 2.8a.

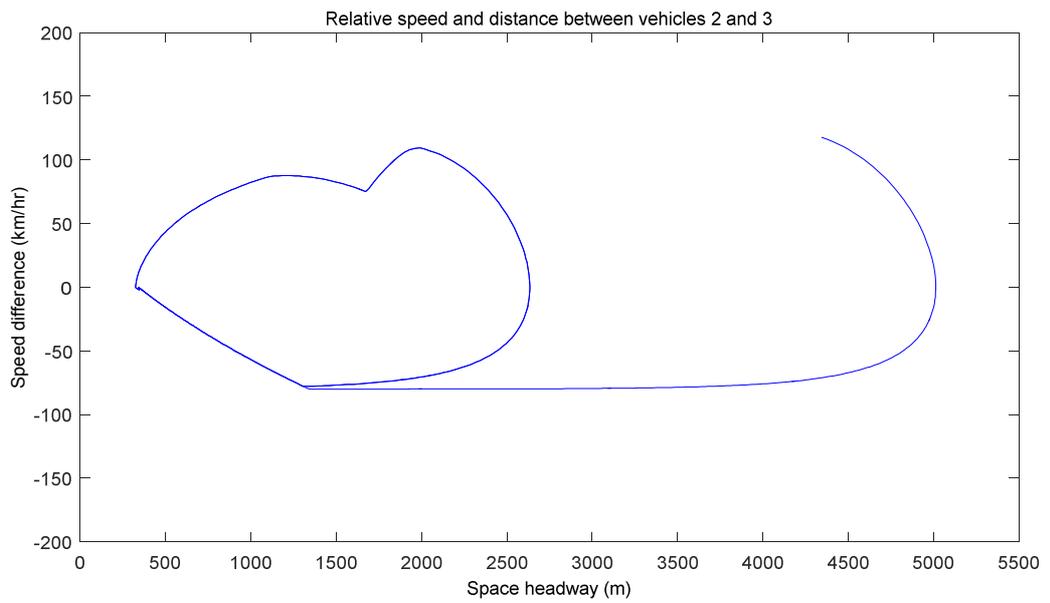
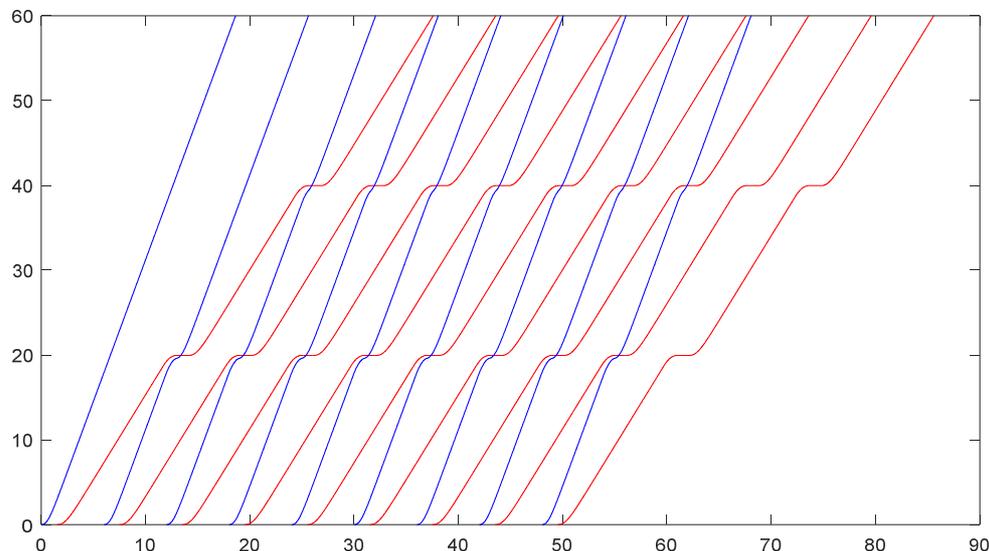


Figure 2.13: Space-time trajectories of a group of 16 fast and slow trains traversing the network Figure 2.8b.



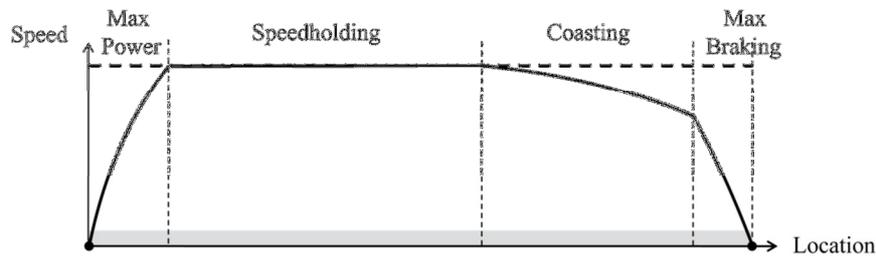
The results presented here show that the TrackULA simulation model captured the response of train driving behaviour with/without overtaking facilities, and demonstrate that the simulation model can be used to test both control-command algorithms as well as network designs.

2.1.5 A multi-train trajectory optimisation model

The classical train trajectory design problem is usually described as minimising the energy consumption of a train travelling freely from one station to the next within a given time period, without the obstruction of another train in front. The optimal control theory provides a powerful tool for looking at this problem from an analytical point of view, which reveals that the optimal operation strategy usually follows four control phases: maximum acceleration, speed holding, coasting, and maximum braking (Asnis et al., 1985), as is illustrated in Figure 2.14.

The four phases can then compose the optimal control strategy for very complex problems involving variable track gradient (including steep climbs and steep descents), variable speed limits, complex train characteristics, and power regeneration (Albrecht et al., 2015a, 2015b). The state-of-the-art review on both continuous and discrete control problems can be found in Howlett et al. (2009) and Albrecht et al. (2015a).

Figure 2.14: Energy-efficient train trajectory with level tracks and constant speed limit



The classical train trajectory optimisation problem deals with a single train running free of any constraint from another train in front. However, trains running close to each other on the same track section frequently interfere with each other. This is especially the case for ERTMS Level 3 as described in earlier sections; there, the speeds of the following trains should be optimised simultaneously.

In this sub-section, we present a multiphase optimal control formulation to optimise simultaneously multiple trains' trajectories in a railway network with heterogeneous trains and multiple stations.

The time order of all trains arriving at and leaving all stations and meeting points is assumed to be known. No prior information is required on the structure of the optimal train trajectories. The formulation incorporates various train running conditions such as location-related track gradients and speed limits as well as speed-related running resistances and maximum tractive/braking forces. Other train operation constraints can be considered, including the safety headways between consecutive trains, arrival/departure/dwell times at stations, inter-station running times, etc. The proposed framework is demonstrated by a case study which considers a fast train following and overtaking a slow train in the moving block system with the objective to save energy.

Multiphase optimal control formulation

In a multiphase optimal control problem, the whole evolution trajectory is divided into several "phases"; within each phase, the dynamic equations and constraints will not change. Two phases can be connected through linkage conditions represented by functions of the system variables of these two phases (Betts, 2001).

In our multi-train trajectories optimisation problem, we call each departure (D) and arrival (A) of each train at each node an "event". A train passing a node without a stop would be treated as an arrival event followed by a departure event. By placing all the events in a time-ascending order, we call the resultant sequence a "D/A-sequence". The evolution process between two adjacent events is called a "phase".

To fit the model into a standard multiphase optimal-control framework, we assume that the D/A-sequence is always given and conflict-free, but the exact timings of the events

need not be known in advance. The objective is then to obtain the train schedules and speed profiles that minimise the total energy consumption over the entire route.

Based on its locations at the starting time and terminal time of this phase, the status of a train at each phase can be categorised as in one of those listed in Table 2.1.

Table 2.1: Categorisation of train status in a phase

| Category | At the beginning of a phase | At the end of a phase |
|----------|--|----------------------------------|
| 1 | Departing from a node | Arriving at the next node |
| 2 | Departing from a node | Not yet arrived at the next node |
| 3 | Already departed from a node | Arriving at the next node |
| 4 | Already departed from a node | Not yet arrived at the next node |
| 5 | Waiting for departure at the origin (node 1) | |
| 6 | Stopping at the intermediate nodes (nodes 2 to $S - 1$) | |
| 7 | Stopping at the destination (node S) | |

We can establish the multiphase optimal control framework for the train trajectories optimisation. The objective is to minimise the total energy consumption of all trains throughout all phases. The constraints include:

- (1) the train running dynamics representing the relation between train location and train speed, and that between train speed and forces;
- (2) the constraints of speed limits, maximum tractive force and maximum braking force on train speed, tractive force and braking force, respectively;
- (3) the boundary conditions specified according to Table 2.1, which represent each train's initial and final locations and speeds in a phase;
- (4) the minimum headway requirement for safety reasons between two consecutive trains running on the same section;
- (5) the linkage for connecting two consecutive phases, which states the relationship of train locations and speeds at these two phases;
- (6) additional constraints which may be applied to each train for practical train operations, e.g., the time window for departure and/or arrival time at a station, the minimum and maximum stopping times at a station, the maximum running time between any two stations, etc.

A case study

In this section, we demonstrate the potential of the above proposed method in solving a practical problem. The case study involves two trains and three nodes, where nodes 1 and 3 are stations and node 2 is a meeting point. The locations of nodes 1, 2 and 3 are at 0km, 40km and 80km, respectively. The speed limit on the track is uniformly 180km/h, and the track gradients are given in Table 2.2. The two trains are identical in characteristics such as mass (6×10^5 kg), resistance, maximum tractive force and maximum braking force, shown as follows, where train speed v is in km/h, while maximum tractive force $\bar{F}(v)$, maximum braking force $\bar{B}(v)$ and resistance $R(v)$ are in kN. The safety distance is 2km for both trains. The original schedule is given in Table 2.3.

$$\bar{F}(v) = \begin{cases} 140, & 0 \leq v \leq 90 \\ 140 - 0.9(v - 90), & 90 < v \leq 180 \end{cases}$$

$$\bar{B}(v) = \begin{cases} 200, & 0 \leq v \leq 60 \\ 200 - 0.8(v - 60), & 60 < v \leq 180 \end{cases}$$

$$R(v) = 1.269 \times 10^{-3} v^2 + 0.101v + 11.4$$

Table 2.2: Track gradients

| Start location (m) | End location (m) | Gradient (%) |
|--------------------|------------------|--------------|
| 0 | 2284 | 0 |
| 2284 | 3871 | -4.2 |
| 3871 | 6094 | 2 |
| 6094 | 8181 | 16.7 |
| 8181 | 10323 | 0 |
| 10323 | 13750 | 1.7 |
| 13750 | 15711 | 0 |
| 15711 | 19314 | 13.3 |
| 19314 | 21222 | 0 |
| 21222 | 23609 | -5.6 |
| 23609 | 27504 | -8.3 |
| 27504 | 30127 | -3.3 |
| 30127 | 32778 | -8.3 |
| 32778 | 36003 | -2.5 |
| 36003 | 38091 | 0 |
| 38091 | 40000 | 0.5 |
| 40000 | 42284 | 0 |
| 42284 | 43871 | -4.2 |
| 43871 | 46094 | 2 |
| 46094 | 48181 | 16.7 |
| 48181 | 50323 | 0 |
| 50323 | 53750 | 1.7 |
| 53750 | 55711 | 0 |
| 55711 | 59314 | 13.3 |
| 59314 | 61222 | 0 |
| 61222 | 63609 | -5.6 |
| 63609 | 67504 | -8.3 |
| 67504 | 70127 | -3.3 |
| 70127 | 72778 | -8.3 |
| 72778 | 76003 | -2.5 |
| 76003 | 78091 | 0 |
| 78091 | 80000 | 0.5 |

Table 2.3: Original schedule

| Train | Departure time | Arrival time |
|-------|----------------|--------------|
| 1 | 08:00 | 08:35 |
| 2 | 08:05 | 08:40 |

Now assume that a malfunction happens on train 1 at 08:10, and its maximum tractive force function is altered to be

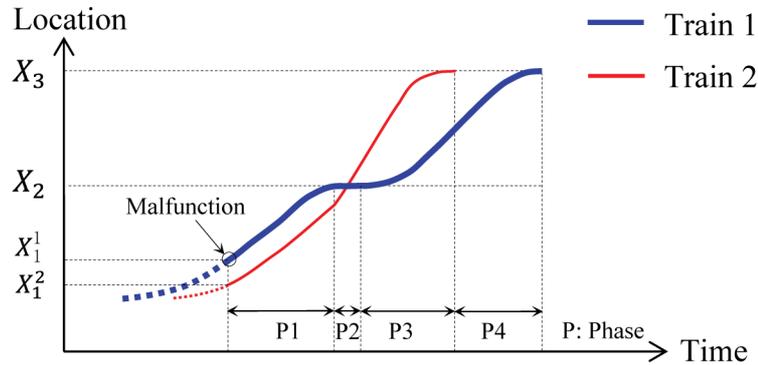
$$\hat{F}(v) = \begin{cases} 50, & 0 \leq v \leq 90 \\ 50 - 0.9(v - 90), & 90 < v \leq 180 \end{cases}$$

while its maximum braking force function is unchanged. At that moment, the location and speed of train 1 is $X_1' = 17626$ m and 28.4m/s, and that of train 2 is $X_2' = 7266$ m and 29.6m/s.

