DITTO Project Deliverable 3.1

Milestone 6

Dynamic Simulation for Real-Time Operations of ERTMS Level 3.

September 2015

Ronghui Liu, Hongbo Ye and Tony Whiteing, University of Leeds, UK.

Abstract

This document reports on progress made towards DITTO Project Deliverable 3.1, on the development and testing of a rail network simulation model and the use of the resulting model to examine the traffic management and optimal control strategies for ERTMS Level 3.

Whilst the advanced technologies employed by ERTMS Level 3, moving away from track-based detection and line-side signalling of the current system to train-borne detection and communication, offer the potential to provide a railway system with enhanced efficiency, improved safety and increased capacity, the radically different methods of operation mean that successful implementation of ERTMS Level 3 (and more advanced systems) depends on key traffic management challenges being addressed.

A model of railway networks operating under the ERTMS Level 3 system is required to help design and test such strategies. We describe the effort made to date in the DITTO project in developing a railway network simulation model for the simulation of traffic performance under ERTMS Level 3. This report explains the various ERTMS Levels and identifies that a key challenge to the success of ERTMS Level 3 lies in the development of network-wide intelligent traffic management and control strategies. Microsimulation is identified as a methodology that can be used to develop tools for the development of such strategies. The state of the art in microsimulation is reviewed, highlighting its key

dimensions. A network simulation model, capable of simulating train operations at Level 3 is then developed. We first set out the general principles underpinning this approach and then provide more detail on the algorithmic approach adopted in the model. The model, once established, is then tested and applied on a range of relevant scenarios. The report ends by setting out the further work on this project deliverable.

1. Introduction

1.1 Background and objectives of the present study

WA3 of the DITTO project proposal envisaged a programme of work in the field of dynamic simulation of railway operations, comprised of two work streams. Work area 3.1 proposed the development, testing and application of a network simulation model whilst in parallel 3.2 would develop a network optimisation approach. This particular deliverable reports on progress towards work stream 3.1 on dynamic simulation.

This work is being conducted in the context of future railway operating paradigms, notably embracing the advanced technologies employed by ERTMS Level 3 (hereafter Level 3) which offer very significant potential to provide a railway system with enhanced efficiency, improved safety and increased capacity. ERTMS Level 3 operates on a moving block system with the locations of the trains continuously monitored and communicated to the control centre. The individual trains themselves become effective moving blocks. The advantage of Level 3 lies in not only removing the need for track-based detection thus reducing infrastructure installation and maintenance cost, but also in reducing the safety headways between trains therefore increasing line capacity.

Provision of a distinct and individual train-based moving block system, traffic management (TM) increases the challenges for Level 3. The control of the speed and spacing between trains, as well as managing the network-wide paths of trains in real time, are clearly central to the success of the system. Yet there is a lack of basic research knowledge in TM in this area.

With this in mind, the primary objectives of this WA, as originally constructed, were to:

- Examine the applicability of developments in road traffic management technologies to rail.
- Consider the implications for ERTMS Levels 2 and 3.

Within the over-arching objectives set out above, more specific objectives of this programme of work are as follows.

- To develop a rail simulation model, based on the principles of road traffic microsimulation.
- To consider the implications of train following behaviour.

1.2 Outline of the deliverable

Section 2 of the Deliverable explains the various ERTMS Levels and sets out the key challenge to the success of the ERTMS Level 3 system in the development of network-wide intelligent traffic management and control strategies.

Section 3 reviews the state of the art in microsimulation, highlighting the key dimensions of existing approaches (microscopic vs macroscopic, discrete time vs discrete event, deterministic vs stochastic).

A railway network simulation model capable of simulating at ERTMS Level 3 is then set out. Section 4 deals with the general principles whilst Section 5 explains the algorithmic approach developed in the model.

Section 6 demonstrates the application of the model to a range of scenarios with other ongoing work plans for further work in the project discussed in Section 7.

2. ERTMS

2.1 ERTMS Levels

The European Rail Traffic Management System (ERTMS) is proposed to overcome the incompatibility of the more than 20 existing railway control systems in European countries, which currently act as a major obstacle for the transnational train operation of both passenger and freight transport. The ERTMS consists of two major subsystems, namely the signalling and control component, European Train Control System (ETCS), and the telecommunication component, Global System for Mobile Communication - Railway (GSM-R). The ETCS is a unified and improved cab-signalling and automatic train protection (ATP) system for the replacement of the existing national ATP systems. The GSM-R is based on the GSM with customized features for railway operation. The ERTMS brings improvement for the current railway system in various respects, including (UNIFE, 2014a, 2014g):

- improving the cross-border interoperability for transnational train operation and the infrastructure interoperability among different suppliers;
- reducing the system complexity and infrastructure cost, both on-board and trackside;
- reducing the headway between trains so increasing railway capacity;
- increasing the train speed, reliability and punctuality; and
- reducing the scope for human error and ensuring safety by applying the brakes when the driver doesn't follow the movement authorities or speed limits.

ERTMS, especially ERTMS Level 2, has been successfully implemented and brings significant improvement for the railway systems in various European countries, such as Italy, Spain, Switzerland, Belgium and the Netherlands (UNIFE, 2014c, 2014d, 2014e, 2014h). The potential benefit also attracts attention and investment from many non-European countries (UNIFE, 2014f).

In ERTMS, the train and the control centre work together during the train running and control process. The control centre obtains the train location and/or the occupancy of the tracks, determines the train movement authorities, and transmits them to the trains. Based on the movement authorities, the trains then calculate the braking curves. According to how the information is transmitted between train and the control centre, and how the block system works, the ERTMS is categorized into three operation levels listed as follows (Hayat, 2013; Jabri et al., 2010; Qiu et al., 2014; UNIFE, 2014b).

- **ERTMS Level 1** can work compatibly with the existing lineside signals. The information is transmitted between the train and the control centre through the balises installed on the tracks. The balises transmit the movement authorities and other control parameters to the trains running over it, and at the same time send the train location to the control centre. The train integrity detection is based on track circuits or axle counters.

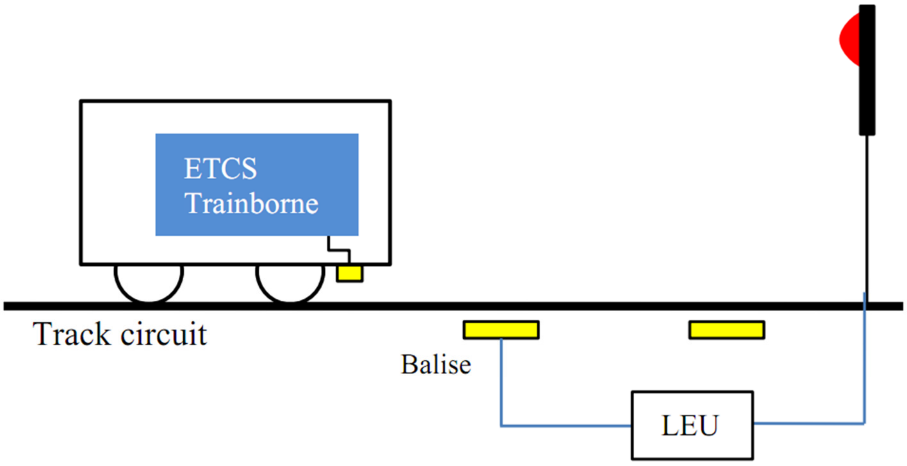


Figure 1. Illustration of ERTMS Level 1 (source: UNIFE, 2014b).

- **ERTMS Level 2** does not require lineside signals. The movement authorities and other real-time line-specific data are transmitted from radio block centre (RBC) to the train through GSM-R. The balises are used for the transmission of the "fixed messages" such as location and speed limit.