To make sure that train 2 arrives at node 3 on time, train 1 was asked to stop at node 2 to let train 2 overtake it without stopping, also the arrival time of train 2 at node 3 was modified to be 08:45. In such a case, the optimal trajectories have to be recalculated.

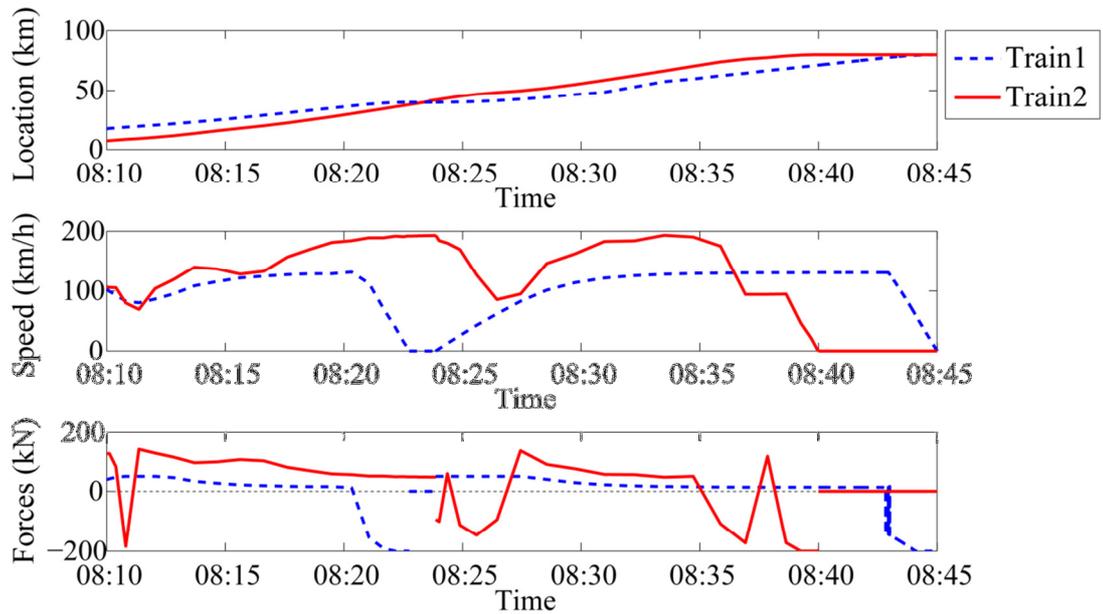
The whole process is divided into 4 phases, as illustrated in Figure 2.15. Phase 1 starts at the time the malfunction happens and ends when train 1 reaches node 2. Phase 2 corresponds to the time interval when train 1 stops at node 2. Phase 3 begins when train 1 departs at node 2 and ends when train 2 arrives at station 3. Then Phase 4 follows until train 1 arrives at station 3.

Figure 2.15: Illustration of Phases

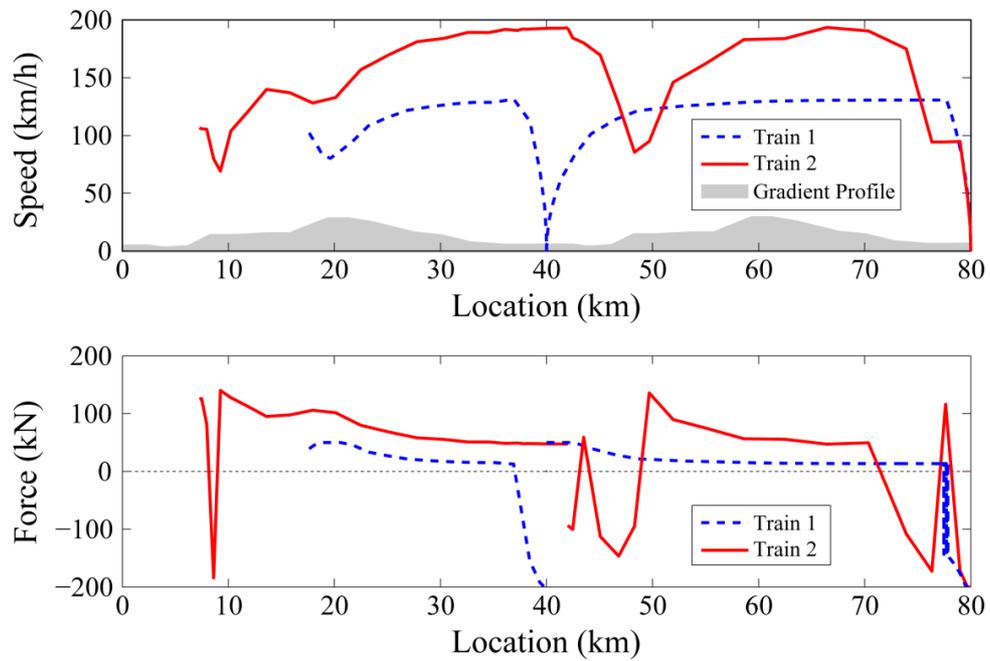


The results of the optimisation are shown in Figure 2.16a, with the track gradients overlaid on Figure 2.16b. We can see that the optimal speeds and forces of the two trains respond well with the gradients on the track: the steep upwards gradients tend to result in reduced speed.

Figure 2.16: Optimal train trajectories and operations



(a) Train locations, speeds and forces versus time



(b) Train speeds and forces versus train locations

2.1.6 Conclusions

A framework to optimise train trajectories has been developed for moving block systems (e.g. ERTMS level 3). This is based on a multiphase optimal control method with multiple trains simultaneously running on the same railway section. The processes of train scheduling and train trajectories optimisation are combined in the framework, and the train running simulation is naturally embedded by incorporating the train running dynamics in the optimal control framework. Future research will extend this to the simulation and optimisation of ERTMS level 2.

The related work on the dynamics of train-following has identified some important practical effects such as potentially unrealistic demands on braking/traction forces, incessant speed fluctuations and unstable traffic flow. Dynamic simulation models have been developed which are useful to test train control algorithms and design solutions of these problems.

2.2 Stochastic Optimisation

In the stochastic optimisation work package of DITTO, we apply operational research techniques to analyse the current utilisation of the rail network and the reliability of the train services offered. Utilisation is measured by the number of trains that travel through the network in a given time period - the more trains there are, the higher the utilisation. Reliability is measured by computing the deviations of the actual train times from the original train timetable due to unforeseen events, e.g., bad weather conditions, prolonged boarding and alighting, and technical malfunctions, which we refer to as uncertain events or uncertainty.

Our aim is to improve both reliability and utilisation. Yet, the two goals tend to be conflicting. Typically, we cannot increase the number of train services without causing reliability to deteriorate. We apply different optimisation techniques to iteratively generate timetables by increasing the number of trains while complying with the safety regulations. By examining the reliability of these timetables when executed under uncertain events, we can gain insights into the trade-off between network utilisation and timetable reliability.

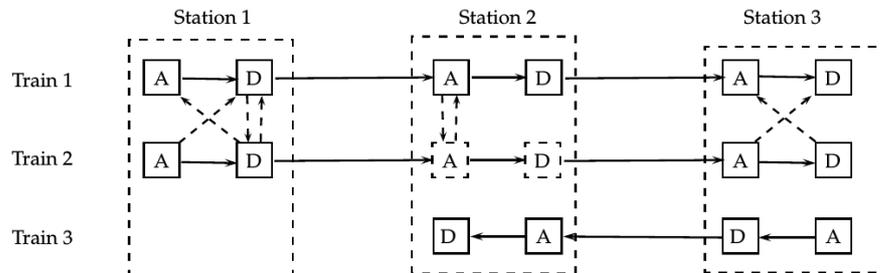
The problem is considered challenging and innovative as it combines optimisation under uncertainty and decision making with multiple objectives. The combination of these two areas has received little attention in the scientific literature. We propose a new mathematical model based on an event-activity graph. Solutions to small-scale instances are generated by commercial optimisation software called CPLEX. For solving large-scale instances, we develop a heuristic algorithm that generates near-optimal solutions in a short amount of computation time. In doing so, we build on earlier work presented as DITTO Deliverable 2.2/Milestone 5 (Bektas et al., 2015).

2.2.1 Model

Our problem is denoted by the train timetabling and scheduling problem (TTSP). The TTSP involves scheduling a set of trains on a network such that all operational constraints are satisfied. Trains might be subject to random delays. The goal is to generate a timetable such that the expected deviations from the arrival and departure times are minimised under such delays. We consider several recourse actions to adhere to the published arrival and departure times as much as possible (so as to avoid knock-on effects, for example), such as increasing train speeds, revisiting the order in which trains travel between stations, changing the train-to-platform assignment and the order in which trains approach a platform. The objective is to minimise the weighted average of the maximum delay that any train experiences and the average delay per train. Our TTSP belongs to the class of two-stage stochastic programs with recourse. In the first stage, we design a timetable by considering the delays after countermeasures have been taken in the second stage.

direction and ends at Station 2.

Figure 2.18: Example of an event-activity graph.



The purposes of different types of activities are described as follows:

Waiting activity: The dwelling of trains at stations is modelled by waiting activities between an arrival and a departure.

Travel activity: Train movements between different nodes, i.e., from a departure event to an arrival event are modelled by travelling activities.

Headway activity: Headway activities are incorporated to avoid conflicts between trains using the same track segment. In the example, Trains 1 and 2 use the same track segment when travelling from Station 1 to Station 2. Clearly, the trains cannot depart simultaneously. Therefore, we have to determine a precedence relation between these two trains: either Train 1 departs before Train 2 or the other way round. Headway activities come in pairs: for each two conflicting events i and j , we add two headway activities (i, j) and (j, i) to the set of activities. In our example, headway activities are represented by vertical dashed arcs. Generating a feasible solution involves selecting exactly one headway activity for each pair of conflicting events. The events of Train 2 at Station 2 are shown by dashed lines to indicate that the train passes through without stopping, i.e., arrival and departure take place at the same time. These dummy events are needed to prevent overtaking on single-track segments, e.g., if Train 1 leaves Station 1 before Train 2, then Train 1 should arrive before Train 2 at Station 2. The first-in-first-out property is modelled by additional headway activities at Station 2.

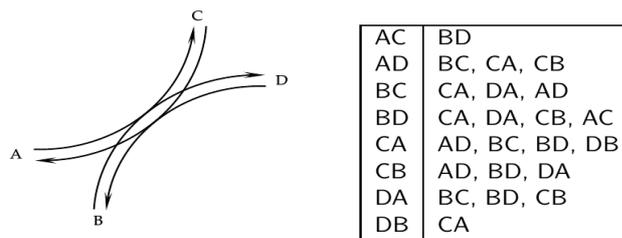
Spread activity: Spread activities have the same structure as headway activities, but they are defined at the stations and only for events belonging to trains on the same route serving the same sequence of stations. Their purpose is to maintain a sufficiently large time gap between trains travelling along the same route. For two trains on the same route, we would dispatch them with a time lag of, say, 15 minutes in between. The duration of a spread activity is based on economic considerations rather than operational or safety constraints.

Platform activity: For each pair of trains that might be assigned to the same platform, we define platform activities. Similar to the headway activities, there is a pair of mutually exclusive platform activities for each pair of conflicting events. Let i and j be

arrival events of two trains assigned to the same platform. Conflicts are resolved by selecting either activity (i, j) or activity (j, i) . Platform activities ensure that enough time has passed between one train leaving a platform and another one approaching it i.e., minimum platform reoccupation times are observed. In Figure 2.18, platform activities are illustrated between trains 1 and 2 at stations 1 and 3.

Junction activity: Platform activities can be generalized to avoid conflicts at junctions. Junctions are modelled as dummy stations where conflicting trains (i.e., trains that cross each other's track) are assigned to the same platform. This idea is illustrated in Figure 2.19. The diagram on the left shows an example with two double-track lines crossing each other. The table next to it demonstrates all potential conflicts between trains traversing different paths. Feasibility is enforced by adding a platform activity to the set of activities for each pair of conflicting trains. Conflicts between trains travelling from A to C and from B to C are avoided by headways activities.

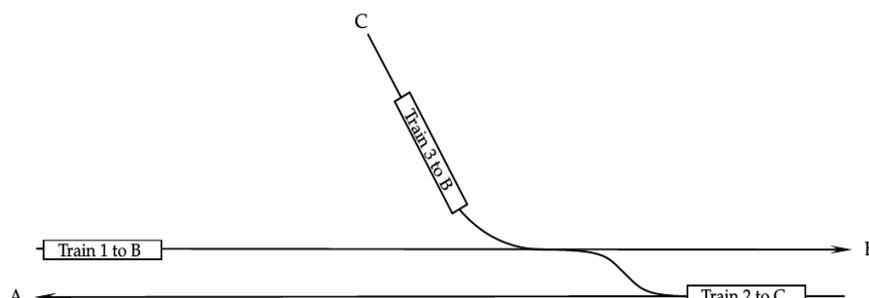
Figure 2.19: Potential conflicts at a junction of two double-track lines.



Bi-directional track activity: The concept of platform activities is also used to prevent conflicts at track segments that are traversed in both directions. In contrast to platform and junction activities, bi-directional track activities are defined between pairs of departure events (i.e., a train can depart only when the oncoming train has arrived at the next track segment).

No-wait activity: No-wait activities are similar to waiting activities, i.e., they connect arrival and departure nodes of the same train. However, they are defined only at points and junctions to make trains leave immediately. These activities are used to avoid overtaking at nodes without the required infrastructure. The reasoning behind no-wait activities is illustrated in Figure 2.20. Three trains have to pass a junction: Train 1 and 3 towards B, and Train 2 towards C. Feasible sequences are 1-3-2, 3-1-2, and 3-2-1. Without a no-wait constraint, a sequence 2-3-1 in which Train 3 arrives at the junction, disappears for the time between its arrival and departure, lets Train 2 pass through the junction, and then continues towards B would be possible, which is clearly not feasible in reality.

Figure 2.20: No-wait activities are incorporated to avoid sequence 2-3-1.



In addition to the described activities, we restrict the total journey time of each train. The reliability of a timetable depends on the planned journey time compared to the minimal journey time. Travel time buffers can be used to catch up on a delay. On the other hand, customer convenience decreases with longer travel times. By allowing the minimal journey time to be stretched by a given threshold, we enable the model to add buffer times where appropriate without slowing down trains excessively.

2.2.2 Solution approaches

The TTSP is a dynamic stochastic combinatorial optimisation problem that is notoriously hard to solve. Generating guaranteed optimal solutions is generally very difficult, particularly for large-scale instances. We follow two strategies to solve the TTSP. First, we apply a commercial black-box solver called CPLEX that is able to provide optimal solutions for small-scale instances. Second, we develop a bespoke heuristic algorithm that can solve large-scale instances in short amount of computation time.

Black-box solver (CPLEX)

Given a mathematical formulation of the problem (e.g. a mixed integer linear program), CPLEX implements a branch-and-cut algorithm to find a feasible solution.

In addition to a solution, CPLEX provides information about the quality of the solution. For small instances (i.e., small networks, few trains, and few scenarios) CPLEX generates optimal solutions in reasonable amount of time. However, the required computational resources grow exponentially with the size of the input.

For this reason, we cannot expect to solve realistic instances to optimality. However, optimal solutions to small instances can help to verify the efficiency and effectiveness of our heuristic algorithm.

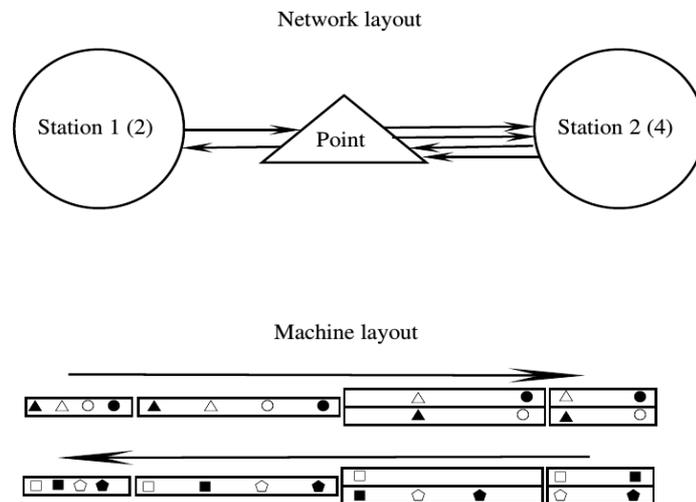
Heuristics

Heuristic algorithms search for a good solution among all feasible solutions according to a specific strategy. They might fail to discover an optimal solution. However, heuristics have been applied successfully to generate near-optimal solutions to many different types of optimisation problems that might be unsolvable with other techniques (assuming limited computational resources). Therefore, heuristics provide a valuable solution methodology.

The first step in solving the TTSP is to transform the event-activity model into a job-shop model that is used in production management. Job-shop models are well studied in the literature and, therefore, provide a solid basis for solving the TTSP. The challenge in the classic job-shop problem (JSP) is to assign a set of jobs to a set of machines such that the finishing time of the last job is minimised. Each job is composed of several operations. All operations of a job have to be executed on different machines in a specific order. Typically, at any time, each machine can process at most one job, and each job can be processed by at most one machine. A solution to the job-shop problem specifies the sequence in which jobs are processed on each machine and the respective starting times. The transformation from the TTSP to the JSP is achieved by replacing track segments with machines and trains with jobs. For each train, the route through a station, i.e. the sequence of track segments, is replaced by the set of operations. The TTSP is a generalization of the JSP: First, blocking and no-wait constraints force jobs to remain on the current machine until the next machine becomes available. Second, some machines can process several jobs simultaneously. Normally, it is not permitted for the same track section to be occupied by several trains simultaneously. In our model, however, a machine can represent several consecutive track sections, e.g. several trains can travel along the tracks between two stations as long as the headway constraints are satisfied. Third, operations have time windows in which their processing must start. Fourth, the cycle time of jobs, i.e., the difference between completion time and starting time, is bounded (see journey time constraints). Finally, particular operations have to be scheduled with a specific length of time to each other (see spread activities).

The transformation from the event-activity model to the JSP is illustrated in Figure 2.21.

Figure 2.21: Transformation from event activity-model into job-shop model.



The figure at the top shows an example of a network layout with two stations and one point. The figure at the bottom shows the layout in the job-shop representation. Rectangles denote machines; there is one for each platform at each station and each track segment. Train movements are denoted by triangles, circles, squares, and pentagons. Symbols of the same shape and same colour are different operations of the same job that are processed at different machines. In this example, there are four trains travelling from Station 1 to Station 2 and four trains travelling in the opposite direction.

The job-shop representation of the TTSP is solved by a metaheuristic called large neighbourhood search (LNS). The idea is to repeatedly destroy a feasible solution by removing jobs from machines in order to reassign them at better positions. The core of the algorithm is an insertion operator that recursively assigns operation to machines. Operations are assigned one after another. Each time when there is no feasible insertion position for a specific operation, we backtrack to the predecessor operation and assign it to a different position. The algorithm guarantees that a job is scheduled if there is a feasible insertion position. The job is left unassigned otherwise. For each iteration of the LNS, we destroy the current solution by removing jobs randomly.

The reliability of the generated solution is evaluated by fixing the sequence of operations and the operation-machine assignments, and adjusting the starting and ending times according to the given scenarios. This step involves solving an auxiliary

linear program. The output of the LNS is a solution to the job-shop problem that can be transformed into a timetable.

2.2.3 Trade-off between reliability and utilisation

We address the bi-objective nature of the problem by repeatedly solving the single objective problem with a fixed number of trains and record the reliability of the resulting timetable.

For each iteration, we add one more train to the timetable and solve the problem again. This process is continued until the current set of trains cannot be scheduled without violating operational constraints. Clearly, the number of trains that can be inserted into the system depends on the lines on which these trains travel (it is easier to insert trains on parts of the network with lower traffic).

If there is no information about the priority of the train services, we replicate trains from the initial set of services and add them to the new timetable. Let us assume that the initial set of services is grouped into different routes. The algorithm adds one train at a time to a line in a cyclic fashion until no train can be added feasibly (i.e., a full cycle has been performed without success).

The output of this approach is a set of timetables with different numbers of trains and different reliability values, respectively. By plotting the two measurements against each other, we can observe how reliability decreases with a higher utilisation of the network. Additionally, by taking a current timetable as input, we get an idea about the efficiency of the system. Based on this information we can evaluate alternatives for improvement (e.g., scheduling trains more efficiently, removing some trains in order to make room for trains with a higher demand, and adding buffers where required).

2.2.4 Experiments

In our computational experiments, we use different instances for different purposes.

The LNS algorithm is tested on various randomly generated test instances. By comparing the results of the LNS and the CPLEX solver, we are able to assess the robustness of the heuristic when applied to different instances (i.e., what is the best, average, and worst performance over all instances). Typically, we have no information about the quality of the solution generated by a heuristic. However, if the LNS performs well on the

benchmark instances, this provides good evidence that it is likely to generate reasonably good solutions on other instances as well. By solving various instances and comparing them to the optimal solutions (if available), we get an idea about the average performance. Finally, we can use the benchmark instances to adjust the parameters of the LNS. For example, we can test how the solution quality changes with a larger number of iterations (i.e. longer computation times). Once we have gained confidence in the quality of our heuristic, we can proceed to our case study with the expectation that meaningful conclusion can be drawn.

Random instances

Random instances are generated by a Python script. The input is the desired number of stations and the time frame in which trains should be scheduled. The algorithm first generates a network layout with the desired number of stations. Points and junctions are assigned randomly. Next, we identify all potential services that could run on this layout. Finally, the number of repetitions of each such service is chosen to be equal. The desired departure time from the origin is set randomly for each train, and the arrival and departure times at the following stations are calculated based on the length of the track segments and the average speed of the trains. Regular trains are assumed to travel at 80km/h on average, while express trains travel at 110km/h on average. Dwell times are either 3 minutes if a stop is scheduled or 0 minutes if the train passes through a station (e.g. in the case of express trains).

Delay scenarios are generated by Monte-Carlo sampling. For each train, the probability of being delayed is set to 50%. If a train is delayed, the length of the delay is sampled from a Gamma distribution. The delay is then designated either as a dwelling activity (with a probability of 0.1) or as a travel activity (with a probability of 0.9).

In total, we have generated 30 test instances with between 3 and 7 stations and with between 34 and 76 trains. With each instance, we associate 10 delay scenarios.

Case study

For the case study, we have chosen the rail network surrounding Peterborough station. The set of trains is selected from a representative weekday (4/11/2015) and is given in CIF-format. The total number of services in the file is 2,877². Delay information is

² Note only around 230 of these are southbound services between Retford and Hungtingdon that are used in later analysis.

gathered from historical delay data, recorded by Network Rail. More than 6 million delays were recorded between 1/12/2013 and 18/04/2015 (i.e., over 503 days). After filtering out trains in the London and North Eastern (LNE), East Midlands, and Anglia region, 1.1 million delays remain. Finally, after removing “not attributed” delays and cancelled, diverted, and short term trains, we are left with almost 800 thousand records.

The case study is again generated by a Python script. The input is a CIF-file, a station S , and a time horizon $[a,b]$. The algorithm filters all trains that:

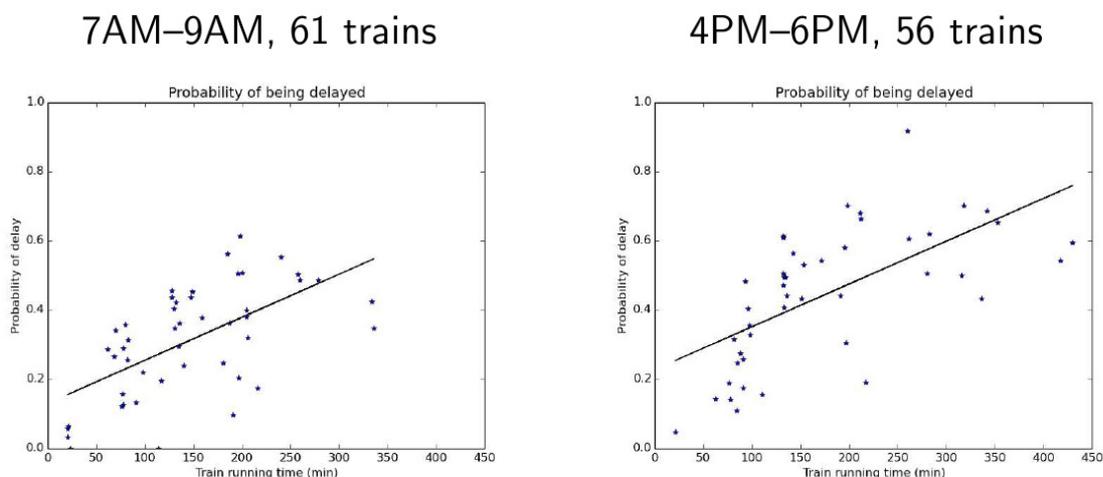
(i) visit station S during $[a,b]$; (ii) are run on a regular basis; and (iii) are either passenger trains or freight trains. Duplicates are removed. The output is a set of relevant trains. There are, for example, 61 trains running through Peterborough from 7am to 9am on a Wednesday.