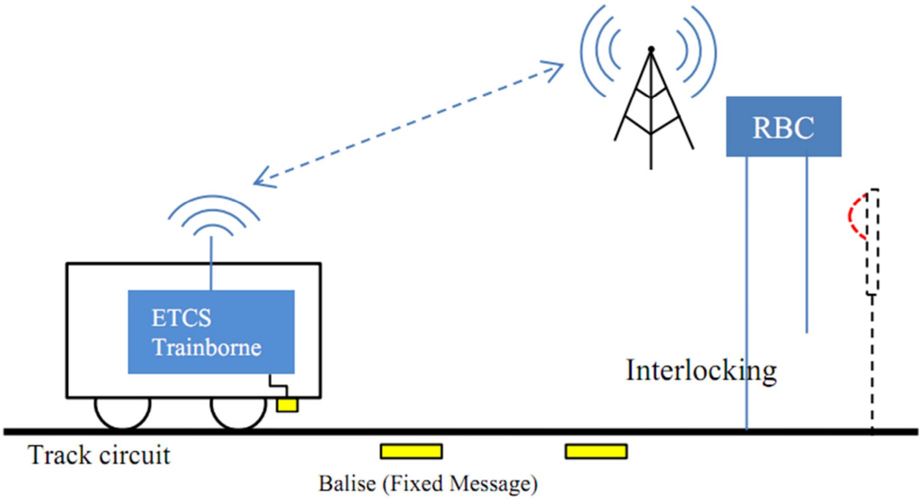


Figure 2. Illustration of ERTMS Level 2 (source: UNIFE, 2014b).

- **ERTMS Level 3** is a conceptual level which introduces the moving block system. In contrast to Level 1 and Level 2, the train integrity is checked by the train itself, and the control centre obtains the continuous train location from the train rather than from the track-based detection equipment.

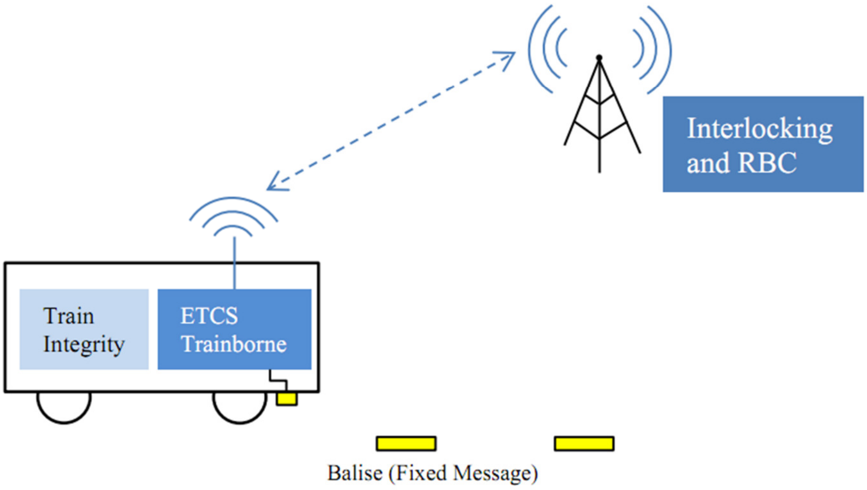


Figure 3. Illustration of ERTMS Level 3 (source: UNIFE, 2014b).

The key features of the different levels of ERTMS are summarized in the following table.

Table 1. Key features of different ERTMS levels

ERTMS	Block System	Transmission of movement authority
Level 1	Fixed block	Balises
Level 2	Fixed block	GSM-R
Level 3	Moving block	GSM-R

2.2 Key challenges for modelling ERTMS Level 3

Moving from track-based detection and line-side signalling of the current system, to train-borne detection and communication in Levels 3 (and above), ERTMS Level 3 (and above) offers the potentials for rapid response to changes in network and traffic conditions and for enhanced capacity and performance.

With the moving block system introduced in ERTMS Level 3, the train now only has to follow the speed limits and maintain a safe distance from the train in front, which increases the railway capacity. Also with the real-time and detailed train running information, the control centre now has the opportunity to arrange the movement authorities more sophisticatedly, which introduces challenges for the real-time scheduling and control algorithms. It is also possible for the control centre to provide sophisticated speed and/or acceleration profiles for the trains to follow, which is helpful for energy saving since the energy consumption is related to the detailed running status of the trains but is considered locally in the current train operation.

A key challenge to the success of the ERTMS Level 3 system therefore lies in the development of network-wide intelligent traffic management and control strategies. A model of railway networks under ERTMS Level 3 system is required to help design and test the proposed traffic management strategies. In this deliverable, we describe the effort made in DITTO in developing a railway network simulation model for the simulation of traffic performance under ERTMS Level 3.

3. Simulation Models of Train Operations

Simulation models include mathematical and logical abstractions of real-world systems and implement them in computer software (Banks and Carson, 1984). Different to analytical models, time is explicitly represented in simulation models. They are therefore capable of, in fact designed for, representing the *dynamical* behaviour of a system (Law and Kelton, 2000).

Simulation models have been applied to a variety of situations in rail planning and operations (Asuka and Komaya, 1997). In the UK, the then British Rail had been using a computer simulation package, General Area Time-based Train Simulator (GATTS), since the 1970s to aid planning changes to infrastructure and timetables and for the design of train regulation strategies.

For an extensive survey of the simulation software tools developed for railway systems, see the review by Barber et al. (2007). We describe below the different simulation approaches adopted in the literature.

Adapted from the definitions of Siefer (2008) and Liu et al. (2013), simulation models can be categorised as follows:

- Microscopic vs Macroscopic

Microscopic models describe train movements in terms of the train performance and track conditions and aim to reproduce the actual operation of the rail system over a user-defined time period (Asuka and Komaya, 1997). Generally speaking microscopic models take as input the infrastructure parameters, signalling systems, rolling stock parameters and the timetable (all in extensive detail) into account and replicate performance over a given time period.

Examples of models in this category include OpenTrack (Nash and Huerlimann, 2004), RailSys (Bendfeldt, et al., 2000; Radtke and Bendfeldt, 2001), SimMETRO (Kooutsopoulos and Wang, 2009), VISIONS (McGuire and Linder, 1994), and EGTRAIN (Euaglietta, 2014; Corman and Quaglietta, 2015).

In contrast to such microscopic models, macroscopic models do not model individual unit (e.g. train) operations nor do they consider how trains are impacted by other trains (Nash and Huerlimann, 2004). An example of a macroscopic model is NEMO (Kettner, et al., 2003). Input data such as infrastructure is modelled with less detail, providing benefits such as reduced computational run times (Huber and Wilfinger, 2006).

- Discrete Time vs Discrete Event

In discrete time simulation, the system is observed and updated at regular time intervals. This approach is widely adopted to simulate systems whose state variables change continuously with time, e.g. train trajectories. VISIONS, RailSys and OpenTrack and SIMMetro are examples of discrete time simulation models. Such models simulate all the trains operating in the modelled network at the same time, they offer a good way of simulating realistic operating conditions. For example, they can be used to determine the impact of delays and their propagation through the network (Radtke, 2006).

In discrete event simulation, the system is observed and updated every time an event takes place. This approach is most suited to model systems whose entities change instantaneously at separate points in time, e.g. railway signals.

In general, the event scanning method is faster to run, but becomes complicated (and less efficient) when the events to handle increase. It depends critically on the identification and definitions of the events. For example, for the simulation of train movements, the events may include and correspond to the actions when the train has to adjust its acceleration (and thus the speed). An example list of events and the corresponding actions are listed in **Error! Reference source not found.**.

Table 2. Example of the discrete events for railway simulation

Event	**Action**
Departing from the station	Accelerate
Reaching the speed limit	Set acceleration equal to zero
The train is asked to stop at the station ahead, which is within its breaking distance	Brake
The train ahead is within the breaking distance	Brake
Arriving at the station and has to dwell	Set acceleration equal to zero

The discrete-event method has also been used for the train scheduling/rescheduling process, by strategically arranging the overtaking and passing plans to avoid deadlock and optimize the running time or delay. Medanic and Dorfman (2002) and Dorfman and Medanic (2004) used the discrete-event method as a train scheduling tool for the first time, aiming to minimize the total running time of the trains. A greedy strategy was used to determine the passing and overtaking priorities; and a simple capacity check algorithm was proposed to avoid the deadlock. Li et al. (2008) improved the model in several aspects. First of all, the acceleration and braking are considered in the simulation. Secondly, priority of the focal train is determined by calculating and comparing the total delay caused by the focal trains under different plans, from the current time until its arrival at the destination. Thirdly, the deadlock prevention algorithm is improved by distinguishing the relative and absolute deadlocks and dealing with them differently. Li et al. (2014) further embedded the discrete-event simulation in a more detailed train scheduling problem of minimizing total delay. The deadlock detection and prevention mechanism was updated by introducing the concepts of a positive plug and a negative plug. Theoretically speaking, the discrete-event-based optimization could not achieve the true optimal solution; however, as reported in the above-mentioned literature, the solutions obtained by the discrete-event-based optimization are very close to the true optimal solution obtained by the mathematical programming method, while the calculation time of the former is much smaller than that of the latter.