The second step in our case study generator is to match filtered trains (T) with trains in the delay data (D). There is no unique identifier that unambiguously links trains in the two sources of data. Therefore, we apply the following strategy. Take the set of relevant trains from the previous step, the delay data, and a time margin TM . Match T with D if: (i) T is a passenger train (delays of freight trains are not recorded); (ii) T and D have the same origin and destination; and (iii) T departs within the departure time of $D \pm TM$. Duplicates are again removed. Additionally, for each train, we count the number of delays, the number of delays that occurred during a dwell period, and fit the parameters of a chosen delay distribution. The output is a set of delay scenarios that are sampled from the fitted distributions. In each scenario, we associate with each train a location and a duration.

The analysis of the data reveals differences in the probability of a train being delayed at different times of the day and with different total journey times. Figure 2.22 shows a linear regression analysis of the delay data from 7am to 9am on the left and from 4pm to 6pm on the right.

As expected, the risk of a delay increases with longer travel times. However, it seems that delays are propagated during the day, making delays in the afternoon more likely. An important implication is that the results of a case study performed in the morning are not representative for other times of the day.

Figure 2.22: Differences in the probability of being delayed.



Figures 2.23 and 2.25 show a histogram of the length of the delays of Train 0 and Train 6, respectively. At the top of each figure, besides the recorded data, we plot a normal, exponential, Gamma, and Weibull distribution with fitted parameters. The figure at the bottom shows an example of 500 random realisations of a random variable following a Weibull distribution.

The goodness of the fit of different probability distributions is examined in Figure 2.24 for Train 0 and in Figure 2.26 for Train 6, respectively, by using QQ-plots. These QQ-plots show the quantiles of the observed data against the quantiles of the respective probability distribution. The closer the quantiles of the observed data are to the quantiles of the probability distribution, the more accurate the model.

The analysis shows that Gamma distribution and Weibull provide the best fit. The Weibull distribution is more accurate when modelling the length of the delay of Train 0 whereas the Gamma distribution is more accurate for Train 6.

Figure 2.23: Delay distribution of Train 0.

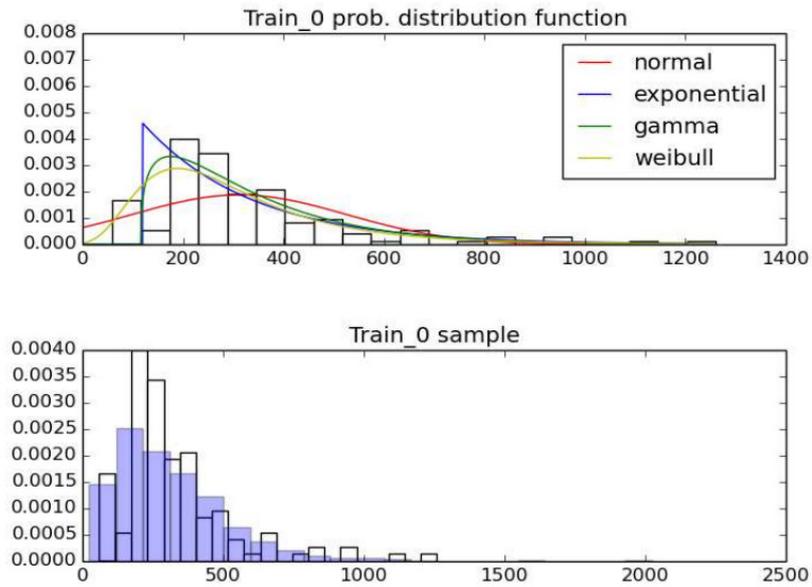


Figure 2.24: QQ-plot for different distributions for Train 0.

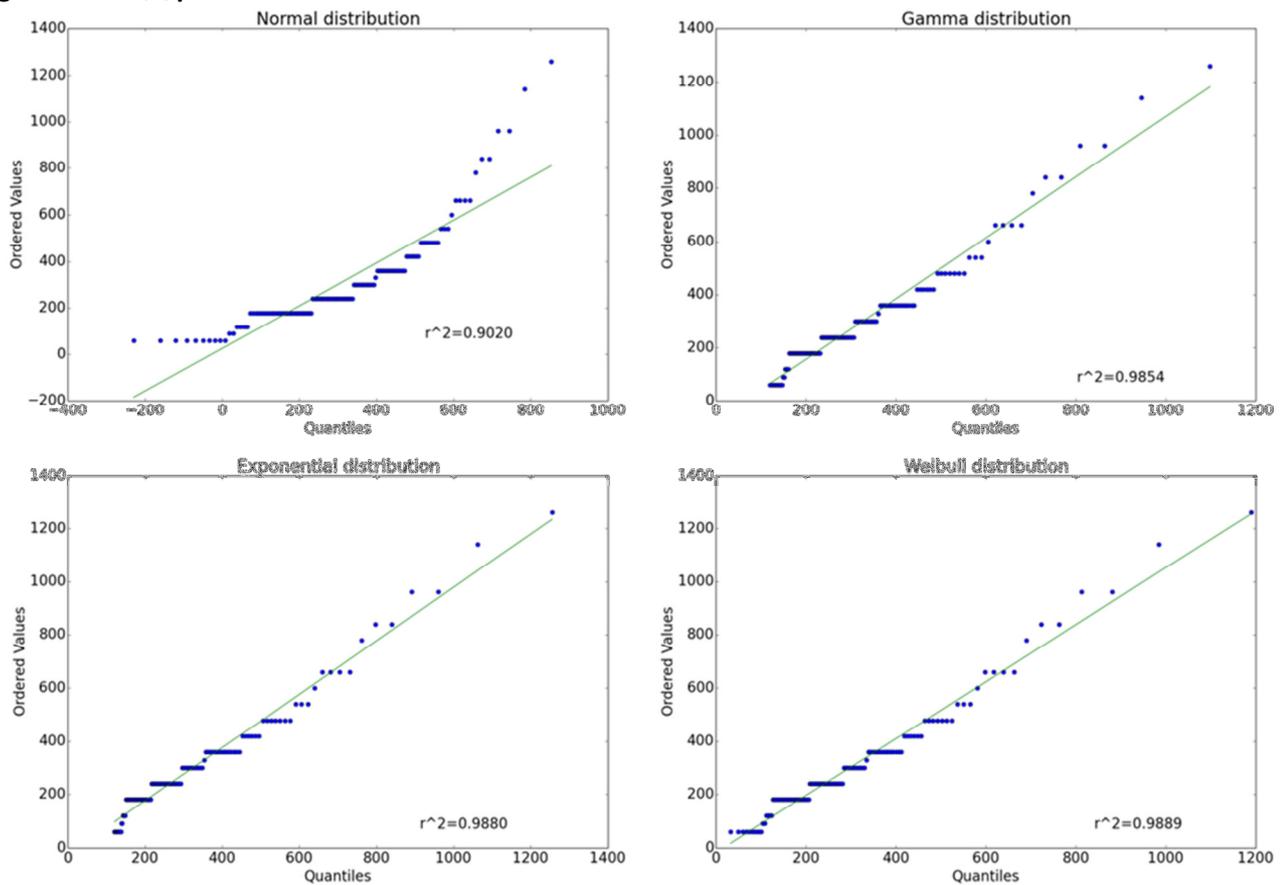


Figure 2.25: Delay distribution of Train 6.

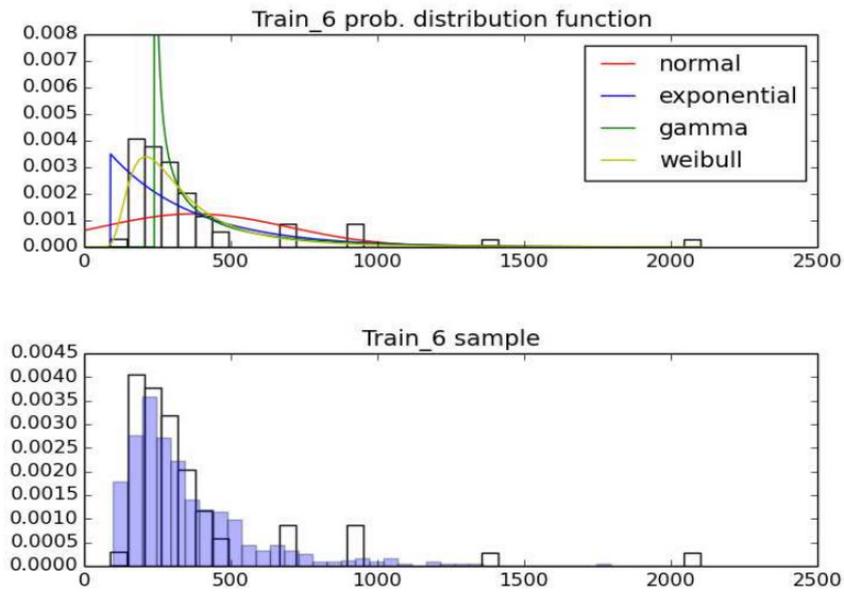
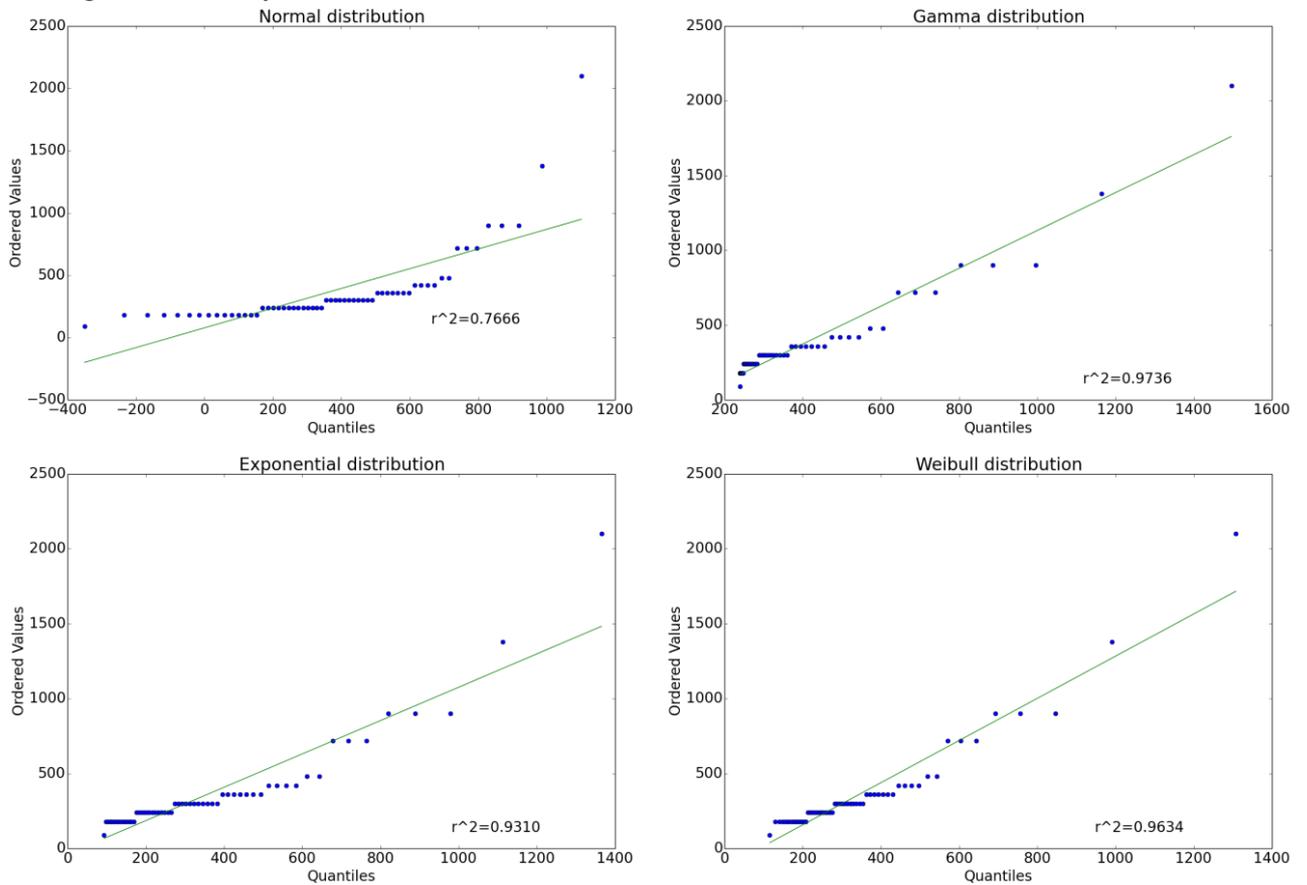


Figure 2.26: QQ-plot for different distributions for Train 6.



Results

All solution algorithms are coded in C++. Mixed integer linear programs (MIPs) and linear programs (LPs) are solved by IBM CPLEX 12.6. Experiments are run on the Iridis-4 supercomputer (i.e., Intel Xeon E5-2670 processors with 2.6 GHz and 4GB of RAM per core) at the University of Southampton.

As an example of an optimization run, the input is the network shown in Figure 2.27 and a set of trains. Trains are given in form of a timetable and illustrated on a time-space graph in Figure 2.28. There is one graph for each branch of the network. The vertical axis shows the nodes of the respective branch and the horizontal axis gives the time in minutes. The movement of express and regular trains is indicated by blue and red lines, respectively.

Figure 2.27: Example rail network

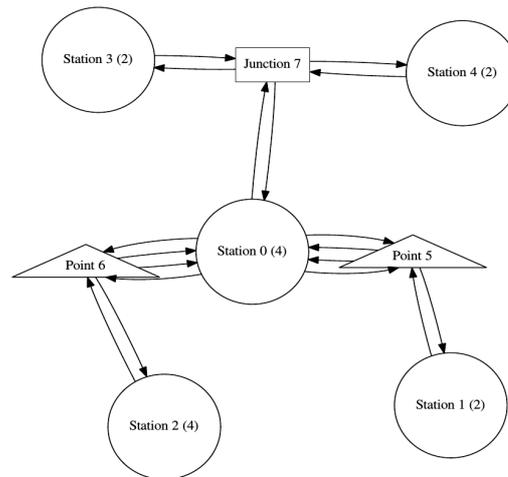
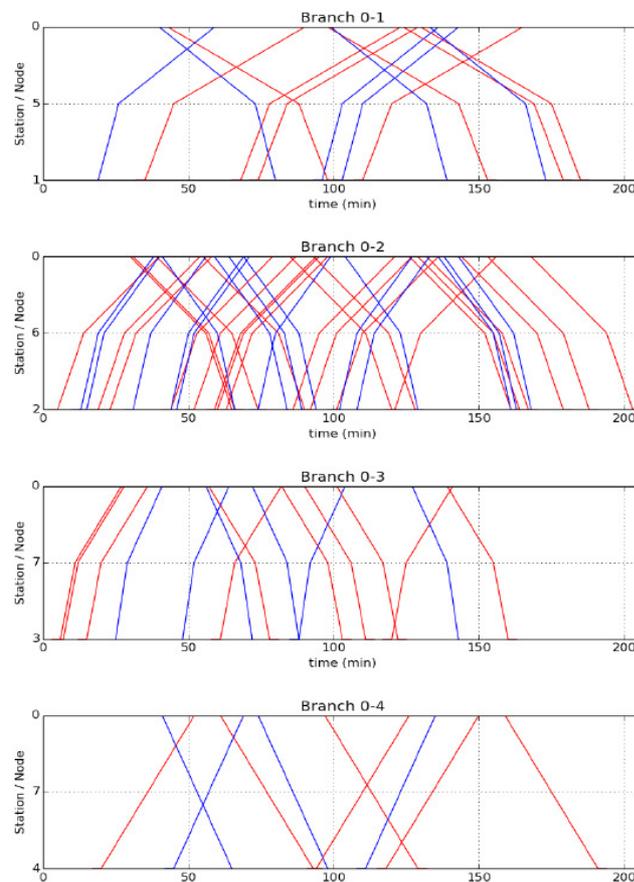


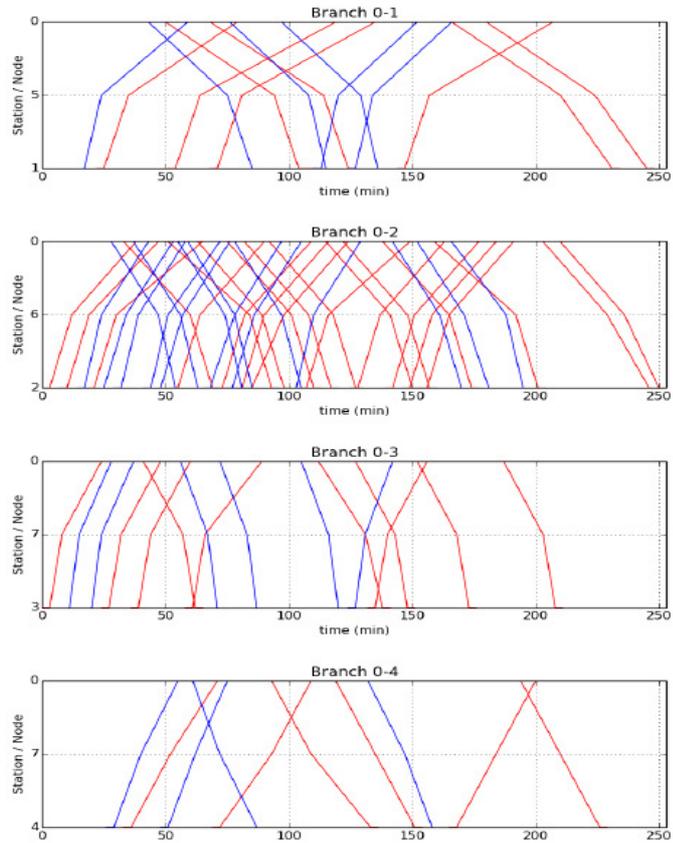
Figure 2.28: Set of trains to be scheduled, given in a timetable and illustrated on a time-space graph.



The input timetable does not have to comply with any constraint. The timetable is read and processed in order to build an event-activity model or a job-shop model. The output is a new timetable that i) complies with all operational constraints (e.g., headways, spread and stretch constraints) and ii) is robust against random delays. The solution is illustrated in Figure 2.29.

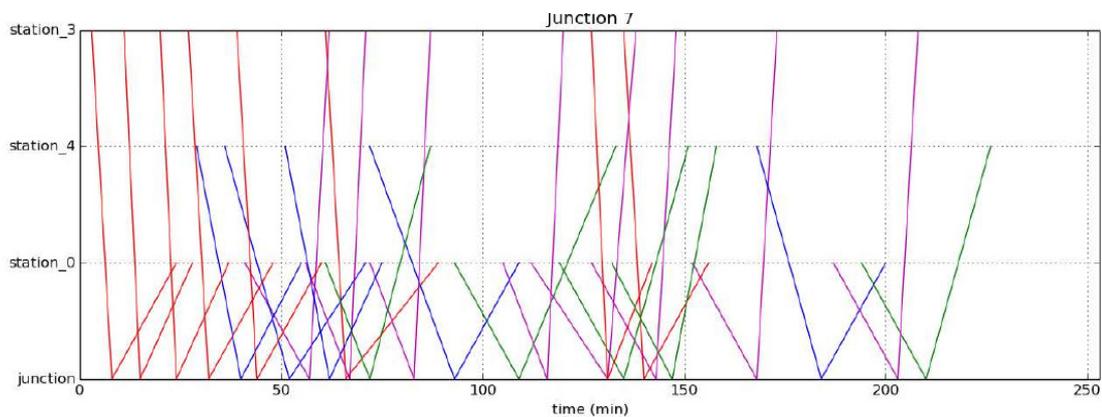
In this project, we are interested in the maximum number of trains that can run on a network. Therefore, we iteratively add new trains to the timetable until no feasible solution exists.

Figure 2.29: Solution timetable that satisfies all constraints.



Junctions require special attention as the risk of conflicts is higher than at other locations. Figure 2.30 shows how a sufficient time gap is kept between conflicting trains. Non-conflicting trains can pass the junction simultaneously. The graph illustrates all train movements that involve Junction 7, that is trains from Station 0 towards Station 3 (violet lines) and Station 4 (green lines), and trains towards Station 0 from Station 3 (red lines) and 4 (blue lines). Trains travelling between the same nodes (e.g., from 3 to 0 and from 0 to 3) are non-conflicting and can pass the junction simultaneously.

Figure 2.30: Junction in solution timetable.



Preliminary experiments on various benchmark instances show that generating exact optimal solutions to real world TTSP instances is beyond reach. However, the proposed heuristic provides high-quality solutions in significantly shorter computation times.

2.2.5 Conclusions and further work

In the stochastic optimisation part of the project, we propose a new mathematical formulation for the timetabling and scheduling problem under uncertainty. Random delays are taken into account by drawing random samples from a probability distribution fitted on the basis of historical data.

The problem is solved by a commercial solver and a bespoke heuristic algorithm. Results show that solving the problem to proven optimality is not possible with reasonable computational resources. The heuristic, however, generates good solutions in a relatively short amount of time.

The next step will be to finalize the experimental setting of the case study. This involves specifying the size of the study area, i.e., the network layout, and characteristic of the random scenarios. Following this, we will be able to perform a bi-objective analysis in order to examine the trade-off between reliability and network utilisation. The results will allow us to evaluate the current timetable of the case study, and explore the potential for adding additional services without creating significant delays.

3. Capacity Utilisation Indices Calculations

3.1. Introduction

Work Area (WA) 2.1 of the DITTO project entails the development of methods and tools to calculate capacity utilisation at railway nodes (i.e. junctions and stations), and the investigation and identification of the relationships between nodal capacity utilisation and service quality (i.e. punctuality and reliability), as reflected by levels of delay experienced by train services. In particular, WA2.1 seeks to identify suitable upper limits of capacity utilisation, within which trains can normally be operated at acceptable levels of service quality.

The methods, tools and limits developed and identified will be used in the subsequent Work Areas of the DITTO project, and will also be applied in the UIC-sponsored ACCVA (Assessment of Capacity Calculation Values) project and, more generally, in the wider railway industry.

This text follows on from Project Deliverable 2.1 (Milestone 4) (Armstrong and Preston, 2015), and reflects the findings of the earlier deliverable's literature review.

3.2. Methodology

As noted previously, the methodology adopted for WA2.1 is based upon and developed from the approaches used in the OCCASION project and Network Rail's Capacity Charge Recalibration (CCR) for Control Period 5 (CP5), undertaken by Arup with support from the University of Southampton and Imperial College London (Arup, 2013; Armstrong et al., 2015). Preliminary work with the data used for the CCR involved the examination of the relationship between performance and link-based CUI measures (Preston et al., 2015). The emphasis of the DITTO Rail Systems project is on the relationship with node-based CUI measures.

3.2.1 Data and network modelling

As described in Deliverable 2.1, the initial activities for this Work Area were based upon the timetable and Congestion-Related Reactionary Delay (CRRD) data used for the 2013 CCR. Additional, more general delay data were required to develop delay distributions and disruptive scenarios for the stochastic optimisation element of DITTO, and the use of more up-to-date data, as well as being more representative of the current operating situation, has several advantages: additional, related data are more readily available for 2015 than for 2013, the data are more reflective of current Timetable Planning Rules, and, in the context of the DITTO study area, the timetable and delay data reflect the recent changes that have been made to the layout of Peterborough station.

Additional delay data were therefore requested from Network Rail, who provided a link to their historic delay attribution repository (Network Rail, 2016a), for 2013-14 Period 9 to 2015-16 Period 1 inclusive (additional Periods have subsequently been added to the online dataset). The dataset included approximately 5 million records for the national network, with locations recorded in terms of STANOXes (Station Numbers, as used in TOPS, the Total Operations Processing System), rather than the TIPLOCs included in CIF timetable data (STANOXes are also inherently 'user-unfriendly', as they are numeric, unlike TIPLOCs, and thus provide no indication as to the network locations they represent). The STANOX values were therefore mapped onto their corresponding TIPLOCs, with records being 'cleansed' as necessary in cases where a single STANOX maps onto two or more TIPLOCs. Records with delays starting and/or ending in any of the LNE, East Midlands or Anglia regions were then extracted, to reflect the wider DITTO model area. The dataset was restricted to weekday delays, so records for Saturdays and Sundays were also excluded. The resulting dataset includes approximately 1.18 million records, and includes CRRD among the more general delay records, and so can be used for the investigation of capacity utilisation/performance relationships as well as the development of delay distributions and disruption scenarios.