- Deterministic vs Stochastic

Deterministic models estimate the arrival, departure and running times according to the schedule (Siefer, 2008). Their primary use is for the preliminary design of a timetable. On the other hand, stochastic models utilise statistical distributions of arrival, departure and running times. Simulation packages mentioned above (such as RailSys and OpenTrack) are equally capable of performing in the deterministic mode as in the stochastic mode, and they contain facilities that enable users to determine the robustness of timetables in the face of disruptions (e.g. rail vehicle breakdown on track) and incidents (e.g. inclement weather) (Watson, 2005).

4. A Railway Network Simulation Model

In order to facilitate the development of traffic management (TM) strategies for Level 3, a simulation model was developed to represent the Level 3 system and to explore TM strategies on the performance at the line and network level.

In broad terms, any modelling framework aimed at evaluating the operations of ERTMS Level 3 (or above) should be able to represent the detailed technologies employed and the interactions between trains and the control systems. The following areas are identified as the key requirements to be represented in such a model:

Network design characteristics:

- A variety of railway lines and tracks, including single and double tracks with different speed limits, passing loops to allow potential overtaking;
- Junctions and crossings, including the directions of travel and conflicts between lines;
- Stations and platforms, including the length of the platforms to allow modelling the choice of platforms according to the train type and multiple stopping trains in a single platform;

Train characteristics:

- Types of trains and train characteristics, including their length, maximum speed, acceleration and deceleration capability
- Scheduled train timetable, including trains paths and departure/arrival times at stations;

Traffic behaviour and control strategies

- Train following behaviour
- Junction/station conflict resolutions

Network and traffic conditions:

- Planned or unplanned disruptions on the network;
- Traffic disruptions, and traffic congestion on network;

Output specifications:

- Speed, journey times and delays for the different trains, on different lines/tracks, and over the whole network;
- Throughputs (capacity) at different locations in the network.

A faithful representation of the above features and traffic interactions can only be achieved using a fully dynamic simulation model, where traffic interactions can be modelled, and the network and traffic conditions performance monitored continuously in space and time.

For these purposes, it was decided to adapt an existing traffic microsimulation model DRACULA (standing for Dynamic Route Assignment Combing User Learning and microsimulAtion) (Liu, 2005; 2010; Liu et al., 2006) which was developed to simulate road traffic in road networks, in order to simulate trains in railway networks. The new railway simulation model, in keeping with its origin, is code named TrackULA (for **Track** **U**nified simu**L**ation **A**lgorithms).

TrackULA is a microscopic simulation model; it represents the movement of individual trains along rail tracks and through railway junctions and stations. The model is based on a discrete time simulation framework, where the speeds and locations of the trains are updated at a fixed time internal (default being one second) according to a 'train following' model and junction/station control. The basic modelling framework and its inputs and outputs are illustrated in Fig. 1, while Table 3 lists the key railway entities (agents) and their paths of communications represented in the model.

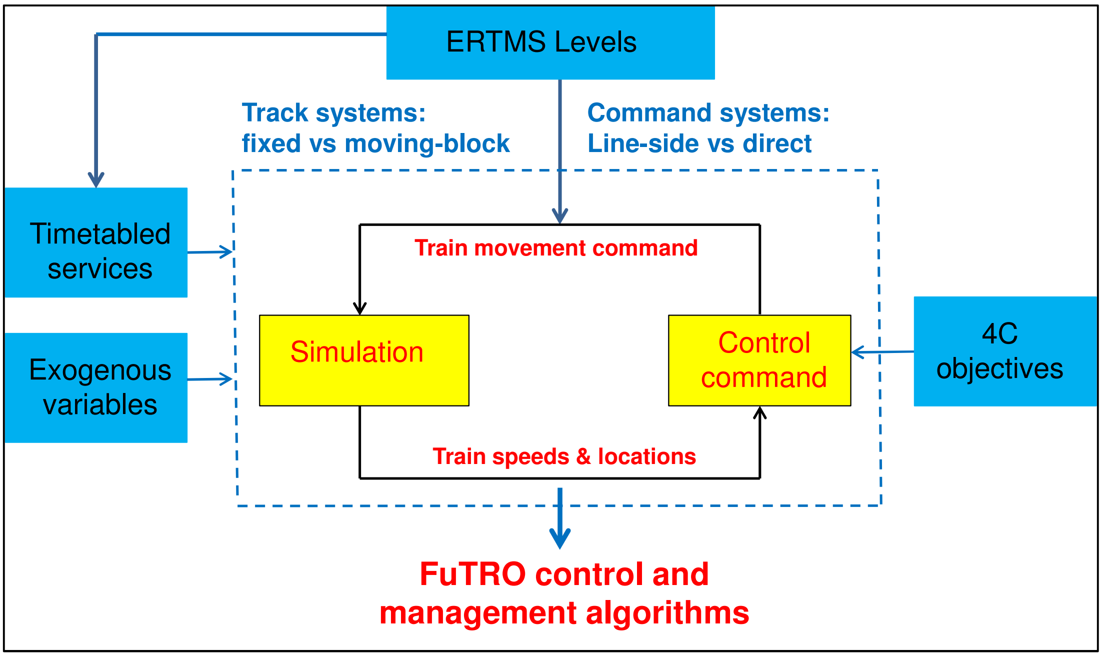
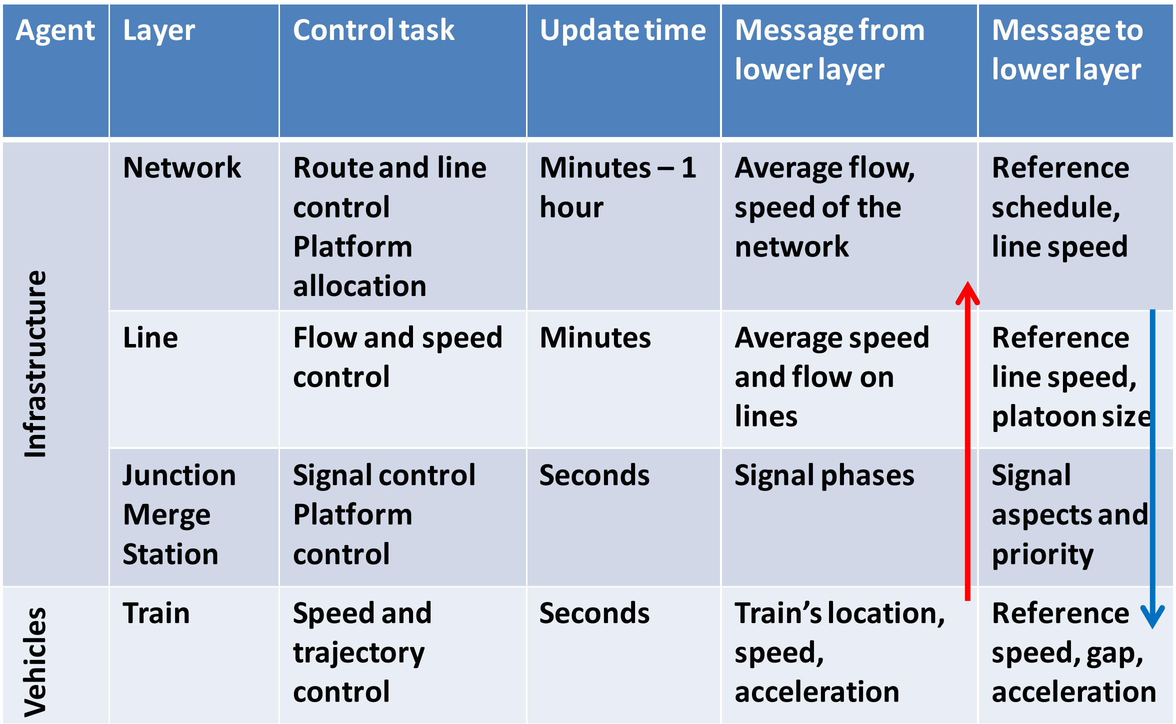


Figure 1: **The TrackULA simulation and control framework (inside the dashed box), and its required inputs and expected outputs.**

Table 3. The railway infrastructure and vehicle entities and communication messages represented in the TrackULA framework.