For the initial development of the model of the East Coast Main Line (ECML) between Finsbury Park and Doncaster, including the Hertford Loop, it was decided to focus on trains running in the Up (i.e. southbound) direction between Retford and Huntingdon. An agreed representation of this route section was developed, as shown in Figure 3.1, including distances and station and junction names and TIPLOCs, where available, and the freight loops along the route section (Newark Flat Crossing was excluded from this initial representation for convenience).

Timetable data for the route were obtained from Network Rail's data feeds webpage (Network Rail, 2016b). The national CIF was filtered to include only trains running southbound through locations between Retford and Huntingdon inclusive on Wednesday 4th November 2015; this date was chosen as a representative, 'neutral' weekday, unaffected by holidays or other timetable variations, such as sometimes occur on Mondays and Fridays. The initial timetable filtering process retained some services running southbound through Peterborough from the Leicester direction towards Ely, and thus excluded from the initial model area; these trains were also removed from the dataset. Some of the remaining 232 passenger and freight train records required further editing during the detailed route assignment process, as described below.

In addition to the network diagram and timetable data, line speed and other information was obtained from the National Electronic Sectional Appendices (Network Rail, 2016c). Only very limited rolling stock information is available from the CIF timetable data, so further information was sought from Network Rail's online Working Timetables (Network Rail, 2016d). However, this is not a complete source of information on train formations (for example, numbers and sub-classes of multiple units are not necessarily included), and additional information has been obtained from Network Rail

in the form of TABS (Track Access Billing System) data, which records the details of the trains that are actually run on any given day (in this case, 4th November 2015). An additional request, for RailSys-based performance data, is pending.

Figure 3.1: Initial ECML Model Area

| Location (TIPLC) | Miles.Chains | Decimal Miles | km |
|---------------------------------|---------------|----------------|----------------|
| | 139.71 | 139.888 | 225.127 |
| Retford (RTFD) | 138.49 | 138.613 | 223.075 |
| | 138.23 | 138.288 | 222.552 |
| | 126.25 | 126.313 | 203.280 |
| Carlton Loop (CRLTOTL) | 125.53 | 125.663 | 202.234 |
| | 120.63 | 120.788 | 194.389 |
| Newark Crossing South Jn. | 120.51 | 120.638 | 194.147 |
| Newark (NEWANG) | 120.08 | 120.100 | 193.282 |
| Newark South Jn. | 119.73 | 119.913 | 192.980 |
| | 115.24 | 115.300 | 185.557 |
| Claypole Up Loop (CLPLLP) | 114.61 | 114.763 | 184.692 |
| Grantham (GTHM) | 105.38 | 105.475 | 169.746 |
| Grantham South Jn. (GTHMSJN) | 105.01 | 105.013 | 169.001 |
| | 101.46 | 101.575 | 163.469 |
| Highdyke Jn. (HGHDJN) | | | |
| Stoke Jn. (SOKEJN) | 99.60 | 99.750 | 160.532 |
| | 79.34 | 79.425 | 127.822 |
| Werrington Jn. (WRNGTNJ) | | | |
| New England North Jn. (NENGLNN) | 77.77 | 77.963 | 125.468 |
| New England Sidings | | | |
| | 76.57 | 76.713 | 123.457 |
| | 76.46 | 76.575 | 123.236 |
| Peterborough (PBRO) | 76.29 | 76.363 | 122.894 |
| | 76.10 | 76.125 | 122.511 |
| | 75.11 | 75.138 | 120.922 |
| Fletton Jn. (FLETTON) | | | |
| Connington South Jn. (CNNGSJN) | 67.20 | 67.250 | 108.228 |
| Up Slow Loop | | | |
| Woodwalton Jn. | | | |
| | 59.20 | 59.250 | 95.354 |
| Huntingdon North Jn. (HNTNNJN) | | | |
| Huntingdon (HNTNGDN) | 58.70 | 58.875 | 94.750 |

it and travel via nodes excluded from the model before re-joining it (examples include trains travelling from Newark North Gate to Peterborough via Spalding, rather than directly). Another difficulty encountered is the specification in the CIF data of passing times at TIPLOCs which include multiple tracks, and where the track used is not explicitly stated: examples include the crossovers at Tallington (TALNGTN) and New England North (NENGLNN). The intended routeings tend to be intuitively obvious, but considerably more difficult to ascertain algorithmically, in the course of which the 'search spaces' can become very large (this issue was encountered and addressed in the Knaresborough model, in the cases of minor two-platform stations, but the routing options were very limited in practice). At New England North, the fundamental choice tends to be between routing along the Up Fast or Up Slow main lines or into the freight yards via the Goods Line: sometimes the track to be used is stated explicitly, but in other cases it is not. In the short term, the choice can be resolved by adding the details of the track to be used to the CIF data; in the longer run, and to improve the robustness and 'generality' of the tool, it is preferable to amend the route search strategy to avoid these problems (for example, by looking beyond the 'current TIPLOC' to the next destination, as has been done previously, but in a different context, for the import of CIF data to EMME-based public transport models).

This routing issue is a specific example of the wider challenge of identifying shortest paths through only relatively complex networks which nonetheless include large numbers of potential intermediate routes, even when the target nodes are explicitly identified. Steps have already been taken to limit the extent of the search space, truncating infeasible branches of the 'search tree' as early as possible in the search process, and further options for restricting the search space will be investigated, to reduce the time required to complete the process and facilitate its incorporation in the stochastic optimisation; this may require a trade-off with the complexity of the description of the network, but it is likely to be one worth making.

3.3. Ongoing and Further Work

The detailed routeings of all 232 modelled trains through the study area have now been established, and the determination of hourly and three-hourly nodal and link-based Capacity Utilisation Indices (CUIs) is underway. The CRRD data will then be extracted from the updated delay dataset, and assigned to the corresponding nodes, links and time periods, and the relationships between CUI and CRRD will be further investigated, building upon the previous findings for Knaresborough, and focussing particularly upon the nodes at Retford, Newark, Grantham, Peterborough, Huntingdon and their intermediate junctions.

The analysis will then be extended beyond the initial study area to further nodes, initially in the LNE route area, and then beyond. The initial focus will be on the nodes comprising the agreed wider study area: there are several 2-track through stations on

the Hertford Loop and at Welwyn North, and mostly 4-platform, 4-track and 2-platform, 4-track stations (with some variations) between Alexandra Palace and Huntingdon.

For the larger, more complex nodes, including Peterborough as necessary, a ‘horses for courses’ approach will be taken, as previously proposed, making use of the updated UIC 406 and other techniques as appropriate. As well as providing project outputs, this will enable a useful comparison of the results obtained from different methods, of particular relevance to the ACCVA project, but also of wider interest.

Simulation techniques, employing RailSys, will also be used to further investigate and validate the reliable upper limits of capacity utilisation for nodes, as indicated by the analytical work based on empirical data.

The capacity utilisation calculation methodology may be implemented in C++ for integration with the stochastic optimisation process.

3.4. Conclusions

Updated delay and timetable datasets have been obtained from Network Rail to meet the modelling requirements of the DITTO Rail Systems project, and these have been filtered to exclude records that are not applicable to the modelled area and timetable.

The ECML route section to be used for initial model development has been agreed and defined, and additional data have been obtained and requested to provide a full set of operational information.

For the purposes of the CUI calculations, the infrastructure model has been defined, and the tools developed for the previous Knaresborough-based analysis have been developed further to accommodate the comparative complexities of the initial ECML model area. The timetabled services for the selected day of operation have been successfully routed through the detailed network. Work is now continuing to produce the capacity utilisation outputs and map the updated delay data onto the nodes and links of the model area, and explore their relationships further.

4. Prototype Development

The development of railway systems is usually supported by a range of tools, each addressing individual, but overlapping concerns such as, e.g., performance or safety analysis. However, it is a challenge for users to organise work-flows; results are often in different, non-aligning data formats; furthermore, tools work on different levels of abstraction from macro to microscopic. Thus, tool integration would be beneficial, and also allow for more playful, experimental prototyping and design. However, such an endeavour needs close collaboration. How to use the various tools and software needs to be demonstrated and documented, especially the relation between the rail system and its model, how this model is encoded into an input format, what commands are used for the analysis, and how the expected result shall look like.

All project partners within DITTO use models of rail systems. These models are designed to serve different purposes and therefore look at these systems from different perspectives. However, what they have in common is that they aim to describe, potentially, the very same rail system. Furthermore, it is understood that computer support is necessary due to the inherent complexity of these models. One of the objectives of the Swansea project partner is to investigate how to establish an integrated tool chain for analysing and optimising a rail system through the modelling of the three partners within the DITTO consortium.

As part of the DITTO goal towards integration, this section reports on lessons learned from the integration of BRaVE – the Birmingham Railway Virtual Environment, see <http://www.Bravesim.org> - and OnTrack, see Ditto Deliverable Report 1.1/Milestone 1 (James et al., 2015a), from Swansea University. These tools are both being developed as part of the wider FuTRO project. We present first steps towards an approach that bridges the gap that occurs from varying details in data sources through automated transformations. This integration provides a seamless environment for prototyping, concept development, and safety analysis under "one roof". We demonstrate the usefulness of our approach by giving integrated simulation and verification results for elements of the ECML model area.

4.1 BRaVE and OnTrack

BRaVE is an easy-to-use railway simulation software package for development, modelling and flow analysis. It is developed by the Birmingham Centre for Railway research and forms a core part of their research and teaching environment. Modelling the operation of train services with all of the complexities of signalling, interlocking, timetabling and train performance is a challenge. BRaVE simplifies the process, using an intuitive graphical interface to design and develop models of routes, signalling systems and timetables. BRaVE is currently being used to validate signalling design and simulate the Communication Based Train Control system for the new Hefei metro in China. A full

account of the features of BRaVE is available at <http://www.Bravesim.org>. BRaVE is written in Java and uses a proprietary format for storing data models. However, it does provide a number of useful data export features that will form the basis of the integration we present.

OnTrack is a toolset developed by Swansea University. It provides features that support the automatic verification of scheme plans against properties via different formal methods. Based on a provided CAD track plan and the associated control tables, OnTrack can be used to draw the Scheme Plan using the OnTrack graphical front end. These scheme plans are models formulated relative to OnTrack's DSL meta-model. A scheme plan is then the basis for subsequent workflows that support its verification, simulation, analysis etc. Scheme plans can be translated to formal specifications in various (specification) formalisms. Once a formal model has been generated, it can then be simulated or verified using the tools associated with the formal specification language that has been used for generation. In Deliverable 2.1/Milestone 2, we have shown how OnTrack will provide the basic prototype for interfacing, with the RailML format and with the OCCASION and Trackula models developed by the Universities of Southampton and Leeds respectively (James et al., 2015b).

4.1.1 Data Sets: East Coast Main Line

The ECML has been selected as the area of interest for a common case study within the DITTO project. In particular, the stretch of railway running from Huntingdon to Retford. This region of railway has already been encoded by the Birmingham Railway centre within the BRaVE tool. The data captured within BRaVE is based on accurate information from Network Rail. As a first exemplar towards integration, the remainder of the section is based on data that has been exported from the BRaVE toolset and then transferred into OnTrack. The data originally comes from simulation models provided to Birmingham Rail Centre by Network Rail. As OnTrack focuses on verification of particular rail nodes, two particular rail nodes from the East Coast main line form the basis of our discussion. In particular we have considered the line between Barkston South Junction and Werrington Junction (Figure 4.1), which is located on the ECML from just north of Grantham and to just north of Peterborough.

4.2 A First Integration

The starting point for integration is to consider the data that is required by both BRaVE and OnTrack. Here we discuss data that is used by BRaVE that is required for a successful integration with OnTrack. An example of this data is given in Figure 4.2.