Agent	Layer	Control task	Update time	Message from lower layer	Message to lower layer
Infrastructure	Network	Route and line control Platform allocation	Minutes – 1 hour	Average flow, speed of the network	Reference schedule, line speed
	Line	Flow and speed control	Minutes	Average speed and flow on lines	Reference line speed, platoon size
	Junction Merge Station	Signal control Platform control	Seconds	Signal phases	Signal aspects and priority
Vehicles	Train	Speed and trajectory control	Seconds	Train's location, speed, acceleration	Reference speed, gap, acceleration

5. The TrackULA Simulation Algorithms

The TrackULA model is a time-based simulation of the movements of individual trains through a network. Unlike some other railway simulation tools such as RailSys and OpenTrack, this model does not model the detailed tractions of the individual trains. Instead, the model calculates the equations of motion, based on a 'train-following' model and the control command of the ERTMS system modelled. More specifically, it calculates the acceleration, speed and position of each train at every time interval, with given acceleration/deceleration profiles whose values are sourced from the literature and in consultation with the railway industry. In railway simulation terms (as defined in Section 3 above), this model falls in between micro- and macro-simulation and could be considered a meso-scopic simulation model.

Another distinct feature of this simulation model, as compared to other existing railway simulation tools, is that it is a stochastic model. It can model stochastic travel times (as opposed to deterministic, scheduled times), disruption, and heterogeneous train characteristics, and variations in drivers' experience and driving behaviour with a given probability distribution, and heterogeneous train operating and train drivers behaviour (e.g. by train type, drivers, and operating rules).

The model outputs each individual train's second-by-second space-time trajectories as well as route/line-based and network-wide statistics. As a result of the stochastic modelling, the simulation outputs include not only the means but also the variances and probability distributions of performance measures.

The simulated train movements are animated through a graphical user interface, which is useful both for debugging purposes and for examining the control impacts on the trains as well as network-wide traffic flow conditions. Fig. 2 and Fig. 3 present animation snapshots of the simulated traffic dynamics in a road and a rail system respectively.

The road simulation model DRACULA has been applied in many studies to help design and evaluate traffic management schemes, and to test future Intelligent Transportation Systems.

In a previous research project (Mei and Liu, 2013), the software has been adapted to simulate a fixed block signalling system (Liu et al, 2013) and was used to evaluate options for conflict resolution at isolated railway junctions (see the snapshots of the simulation in Fig. 3).

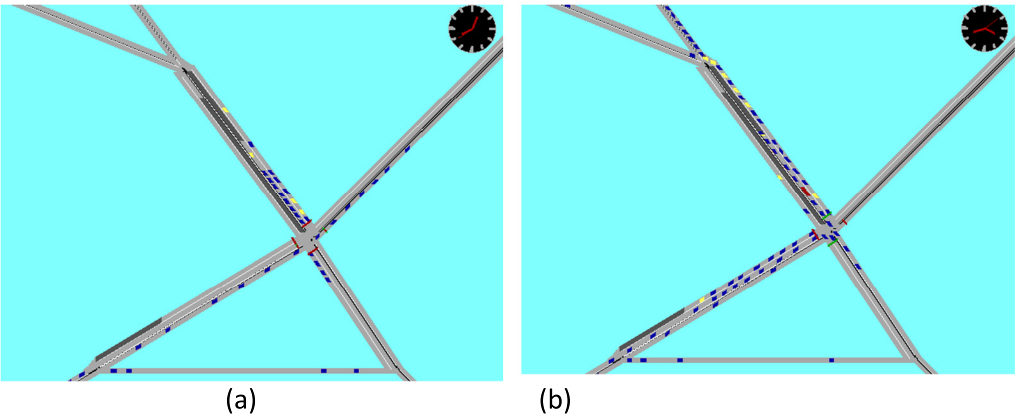


Figure 2: **A snap-shot of the DRACULA simulation of road traffic condition at the Clifton Green intersection, York, during a morning rush hour. It shows the building-up of the traffic congestion level between (a) 08:04, and (b) 08:20.**

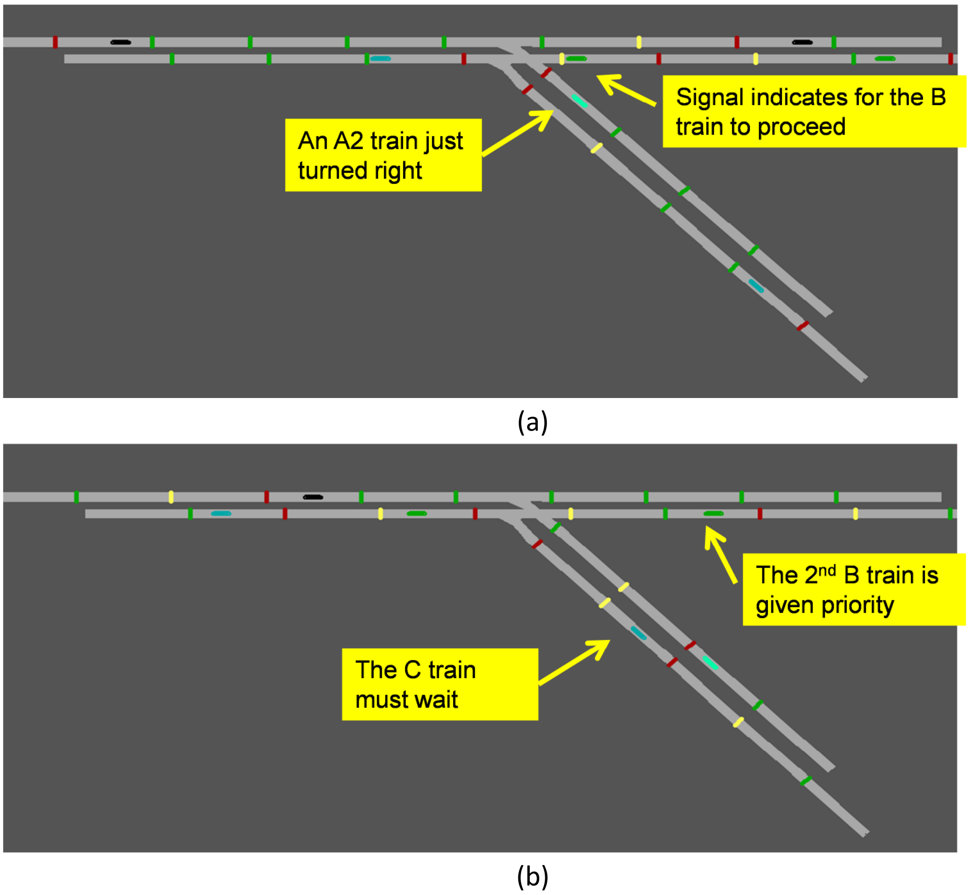

Signal indicates for the B train to proceed
An A2 train just turned right
(a)
The 2nd B train is given priority
The C train must wait
(b)

Figure 3: **A snapshot of TrackULA simulation of signalling in a fixed block system at classical railway junctions.**

We describe in the following sub-sections the core functions of TrackULA, adapted from and further developed based on the general framework of DRACULA.

5.1 Simulation loop

The simulation is based on a fixed time increments; the speeds and positions of individual trains are updated at an increment of one second by default. Spatially, the simulation is continuous in that a train can be positioned at any location along a track, a mimic of the moving block system.

The simulation starts with an initialisation process which loads the input data (including the network data, train timetable data, ERTMS system command architecture, and other exogenous simulation parameter values) and sets the simulation clock (to be zero by default). It then runs through an iterative procedure at a pre-defined time increment (one second by default), within which the following simulation tasks are performed:

1. Start the simulation: load network and train timetable data. Set simulation clock t=0.
2. Train generation: generate new entry of trains according to the given schedule, assign the scheduled route to the trains and place them on their entrance track (see detailed description on models of train timetable and route in Section 5.2); Each train is assigned a set or train-driver characteristics (details are described in Section 5.3).
3. Train simulation: loop through all trains in the network, and for each one of them:
    (a) Calculate the new acceleration and speed for the train according to a train-following model (Section 5.4);
    (b) Advance the train to its new position. If the train has reached the end of the link, pass it to its next link en-route, or if the train has arrived at its destination, remove it from the network;
    (c) Record the train performance measures (Section 5.6);
4. Control command for junctions (locations where train paths can diverge, merge or cross): loop over all such locations in the network, and for each one of them:
    (d) Obtain position data for all trains in the vicinity of a junction, identify potential conflicting train trajectories;
    (e) Calculate the expected arrival times of the conflicting trains, and devise a priority plan and issue acceleration/deceleration commands to each train (ongoing);
5. Control command for stations: loop over all stations in the network, and for each:
    (f) Obtain trains at the station, and their scheduled dwell times and departure time;
    (g) Check the times they have stopped at the station, if exceeding the scheduled dwell time, go to 5(h);
    (h) Hold the train at the station until their schedule departure time (Section 5.5);
6. Update the graphical animation;
7. Update the simulation clock $t:=t+\Delta t$, and go to step 2 until simulation ends.

A schematic illustration of the above procedure is also illustrated in Fig. 3 below.

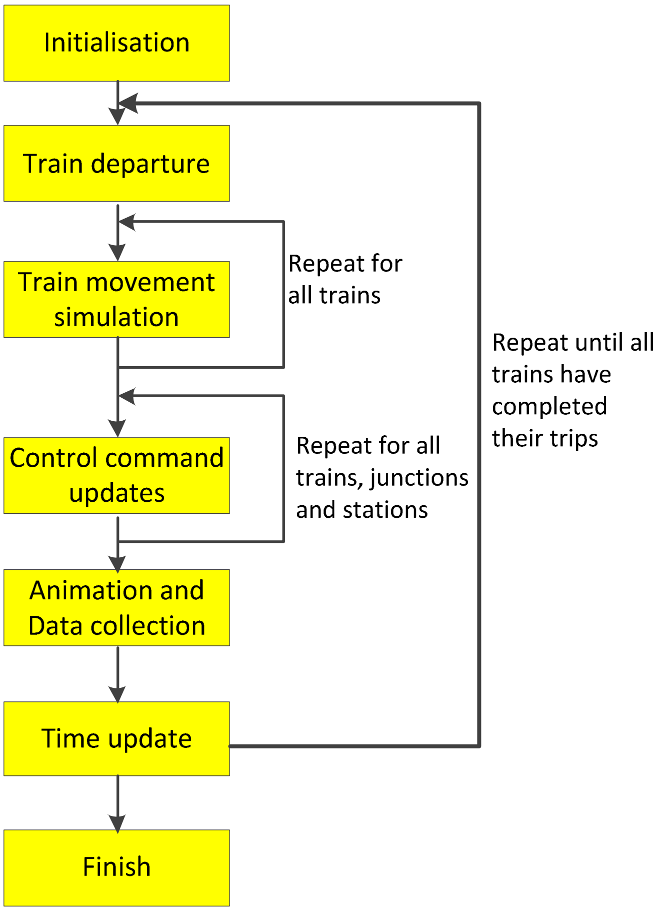

Initialisation
Train departure
Train movement simulation
Repeat for all trains
Control command updates
Repeat for all trains, junctions and stations
Animation and Data collection
Time update
Repeat until all trains have completed their trips
Finish

Figure 4: **The train simulation loop in TrackULA.**

5.2 Railway network representation

The railway network is represented by nodes and links. A node is used to represent a terminus where a train enters or leaves the network, a signal box (for fixed block systems), a station, an intersection (when two or more lines cross each other), or a merge point (where two tracks from the same or different lines merge into a single track).

A link is a directional running line connecting two nodes, and is specified by its upstream and downstream nodes, the turns permitted at the downstream nodes, and the speed limit of the line. Bi-drectional track not implemented in the current version of TrackULA – this is a feature to be considered in the future.

5.3 Railway timetable and train route representations

A railway timetable is represented in terms of trains routes, and departure/arrival times of each train en-route. Train and crew scheduling is not represented in this model; all scheduled trains and crew members are assumed to be available.

A train route is a one-way path connecting a sequence of nodes in the network, the stations at which the train is scheduled to stop, the dwell time at each stop, and the type of train (e.g. passenger/freight, size/length of the train) scheduled to operate it.

Railway timetable and train route are assumed to be given; they are essential input data to the simulation model.

5.4 Train and driver behaviour representations

A train-driver unit (TDU) is modelled to represent the rolling stock and the train drivers' driving behaviour.

Trains are individually represented in the simulation, according to their scheduled time of arrival into the network, scheduled route, and rolling stock type.

Trains are generated and entered to the network at their scheduled departure times. In the current version of the software, the trains follow the pre-specified fixed route. Dynamic re-routing, such as change of track (between a slow and fast track on the same line) or a diversion (change of a section of the previous route), can be implemented in the future.

Different rolling stock types are modelled. A train type is characterized by its physical length (the width or a train is modelled here), maximum speed, acceleration and deceleration capability. A train is modelled as a rigid body; the whole train moves at the same speed and it accelerates instantly.

Driving behaviour is represented in the model in terms of drivers' reaction times, which are drawn from a normal distribution with means and variances representing the average and variations in drivers' response capabilities.

5.5 Train movement simulation

The essential property of the TrackULA train simulation model is that the trains move in real-time and their space-time trajectories are determined by a train-following model and the control commands at junctions and stations.

In modelling the movements of a train, we do not solve differential equations to calculate the acceleration of the train based on its load, running speed, gradient and curvature resistance on the track. Rather, we represent the motion of the train by calculating the equation of motion, noting its acceleration/deceleration, speed and position at every point on time on a fine time interval.

The train movements are therefore determined based on its desired (or maximum) speed, the line speed limit, and need to stop at stations. In congested parts of the network, the train's movements are also constrained by the train(s) in front along the same line, and/or its neighbouring train(s) from different lines approaching the same intersection or merge node.

In this section, we describe a train-following model developed to compute the space-time movements of a train on a single track.

We first introduce the notations used in the train-following model:

Indices

n Index of trains travelling along a single track. Train *n* follows train *n-1* in front.

t Index of time

Model parameters

Δt Simulation time increment (s)

V^* An optimal following speed (m/s)

S^* An optimal following space gap (m)

J_n Jerk (i.e. the rate of change of acceleration) of train *n* (m/s^3)

A_n Maximum acceleration of train *n* (m/s^2)

D_n Maximum deceleration of train *n* (m/s^2)

V_n Maximum or desired speed of train *n* (m/s)

L_n Length of train *n* (m)

Model variables

$a_n(t)$ Acceleration of train *n* at time *t* (m/s^2)

$v_n(t)$ Speed of train *n* at time *t* (m/s)

$x_n(t)$ Location of the front of the train *n* at time *t*, relative to the start of the line (m)

The train–following model is a theory describing how one train follows another train along a continuous track under ERTMS Level 3. The model calculates, at every time instance, a train's acceleration and speed based on its own desired movements and the relative speed and distance to its preceding train(s). Fig. 5 illustrates the general concept of the train-following model under ERTMS Level 3.

We assume that under Level 3 (or higher) ERTMS systems, trains' locations and speeds are known and are communicated either directly to each other (via the train-to-train, T2T, communication), or through a control-command centre (via train-to-infrastructure T2I and infrastructure-to-train I2T communication).

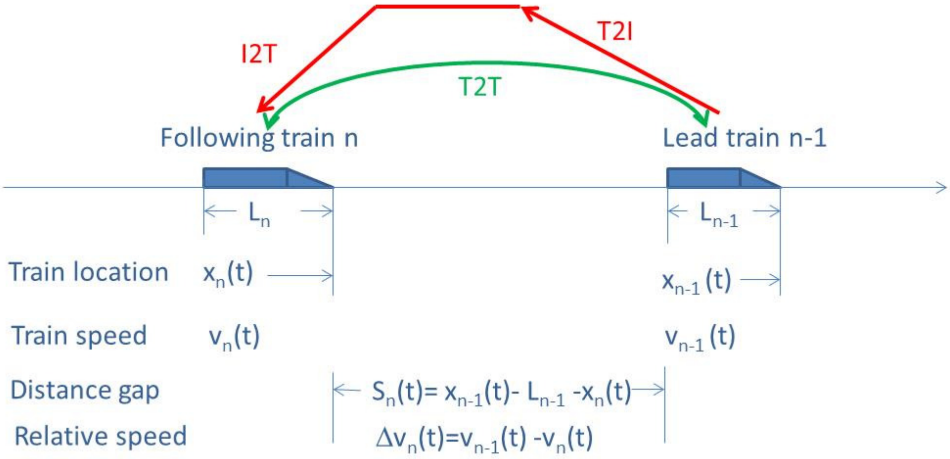
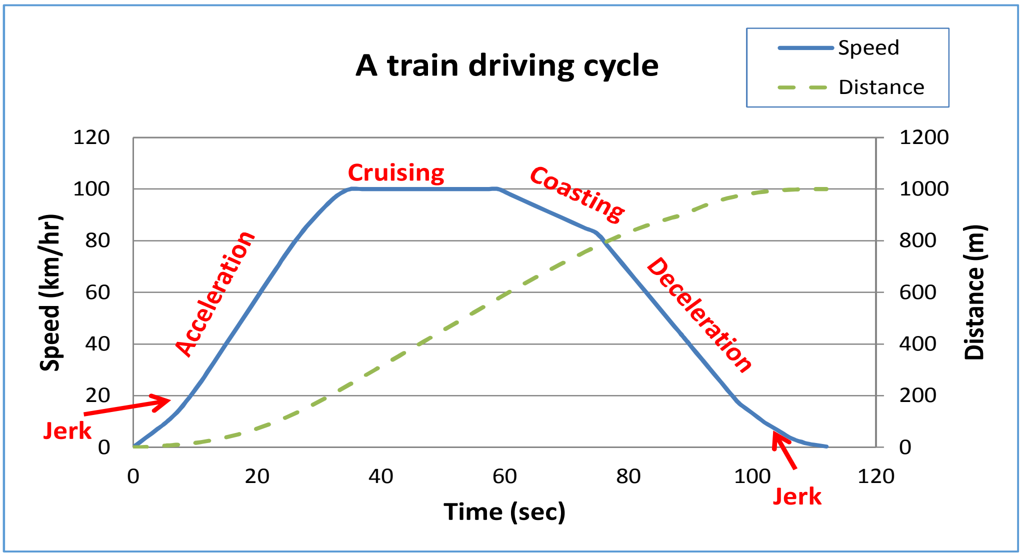