Figure 4.1: Barkston South Junction Werrington Junction as displayed in BRaVE

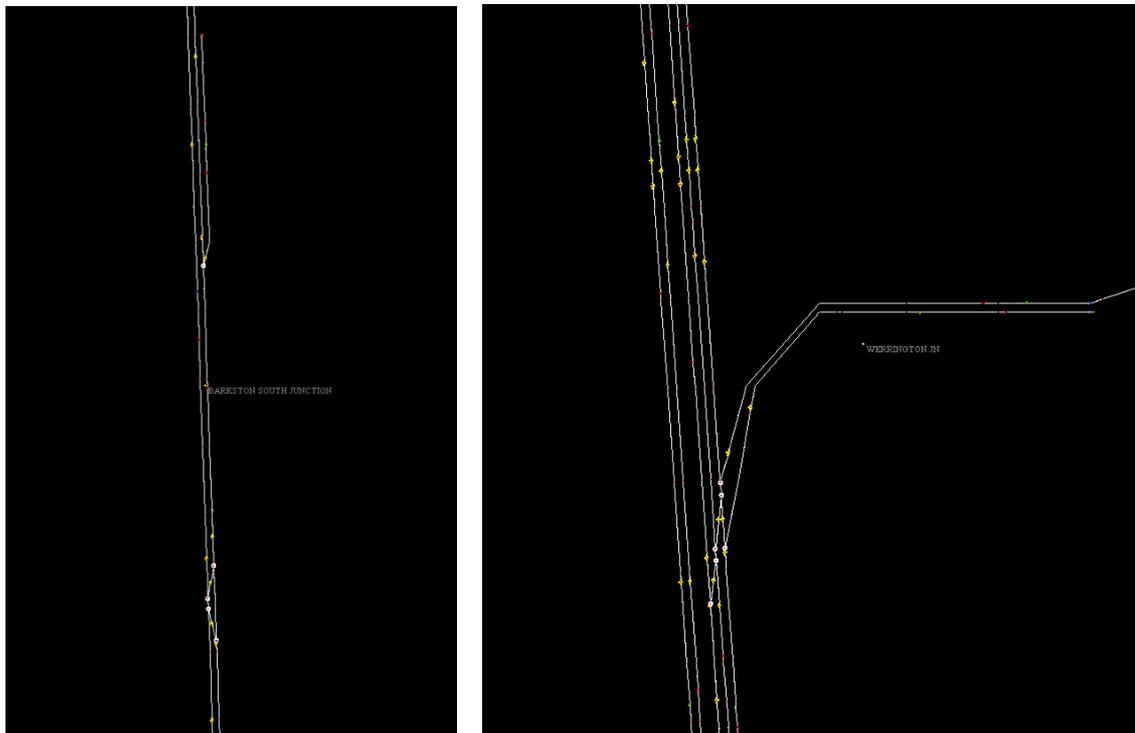


Figure 4.2: Examples of BRaVE and OnTrack Data

| | | | | | | | | |
|-------|------------|-----------|-------|-------|--------------|--------------|-------|--|
| N1231 | -169292.39 | 3756402 | P1240 | P1325 | SIG : AL3434 | SIG : AL3434 | 0.901 | |
| N1232 | -168556.67 | 3757064.5 | P1326 | P4955 | 1000MM | 1000MM | 1 | |
| N1233 | -173119.65 | 3753225 | P1295 | P1323 | SD : AL3432 | SD : AL3432 | 0.404 | |
| N1234 | -172599.45 | 3753693.4 | P1242 | P1324 | SIG : AL3432 | SIG : AL3432 | 0.474 | |
| N1235 | -171982.63 | 3754248.8 | P1243 | P1244 | GRAD | GRAD | 0.557 | |
| N1236 | -171261.78 | 3754897.8 | P1244 | P1246 | OL AL3432 | OL AL3432 | 0.654 | |
| N1237 | -169470.81 | 3756510.4 | P1245 | P1322 | SD : AL3433 | SD : AL3433 | 0.895 | |

```

<hasUnits xsi:type="bjoernercomplete:Linear" hasC1="/@hasConnectors.14" hasC2="/@hasConnectors.15" name="BV"/>
<hasUnits xsi:type="bjoernercomplete:Linear" hasC1="/@hasConnectors.15" hasC2="/@hasConnectors.16" name="BU" hasS
<hasUnits xsi:type="bjoernercomplete:Linear" hasC1="/@hasConnectors.16" hasC2="/@hasConnectors.17" name="Entry2"
<hasUnits xsi:type="bjoernercomplete:Linear" hasC1="/@hasConnectors.18" hasC2="/@hasConnectors.19" name="CM" hasS
<hasUnits xsi:type="bjoernercomplete:Linear" hasC1="/@hasConnectors.19" hasC2="/@hasConnectors.20" name="CL"/>
<hasUnits xsi:type="bjoernercomplete:Linear" hasC1="/@hasConnectors.20" hasC2="/@hasConnectors.21" name="Exit3"/>
<hasUnits xsi:type="bjoernercomplete:Linear" hasC1="/@hasConnectors.22" hasC2="/@hasConnectors.23" name="DR"/>
<hasUnits xsi:type="bjoernercomplete:Linear" hasC1="/@hasConnectors.23" hasC2="/@hasConnectors.24" name="DP" hasS
<hasUnits xsi:type="bjoernercomplete:Linear" hasC1="/@hasConnectors.24" hasC2="/@hasConnectors.25" name="Entry3"
<hasUnits xsi:type="bjoernercomplete:Linear" hasC1="/@hasConnectors.26" hasC2="/@hasConnectors.8" name="Exit2"/>
<hasConnectors hasUnit1="/@hasUnits.0"/>
<hasConnectors hasUnit1="/@hasUnits.1" hasUnit2="/@hasUnits.0"/>
<hasConnectors hasUnit1="/@hasUnits.2" hasUnit2="/@hasUnits.1"/>
<hasConnectors hasUnit1="/@hasUnits.3" hasUnit2="/@hasUnits.2"/>
<hasConnectors hasUnit1="/@hasUnits.4" hasUnit2="/@hasUnits.3"/>
<hasConnectors hasUnit1="/@hasUnits.5" hasUnit2="/@hasUnits.4"/>
<hasConnectors hasUnit1="/@hasUnits.6" hasUnit2="/@hasUnits.5"/>

```

As part of the East Coast main line model, the BRaVE dataset includes the following information:

- Nodes: These are the basic entities that are used to indicate points of interest along a railway line. For example, nodes may indicate the position of a point or a signal, a speed change, a gradient change etc.

- Paths: Paths are used to connect nodes. A path also has particular properties associated to it, including a speed restriction for that section of railway along with gradient information.
- Track Circuits: Track circuits are used to indicate where the physical track circuits (i.e. physical detection units) are placed along a railway line. A track circuit may be made up of several paths and nodes.
- Points: Points are listed as a specialised type of node. In particular, these special nodes store the information of how long that particular point takes to switch.
- Signals: Signals are also specialised types of node, they have a type (indicating the number of aspects), and also information on the routes that they control entry to.
- Control Tables: Finally there is an encoding of control tables as a list of routes. Each route has information on the track circuits that need to be clear and the points that need to be set (in a particular direction), for the route to be granted to a train.

Considering this data, it is clear that many of the elements stored within the BRaVE tool match those required by the OnTrack toolset's input domain specific languages (see DITTO Deliverable Report 1.1/ Milestone 1: OnTrack Documentation for details). However, the granularity level used within the Brava data for representing the topological layout of the railway is much finer than the details used within OnTrack. Hence the first challenge to overcome was that of abstraction levels within the data.

4.2.1 Challenges in Abstraction Levels

Within OnTrack, the concepts presented in the previous section are all required, however the basic level at which track elements are connected is modelled at a more abstract level than within BRaVE. In particular, the BRaVE modelling approach uses nodes and paths to form the basic topology of the railway. Each node is connected to another using a path. On top of this, track circuits are then positioned over a series of paths. Within OnTrack, the basic connectivity of the railway is modelled as a sequence of track circuits that are connected to make a so-called topological route.

Here, the first challenge of integration is to use the BRaVE data to construct the relevant topological connectivity information for OnTrack. In particular, this process involved writing a graph traversal algorithm, in which variables are indicated using \$ symbols.

The algorithm traverses the topological structure of the BRaVE data by traversing nodes and paths, and constructs a new topological route (which is a list of track circuits) between every pair of signals from the original BRaVE data. This is illustrated by Figure 4.3. Overall, the implementation of this abstraction consists of about 1,200 lines of code. Its running time is below one minute for the examples of Barkston South Junction Werrington Junction as shown in Figure 4.1.

Once this algorithm has been run on the data, the remainder of the construction process involves parsing and performing minor adjustments to the BRaVE dataset.

Figure 4.3: Algorithm to Construct Topological Route

Algorithm to Construct Topological Routes

1. **For each** signal node $\$Node$:
2. $\$TopoRoute = \text{new topological route}$;
3. Set start node of $\$TopoRoute$ to be $\$Node$;
4. **Do**:
5. **For each** path $\$Path$:
6. **If** the start node of $\$Path = \$Node$:
7. $\$Node = \text{end node of } \$Path$;
8. **For each** track circuit TC:
9. **If** $\$Path$ is part of $\$TC$:
10. Add $\$TC$ to $\$TopoRoute$;
11. **While** $\$Node$ is not a signal;
12. Set end node of $\$TopoRoute$ to be $\$Node$;
13. Add $\$TopoRoute$ to OnTrack model;

4.2.2 Challenges in Verification due to Differences in Modelling

Once the BRaVE data was translated into the corresponding OnTrack format, it was possible to use OnTrack's internal model transformations to generate formal models of the scheme plans ready for verification. Here, direct verification of the models produced several counter-examples illustrating that the models were unsafe. This however was not due to the design of the interlocking and control tables, but to different assumptions that are made within the BRaVE and OnTrack toolsets. In order to show the importance of such assumptions, we list several differences between the modelling approaches below.

Overlaps

Firstly, the modelling within OnTrack assumes that all routes are protected by an overlap. However, spurious counter-examples in the verification process highlighted that the data that had been extracted from BRaVE was missing overlaps. This was not a problem with the BRaVE data, where overlaps are just treated as specialised track circuits, but was a problem due to the lack of their explicit representation within the

translated OnTrack data. To solve this problem, the translation process was altered to include the next track circuit after a route ends as an overlap. This is an assumption we have currently made, but would like to remove in future developments of the integration process.

Track Circuits around points

Within the BRaVE data, points are modelled as a particular node with three track circuits being associated with the incoming, normal and reverse branches of the point. However, in OnTrack, points are assumed to be a single track circuit. This simplification is again something that we hope to remove in future development phases of OnTrack. However, to enable a current integration, the translation process was augmented with a manual step to collapse the three track circuits from the BRaVE data into a single track circuit within OnTrack. This of course not only involves editing the topological layout of track circuits, but also the logical data for the control tables. In future revisions of our translation, we hope to make this automatic, or not to need to perform such a step due to improvements within the OnTrack toolset.

Addition of “Entry” and “Exit” tracks

Finally, the last addition that has to be made to the translation is a consideration towards the “open” end points of a particular scheme plan. As the BRaVE data models the entire ECML, when particular scheme plans or regions are extracted for verification, there are clearly going to be areas on these scheme plans where trains can enter and exit the plan. As within the BRaVE data these areas do not exist, there is the need to add them for use within OnTrack. Such an addition is a fairly straightforward extension of our integration process. It simply involves a post translation step that considers the translated scheme plan and looks for track circuits that do not have a “successor” or “predecessor” track circuit attached to them. At these points, specialised OnTrack “Entry” and “Exit” track circuits can be added to the model.

4.3 Verification of the Translated Models

The translation described above has been applied to two exemplar scheme plans from the East Coast main line data exported from BRaVE, namely Barkston South Junction and Werrington Junction. This translation has allowed the models designed in BRaVE to be automatically verified within the OnTrack toolset. The verification process within OnTrack involves the “splitting” of these models into several smaller models for verification (see James et al., 2014, for details). For example, the Barkston South Junction scheme plan is split into a series of 16 smaller scheme plans. Each of these sub scheme plans has then been checked for three safety properties including:

1. Collision-freedom excluding two trains occupying the same track;
2. Run-through-freedom stating that whenever a train enters a point, the point is set to cater for this train;

3. No-derailment saying that whenever a train occupies a point, the point does not move.

Overall, verification times for the 16 smaller scheme plans ranges from 0.16 to 410.86 seconds. These results are encouraging and we expect that OnTrack will be able to verify larger railway areas from the East Coast main line. Although we leave validation of this claim as future work, here we provide a proof of concept.

An abstract “Integrated simulation and formal verification of rail yard designs - an experience report based on the UK east coast mainline” reporting on this work has been submitted by Lei Chen, Phillip James, David Kirkwood, Hoang Nga Nguyen and Markus Roggenbach to the IEEE ICIRT 2016 conference.

4.4 Conclusions and Further Work

Overall, both the BRaVE and OnTrack datasets align at a fairly similar level with regards to data abstraction. However several key messages and lessons should be taken away from the integration process:

- **Sharing Data:** The idea of sharing data between different platforms within rail simulation, verification and design is possible and provides a number of advantages, including reduced time and effort in data entry, guaranteed consistency between models, and a seamless and automatic workflow.
- **Granularity of Data:** Even models on the same level of abstraction can differ in granularity. This provides a challenge in creating an automated model transformation. Naturally, one has to start with the more detailed model (or model element) and construct a suitable abstraction.
- **Modelling assumptions:** The model transformation has to moderate between different sets of assumptions used within the different toolsets. Good documentation and understanding of these assumptions is necessary for success.
- **Adaptability of Involved Tools:** In tool design it would be beneficial to strive for flexibility with regards to modelling assumptions in order to deal with small changes enabling integration.