Figure 5: **A train-following scenario, and key model variables in a train-following model.**

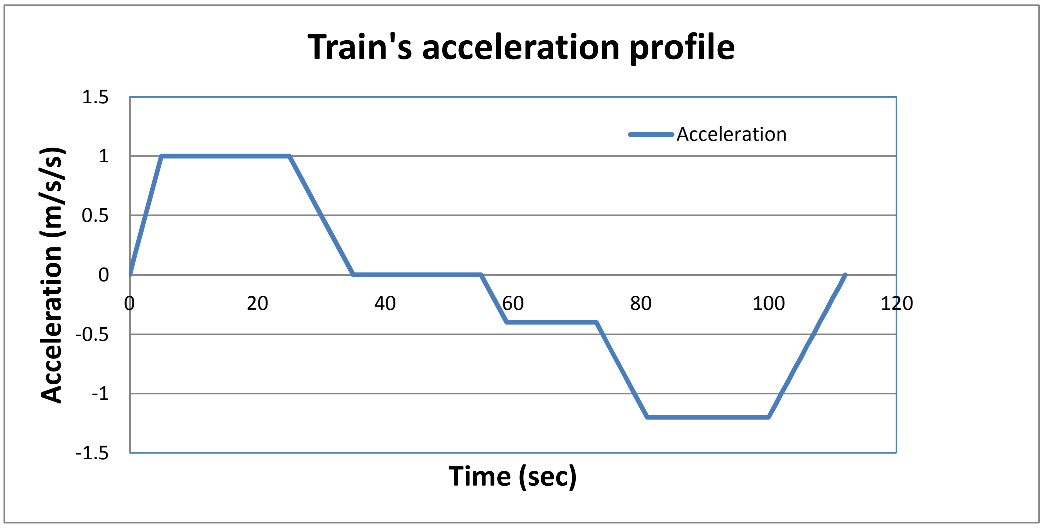
Depending on the magnitude of the relative distance, a train is considered moving under two regimes: (1) a free-flow regime, and (2) a following regime. We present below the mathematical equations used to model the acceleration of a train as they traverse the route under each of the two different regimes. A minimum acceleration between the two is chosen as the final solution.

- Free-flow model

When a train is the lead vehicle on the line (till the next node), the train will not be under the influence of the train in front and will simply follow its own desired driving cycle. Fig. 5a illustrates a typical driving cycle in a free-flow regime.



(a)



(b)

Figure 6: **A free-flow driving cycle as represented in terms of: (a) the speed profile of the train over time, and (b) the acceleration profile of the train.**

The train first accelerates freely to its desired speed or the max line speed whichever is the smaller one. It then maintains and cruises at that speed, till approaching a ‘braking distance’ to the next stop. The train may also be coast for some distance at the track’s resistance, before applying its maximum deceleration to stop at the next stopping point.

The train-following model is adapted from car-following models widely used in road traffic simulation. In most of the car-following models, the acceleration and deceleration of the cars are assumed to occur instantaneously and at the car's maximum acceleration/deceleration capability.

In this train-following model, we introduce a new variable, 'jerk' *J,* to represent the rate of change in a train's acceleration and deceleration. The acceleration profile for a typical driving cycle can thus be given as in Fig. 5b.

- Train-following model

Based on the speeds and locations of the leader and follower trains at time t, the acceleration of the following train at the next time instant $t+\Delta t$ can be formulated as in eq. (1) below:

$$a_n(t+\Delta t)=\alpha[V^*-v_n(t)]\mathrm{H}[V^*-v_n(t)]+\beta[s_n(t)-S^*]\mathrm{H}[s_n(t)-S^*] \quad (1)$$

where $s_n(t)=x_{n-1}(t)-L_{n-1}-x_n(t)$ is the space gap between the head of the following train *n* and the trail of the train *n-1* in front. $\mathrm{H}(x)$ is a Heaviside step function defined as:

$$H[x]=\begin{cases}0, & x\le 0\\ 1, & x>0\end{cases} \quad (2)$$

The first part of the RHS of eq. (1) represents the desire of the train drivers to accelerate to reach an optimal speed *V**. The choice of this 'optimal' speed depends on the traffic management strategy. For example, if the control strategy is to form a platoon of trains, for the benefit of energy consumption for example, this optimal speed can be chosen to be that of the train in front.

The 2nd part of the RHS of eq. (1) restrain the following train to keep a safe space headway (*S**) to the train in front. Again, the choice of the parameter value *S** represents the balance between safety and capacity objectives. Clearly, a larger *S** value leads to a safer system but at the expense of a lower throughputs.

Giving the acceleration (and deceleration) profile of the train, then, according to the Newton's equation of motion, the new speed and location of the train can be computed according to the following equations:

$$v_n(t+\Delta t) = v_n(t) + a_n(t+\Delta t)\Delta t \quad (3)$$

$$x_n(t+\Delta t) = x_n(t) + \frac{1}{2}[v_n(t) + v_n(t+\Delta t)]\Delta t \quad (4)$$

5.6 Control command simulation

Ideally, the simulated train trajectories (with their current locations and accelerations) would feed into an optimisation programme, which would provide the optimal control commands to all trains across the network (or the route it commands). In the absence of such an optimisation tool, we aim to implement in TrackULA a set of reasonable control strategies (such as First In First Out (FIFO), giving priority to fast/long-distance/most delayed trains, etc.), and then test their performances under different network conditions. This is part of our on-going work (see also Section 7).

Already implemented in TrackULA are models of railway stations and a set of rules that command the movements of trains in/out of stations.

Similar to models of bus stops in DRACULA, a railway station modelled in TrackULA is described by its unique identification and the length of the platform. The platform length determines the type and the number of trains that can be stopped at each platform.

Trains stop at their scheduled stops. They decelerate upon approaching a scheduled stopping station (following the train driving cycle modelled in Section 5.5), and they stop for a pre-specified dwell time. The dwell times can vary by routes (see Section 5.3). The train accelerates away from the station when their scheduled departure times are due, or after they have stayed on the platform for the duration of the scheduled dwell time, whichever is later.

In the model, the trains' arrival times to a station are not pre-specified (unlike in the published timetable); they are determined by the simulated trajectories and travel time (i.e. an output of the simulation). The simulated arrival times can be compared with the scheduled ones as a way to test the feasibility of a timetable.

5.7 Simulation outputs

At its most detailed level, the TrackULA simulation records, for each individual train, their second-by-second locations and speeds. For ease of post-simulation analysis, the standard outputs are the averages and the distributions of system performance measures, measured for a user-specified output time interval, over different spatial coverages and by individual trains. These are listed below:

- Outputs by train routes:

The simulation outputs, for each train route, are the mean and standard deviation of total journey time over the entire route, between a pair of nodes and between stopping stations, and dwell time at stopping stations.

Similar statistics are collected by the simulation for the entire network.

- Outputs by links (section of line between two nodes):

The link outputs are the number of trains traversed through the link, and an average and a variance of their travel times. This allows analysis of the performance on individual sections of the network.

- Outputs by individual trains:

The individual outputs are the departure and arrival times at each station, and travel times on the links passed.

6. Example tests with TrackULA

The TrackULA model is developed as a tool to investigate the dynamics between train schedules and ERTMS control systems in a railway network. This section presents example simulation tests using TrackULA; the results and discussion are primarily intended to illustrate the applicability of the model and to show that the model responds logically to changes in model parameters.

The simulation tests are conducted on a single railway line of four stations illustrated in Fig. 7. The three sections of the lines are each of 20km long. There are 16 trains scheduled to traverse the line from A to D, with 3min headway, stopping at stations B and C for 1 min each.

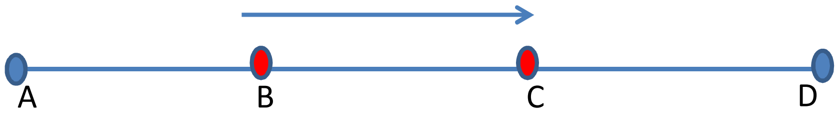


Figure 7: **The test network. Trains enter at A and exit at D, while stopping en-route at stations B and C.**

Two types of trains are modelled: a fast train and slow train. The model parameter values used in the simulation are listed in Table 4.

Table 4. Simulation parameter values

Parameter values (and model variable)	**Fast train**	**Slow train**
Train length (L)	250 m	75 m
Reaction time (T)	1 s	1 s
Maximum speed (V)	200 km/hr	120 km/hr
Safety distance headway	2000 m	1000 m
Optimal speed (V0)	200 km/hr	200 km/hr
Acceleration (A)	1.0 m/s/s	1.0 m/s/s
Deceleration (D)	-1.0 m/s/s	-1.0 m/s/s

Two test scenarios are conducted:
Scenario I: All fast trains; and
Scenario II: A mixture of fast and slow trains.

The simulated train trajectories are presented in Figs. 8 and 9. We compare the performances of the scenarios in terms of flow stability and total network travel times.

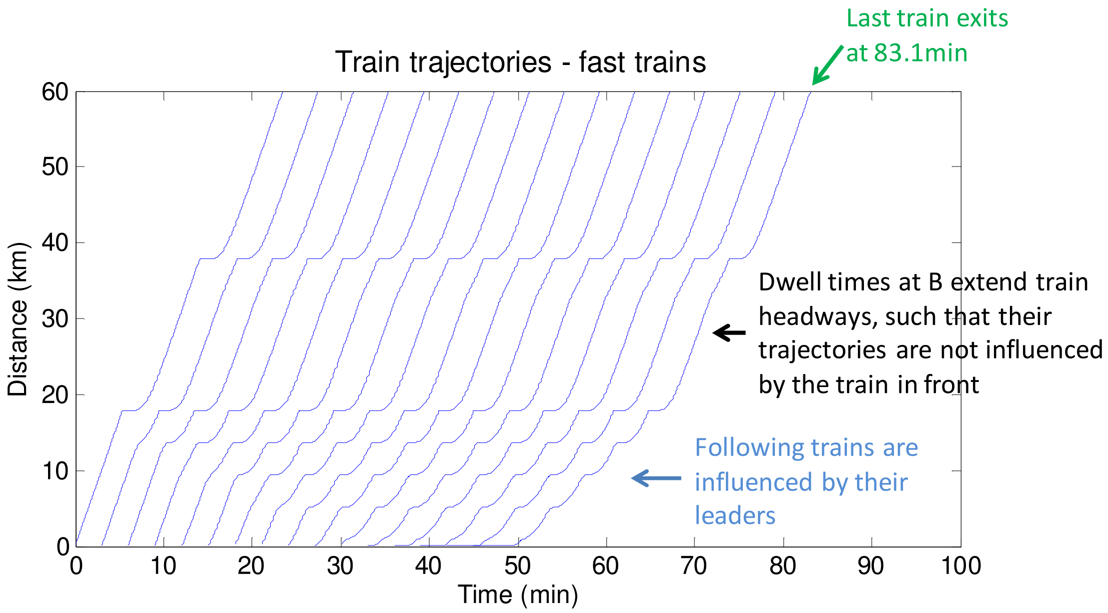
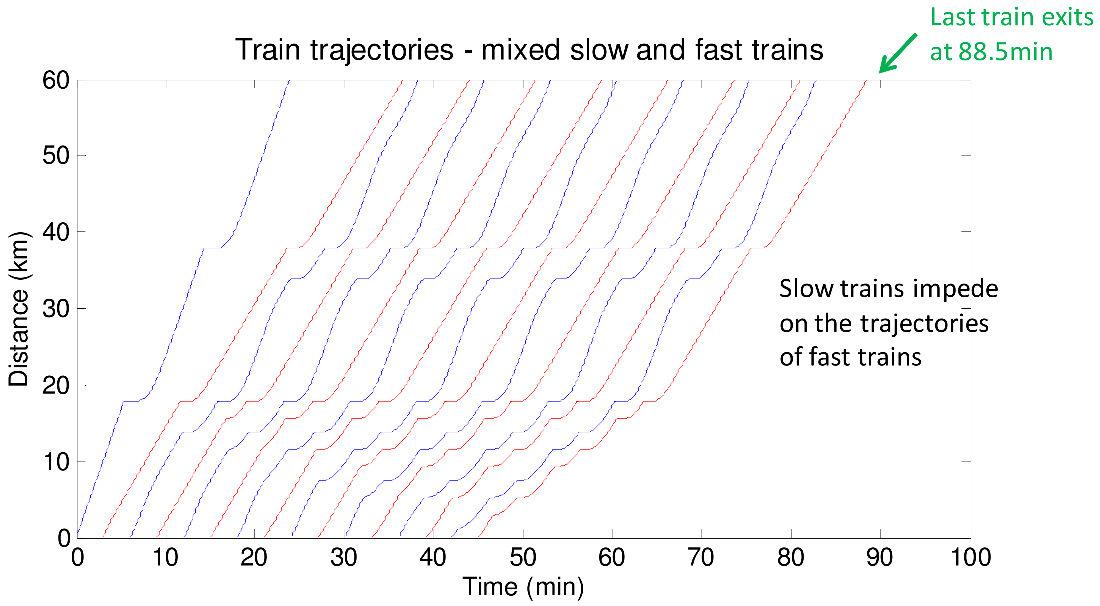


Figure 8: **Simulated train trajectories for scenario I.**

It can be seen in Figs. 8 that, over the section between nodes A and B, the following trains' trajectories show increasing degree of stop-and-start movements. This suggests that the scheduled time headway (of 3 min) may be too small, such that the following vehicles have to slow down to keep to the safety distance to the front train. This effect magnifies upstream over time: the later trains having to stop-and-go earlier in their journeys, while the last five trains having to delay their departure from the origin stations.

After stopping at station B for their scheduled 1min dwell time, the trains' spacing is spread out and all trains appear to move without being constrained by their preceding trains. The last train exits the network at 83.1 min from the start of the simulation (when the first train departed).

In scenario II, one fast train is followed by one slow train with the same 3min interval. Fig. 9 shows that the impediment of the slow trains on the movements of the fast trains is clearly present throughout the entire network. Compared to Fig. 8, however, the small 3min departure headway does not seem to have any significant impact. All trains have departed on time. The delays, and the deceleration-and-acceleration waves, of the following trains seem to be mainly affected by the speeds of the slow trains and the safety headway constraint.



Last train exits at 88.5min

Slow trains impede on the trajectories of fast trains

(b)

Figure 9: **Simulated train trajectories for scenario II. The blue lines are trajectories of fast trains, while the red lines the slow trains.**

7. Ongoing and Further Work

Further work is required to fully develop the algorithms for junction operations (train diverge, merge and crossing conflict situations) to ensure that they are fully robust. Work is also required on station control algorithms, addressing such issues as platform allocation and train dwell times.

Microsimulation models employ parameters to represent the detailed behaviour of the system. In TrackULA, realistic data is needed for a wide range of model parameters. Wherever possible, these parameter values are gathered from academic literature, industry contacts and reports and ERTMS technology specifications. In the absence of any known values, TrackULA follows conventional practice of conducting sensitivity tests of any model parameters, to identify the impacts of adopting different assumed values and to determine the robustness of model solutions to such values.