Thanks to careful consideration of the different modelling assumptions made by both the tools, we have successfully provided a first integration of BRaVE and OnTrack. This integration provides a seamless environment for prototyping, concept development, and safety analysis without the need to re-enter and develop models in different software packages. The usefulness of our approach is illustrated by example, through considering the East Coast main line. BRaVE allows users to simulate and design the overall model for the whole East Coast main line, and several sections of this line, namely Barkston South Junction and Werrington Junction have been translated into

OnTrack and formally verified against various safety properties. It should be noted that a prerequisite for BRaVE is that the scheme plans to be simulated are safe. If this is not the case, BRaVE simulations won't produce reliable data. Intergration with OnTrack allows to perform the necessary safety check.

Looking to the future, we plan to fully automate the model transformations considered in this section and put them to test with more and larger examples from the ECML. Learning from this experience, we aim to transfer our gained knowledge towards studying how to develop further model transformations that allow the integration of OnTrack with the Southampton University Rail Scheduler and the Leeds University Trackula tool.

5. Interaction with Other T1071 Projects (DEDOTS and Safecap Plus)

DITTO is undertaken within the FuTRO (Future Traffic Regulation Optimisation) T952 programme in general and more specifically within T1071 FuTRO: Increase Fundamental Knowledge for Optimising Traffic Management³. This builds on three projects that received EPSRC co-funding under the Capacity at Nodes programme. DITTO is therefore committed to collaborating with the other T1071 projects, DEDOTS – which involves the University of Birmingham and University College London, and Safecap Plus – which involves the University of Newcastle. This collaboration has included the exchange of approved deliverables and six-monthly joint meetings.

We consider that DEDOTS and Safecap Plus are primarily interested in the micro-scale – that is the performance of individual sets of signals and points and their impact on individual train trajectories. Although we have some interests at this scale, DITTO's primary focus is on the meso-scale, that is the assemblage of, for example, points and signals (and tracks and platforms) that make up a node such as Peterborough. We are also interested in the macro-scale, that is the assemblage of nodes such as Peterborough that make up a route such as the ECML. We are also interested in the collection of train services that constitutes a timetable.

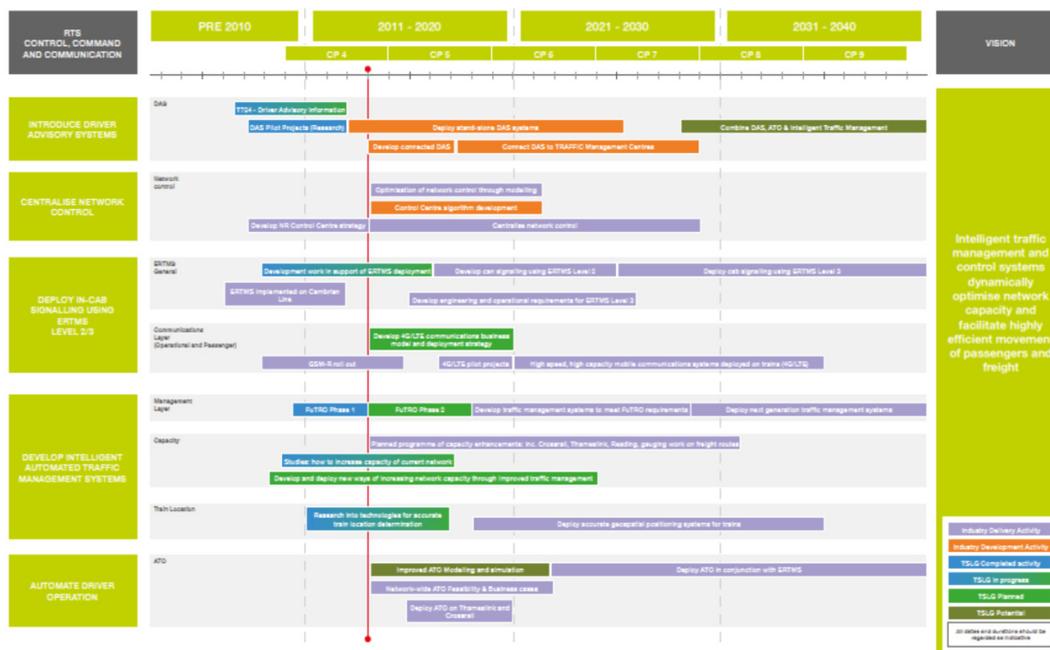
There are, though, synergies with DEDOTS and Safecap Plus, as we have illustrated in section 4 with the work on combining OnTrack and BRaVE. There may also be synergies in terms of safety verification and optimal train control, which should be investigated further. Although DEDOTS (Cambrian Line) and Safecap Plus (Reading, Swindon) both have case studies outside our area of interest, they both also have case studies of the ECML and we will be interested in developing aspects of a common case study and a common test bed.

³ <http://www.rssb.co.uk/library/research-development-and-innovation/research-brief-T952.pdf>

6. Implementation Plan

From its inception we have attempted to align the objectives of DITTO with those of the Rail Technical Strategy (RTS), particularly concerning Control, Command and Communication, with the latter illustrated by Figure 6.1.

Figure 6.1: Rail Technical Strategy



DITTO particularly contributes to the development of intelligent, automated Traffic Management Systems (TMSs), especially with respect to capacity. However, our tools will also have some relevance to driver advisory systems (DAS), centralised network control and automatic train operations (ATO) (and vice-versa). DITTO will complete in September 2017, with in-cab signalling and ERTMS Level 3 deployment envisaged from Control Period 7 onwards (2024). This will give time for the tools developed by DITTO to move up the Technology Readiness Levels. In particular, commercial development will be investigated with our partner organisations, Arup, Siemens Rail Automation and Tracsis. The next stage in development may be investigated through Knowledge Transfer Secondments. At this stage, we believe our quickest wins relate to the use of the stochastic optimisation model to identify additional train paths and the use of the dynamic simulation models to test train control algorithms (e.g. for ERTMS Level 3) and design solutions related to unrealistic demands on braking, unrealistic traction forces, incessant speed fluctuations and unstable traffic flow. We intend to move up the Technology Readiness Levels through involvement in European projects such as Capacity4Rail (2013-17)⁴ and follow-up projects initiated by Shift2Rail.

⁴ <http://www.capacity4rail.eu/>

7. Conclusions and Next Steps

In this interim report we have shown the progress we have made in developing an integrated tool to optimise rail systems. In particular, we have shown the progress we have made in four broad areas.

1. We have developed a stochastic optimisation scheduling model to develop robust timetables and we have shown that this method is feasible using a test environment.
2. We have demonstrated the progress we have made in enhancing the dynamic simulation model, Trackula, to work with ERTMS Level 3 and to determine optimal train trajectories. Using another test environment, we have shown how this modelling work can lead to the identification of stable operations.
3. We have shown how we have assembled data for a study of the East Coast Main Line, with an emphasis on the section between Retford and Huntingdon and a particular focus on Peterborough. We have made further progress in developing nodal Capacity Utilisation Indices and in examining their relationship with Congestion Related Reactionary Delays.
4. We have outlined the latest developments with OnTrack, an open toolset for railway verification, which will be our core interfacing tool. We have demonstrated the interface between OnTrack and BRaVE (Birmingham Railway Virtual Environment) with reference to the East Coast Main Line.

Our next tasks will be to complete the stochastic optimisation (Milestone 9), including an application to Peterborough, and to complete the dynamic simulation, including an extension to ERTMS Level 2 (Milestone 7). We will also complete the interfaces between these tools and the safety verification tool, OnTrack (Milestone 8). We will then bring the lessons we have learnt from this work to develop a good practice guide (Milestone 10).

8. References

- Albrecht, A., Howlett, P., Pudney, P., Vu, X., Zhou, P. (2015a). The key principles of optimal train control-Part 1: Formulation of the model, strategies of optimal type, evolutionary lines, location of optimal switching points. *Transportation Research Part B* (in press)
- Albrecht, A., Howlett, P., Pudney, P., Vu, X., Zhou, P. (2015b). The key principles of optimal train control-Part 2: Existence of an optimal strategy, the local energy minimisation principle, uniqueness, computational techniques. *Transportation Research Part B* (in press).
- Armstrong, J., Preston, J., Hood, I. (2015). Evaluating Capacity Utilisation and its Upper Limits at Railway Nodes. In *Proceedings of the Conference on Advanced Systems in Public Transport 2015 (CASPT2015)*, Rotterdam, Netherlands.
- Armstrong, J. and Preston, J. (2015). *Node-based Capacity and Congestion Estimator documentation*. DITTO Deliverable 2.1/Milestone 4. September.
- Arup (2013). *Recalibrating the Capacity Charge for CP5* [online]. Available from <http://www.networkrail.co.uk/WorkArea/DownloadAsset.aspx?id=30064786027&cd=2> [Accessed 17 February 2016].
- Asnis, I.A., Dmitruk, A.V., Osmolovskii, N.P. (1985). Solution of the problem of the energetically optimal control of the motion of a train by the maximum principle. *USSR Computational Mathematics and Mathematical Physics* 25 (6), 37-44.
- Bando, M., Hasebe, K., Nakayama, A., Sugiyama, Y. (1995) Dynamic model of traffic congestion and numerical simulation. *Physical Review E*, 51, 1035-1042.
- Betts, J. (2010). Practical Methods for Optimal Control and Estimation using Nonlinear Programming. *Society for Industrial and Applied Mathematics*.
- Betktas, T., Kovacs, A. and Potts, C. (2015) *Scenario generation for train timetabling and train scheduling under uncertainty*. DITTO Deliverable 2.2/Milestone 5. September.
- Chen, J. and Liu, R. (2016) *A multi-anticipative car-following model with consideration of the desired following distance*. Working paper, Institute for Transport Studies, University of Leeds.
- Gipps, P. (1981) A behavioural car-following model for computer simulation. *Transportation Research, Part B*, Vol 15, 105-111.
- Howlett, P.G., Pudney, P.J., Vu, X. (2009). Local energy minimisation in optimal train control. *Automatica* 45 (11), 2692-2698.
- James, P., Moller, F., Nguyen, H.N., Roggenbach, M., Schneider, S. and Treharne, H. (2014) Techniques for modelling and verifying railway interlockings . *International Journal on Software Tools for Technology Transfer*.

- James, P, Roggenbach, M. and Moller, F. (2015a) *On Track Documentation*. DITTO Deliverable 1.1/Milestone 1. June.
- James, P, Roggenbach, M. and Moller, F. (2015b) *DITTO Tool Integration: Basic Prototypes for Interfacing*. DITTO Deliverable 1.2/Milestone 2. September.
- John, A.D. and Kobett, D. R. (1978) Grade effects on traffic flow stability and capacity. *NCHRO Report 185*, Transportation Research Board, Washington DC.
- Liu, S.Q. and Kozan, E. (2009) Scheduling trains as a blocking parallel-machine job shop scheduling problem, *Computers & Operations Research* 36, 2840 -2852.
- Liu, R., Ye, H. and Whiteing, A. (2015) *Dynamic simulation for real-time operations of ERTMS Level 3*. DITTO Project Deliverable 3.1/Milestone 6. September.
- Network Rail (2014) *ECML Capacity Options Report to Support Access Application Process*. October.
- Network Rail (2016a). *Historic delay attribution* [online]. Available from <http://www.networkrail.co.uk/transparency/datasets/#H> [Accessed 19 February 2016].
- Network Rail (2016b). *Data feeds* [online]. Available from <http://www.networkrail.co.uk/data-feeds/> [Accessed 19 February 2016].
- Network Rail (2016c). *National Electronic Sectional Appendix* [online]. Available from <http://www.networkrail.co.uk/asp/10563.aspx> [Accessed 19 February 2016].
- Network Rail (2016d). *Working Timetable (WTT)* [online]. Available from [https://www.networkrail.co.uk/browseDirectory.aspx?dir=\Timetables\Working%20Timetable%20\(WTT\)&root=\Timetables\Working%20Timetable%20\(WTT\)](https://www.networkrail.co.uk/browseDirectory.aspx?dir=\Timetables\Working%20Timetable%20(WTT)&root=\Timetables\Working%20Timetable%20(WTT)) [Accessed 19 February 2016].
- Wendler, E. (2007) Influence of ETCS on Line Capacity. *UIC ERTMS World Conference* 11-13 September. Berne, Switzerland.
http://www.uic.org/cdrom/2007/14_ERTMS_conference07/docs/presentations/G4_wendler.pdf [Accessed 25 February 2016]

Acknowledgement

We are grateful for the contributions of our colleagues to this deliverable, including Tony Whiteing at the University of Leeds, and Tolga Bektas and Chris Potts at the University of Southampton.