The formulation of the train-following model in eq. (1) represents a train's response to the train immediately in front of it. Recently, Chen and Liu (2015) formulated a multi-anticipative vehicle-following model, in which the following vehicle responds to more than one vehicle in front. They show that the stability of the multi-anticipative system is stronger. With ERTMS Level 3 technology, it is possible to encompass the multi-train following scenario in the traffic management strategies.

Once work in modelling ERTMS Level 3 is substantially complete, we will then move onto similar work for ERTMS Level 2.

References

Asuka M. and Komaya K. (1997) A combined approach of microscopic and macroscopic simulation for rail traffic. *Electr. Eng. Jpn.* 1997; 119: 61–68.

Banks, J. and Carson, J.S. (1984) *Discrete-Event System Simulation*. Prentice-Hall, New Jersey.

Barber, F., Abril, M., Salido, M.A., Ingolotti, L.P., Tormos, P. and Lova, A. (2007) *Survey of Automated Systems for Railway Management*. Available online from http://www.dsic.upv.es/docs/bib-dig/informes/etd-01152007-140458/AutomatedSystems.pdf [Accessed 22 September 2015].

Bendfeldt J-P, Mohr U and Muller L. (2000) RailSys: a system to plan future railway needs. In Allan J, Brebbia CA, Hill RJ, Sciutto G and Sone S, (eds) *Computers in Railways VII*, 7th International Conference on Computers in Railways. Southampton: WIT Press, pp. 249-255.

Chen, J. and Liu, R. (2015) Stability analysis of a multi-anticipative car-following model. In preparation.

Corman, F., and Quaglietta, E. (2015) Closing the loop in real-time railway control: Framework design and impacts on operations. *Transportation Research Part C*, 54, pp. 15-39.

Dorfman, M.J. and Medanic, J. (2004) Scheduling trains on a railway network using a discrete event model of railway traffic. *Transportation Research Part B* 38 (1), 81-98.

Hayat, S., (2013) Modeling of new driver assistance system for dysfunction of the signaling system ERTMS/ETCS. *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management* (IESM), pp. 1-7.

Huber H-P. and Wilfinger G. (2006) Integration-Enhancements for Microscopic and Macroscopic Railway Infrastructure Planning Models, *Proceedings of 7th World Congress on Railway Research*, 2006.

Jabri, S., El Koursi, E., Bourdeaud'huy, T. and Lemaire, E. (2010) European railway traffic management system validation using UML/Petri nets modelling strategy. *European Transport Research Review* 2 (2), 113-128.

Kettner M, Sewcyk B, and Eickmann C. (2003) Integrating microscopic and macroscopic models for railway network evaluation. *Proceedings of European Transport Conference*, Strasbourg, France, October 8-10, 2003.

Koutsopoulos, H. and Wang Z. (2009) An urban rail operations and control simulator: Implementation, calibration and application. In Chung E and Dumont A (eds) *Transport simulation: Beyond traditional approaches*. CRC Press, 153 – 169.

Law, A.M. and Kelton, D.W. (2000) *Simulation Modelling and Analysis.* 3rd ed., McGraw Hill Inc., New York.

Li, F., Gao, Z., Li, K. and Yang, L. (2008) Efficient scheduling of railway traffic based on global information of train. *Transportation Research Part B* 42 (10), 1008-1030.

Li, F., Sheu, J. and Gao, Z. (2014) Deadlock analysis, prevention and train optimal travel mechanism in single-track railway system. *Transportation Research Part B* 68, 385-414.

Liu, R. (2005) The DRACULA dynamic network microsimulation model. In: Kitamura, R. Kuwahara, M. (Eds), *Simulation Approaches in Transportation Analysis: Recent Advances and Challenges*, Springer, pp23-56. ISBN0-387-24108-6.

Liu, R. (2008) *Dynamic Route Assignment Combining User Learning and microsimulAtion*. University of Leeds. http://www.its.leeds.ac.uk/software/dracula/.

Liu R. (2010) Traffic Simulation with DRACULA. In, Barcelo J (ed) *Fundamentals of Traffic Simulation*, New York: Springer, pp. 295-322.

Liu R. Van Vliet D, and Watling D. (2006) Microsimulation models incorporating both demand and supply dynamics. *Transportation Research Part A-Pol*, 40(2), 125-150.

Liu, R, Whiteing, A E and Koh, A (2013) Challenging established rules for train control through a fault tolerance approach: Applications at a Classic Railway Junction. *Journal of Rail and Rapid Transit,* 227(6), 685-692.

Medanic, J. and Dorfman, M.J. (2002) Efficient scheduling of traffic on a railway line. *Journal of Optimization Theory and Applications* 115 (3), 587-602.

Mei, T. and Liu, R (2013) Challenging established rules for train control through a fault tolerance approach. Project Report to RSSB.

McGuire, M. and Linder, D. (1994) Train simulation on British Rail. In Murthy TKS, Mellit B, Brebbia CA, Sciutto G and Sone S, (eds) *Computers in Railways IV*, 4th International Conference on Computers in Railways. Boston, 1994, pp. 437-444.

Nash, A. and Huerlimann D. (2004) Railroad simulation using OpenTrack. In Allan J, Brebbia CA, Hill RJ, Sciutto G and Sone S, (eds) *Computers in Railways IX*, 9th International Conference on Computers in Railways, WIT Press, pp. 45-54.

Qiu, S., Sallak, M., Schon, W. and Cherfi-Boulanger, Z. (2014) Modeling of ERTMS level 2 as an SoS and evaluation of its dependability parameters using statecharts. *IEEE Systems Journal* 8 (4), 1169-1181.

Quaglietta, E. (2014) A simulation-based approach for the optimal design of signalling block layout in railway networks. *Simulation Modelling Practice and Theory*, 46, 4-24.

Radtke A. (2006) Timetable management and operational simulation: methodology and perspectives. In Allan J, Brebbia CA, Rumsey AF, Sciutto G, Sone S and Goodman CJ (eds) *Computers in Railways X*, 10th International Conference on Computers in Railways, Southampton: WIT Press, 2006, pp. 579-589.

Radtke A and Bendfeldt J. (2001) Handling of railway operation problems with RailSys. *Proceedings of 5th World Congress on Rail Research*. 2001.

Siefer T. Simulation. In: Hansen I and Pachl J (eds). *Railway Timetable and Traffic.* Hamburg: Eurail Press, 2008, pp. 155-169.

UNIFE (2014a) ERTMS Factsheet #1: From trucks to trains - How ERTMS helps making rail freight more competitive. Available at http://www.ertms.net/wp-content/uploads/2014/09/ERTMS_Factsheet_1_From_truck_to_train.pdf

UNIFE (2014b) ERTMS Factsheet #3: ERTMS Levels. Available at http://www.ertms.net/wp-content/uploads/2014/09/ERTMS_Factsheet_3_ERTMS_levels.pdf

UNIFE 2014c) ERTMS Factsheet #4: ERTMS Deployment in Italy. Available at http://www.ertms.net/wp-content/uploads/2014/09/ERTMS_Factsheet_4_ERTMS_deployment_Italy.pdf

UNIFE (2014d) ERTMS Factsheet #5: ERTMS Deployment in Spain. Available at http://www.ertms.net/wp-content/uploads/2014/09/ERTMS_Factsheet_5_ERTMS_deployment_Spain.pdf

UNIFE (2014e) ERTMS Factsheet #6: ERTMS Deployment in Switzerland. Available at http://www.ertms.net/wp-content/uploads/2014/09/ERTMS_Factsheet_6_ERTMS_deployment_Switzerland.pdf

UNIFE (2014f) ERTMS Factsheet #7: ERTMS Deployment outside Europe - ERTMS as a global standard. Available at http://www.ertms.net/wp-content/uploads/2014/09/ERTMS_Factsheet_7_ERTMS_deployment_outside_Europe.pdf

UNIFE (2014g) ERTMS Factsheet #9: A unique signalling system for Europe. Available at http://www.ertms.net/wp-content/uploads/2014/09/ERTMS_Factsheet_9_A_unique_signalling_system_for_Europe.pdf

UNIFE (2014h) ERTMS Factsheet #12: ERTMS Deployment in Belgium & The Netherlands. Available at http://www.ertms.net/wp-content/uploads/2014/09/ERTMS_Factsheet_12_ERTMS_deployment_in_Belgium_and_Netherlands.pdf

Watson R. (2005) Using stochastic simulation to predict timetable performance - status and developments in the UK. In *Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis*.