SUMO 2016 – Traffic, Mobility, and Logistics

Proceedings

Deutsches Zentrum
für Luft- und Raumfahrt

Berichte aus dem DLR-Institut für Verkehrssystemtechnik

Volume 30

Proceedings of the
SUMO2016

Traffic, Mobility, and Logistics

May 23 − 25, 2016
Berlin-Adlershof

Preface

Dear reader,

You are holding in your hands a volume of the series „Reports of the DLR-Institute of Transportation Systems". We are publishing in this series fascinating, scientific topics from the Institute of Transportation Systems of the German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt e.V. – DLR) and from his environment. We are providing libraries with a part of the circulation. Outstanding scientific contributions and dissertations are here published as well as projects reports and proceedings of conferences in our house with different contributors from science, economy and politics.

With this series we are pursuing the objective to enable a broad access to scientific works and results. We are using the series as well as to promote practically young researchers by the publication of the dissertation of our staff and external doctoral candidates, too. Publications are important milestones on the academic career path. With the series „Reports of the DLR-Institute of Transportation Systems / Berichte aus dem DLR-Institut für Verkehrssystemtechnik" we are widening the spectrum of possible publications with a building block. Beyond that we understand the communication of our scientific fields of research as a contribution to the national and international research landscape in the fields of automotive, railway systems and traffic management.

With this volume we publish the proceedings of the SUMO Conference 2016 which was held from 23rd to 25th May 2016 with a focus on traffic, mobility, and logistics. SUMO is an open source tool for traffic simulation that provides a wide range of traffic planning and simulation functionalities.The conference proceedings offer an overview of the applicability of the SUMO tool suite as well as its universal extensibility due to the availability of the source code. The major topic of this fourth edition of the SUMO conference are the different facets of moving objects occurring as personal mobility and freight delivery as well as communicating networks of intelligent vehicles. Several articles cover heterogeneous traffic networks, junction control and new traffic model extensions to the simulation. Subsequent specialized issues such as disaster management aspects and applying agile development techniques to scenario building are targeted as well. At the conference the international user community exchanged their experiences in using SUMO. With this volume we provide an insight to these experiences as inspiration for further projects with the SUMO suite.

Prof. Dr.-Ing. Karsten Lemmer

Contents

1 Bicycle modelling in SUMO for accurate traffic simulation

Robbin Blokpoel and Mahtab Joueiai; Dynniq
{Robbin.Blokpoel, Mahtab.Joueiai}@Dynniq.com

Abstract

Microscopic modelling of bicycle flow is important to design the infrastructure that is used by cyclists. The safety of cyclists and level of service of the available infrastructure can only be assessed with efficient knowledge of bicycle traffic. SUMO allows modelling the bicycle traffic either as slow cars or as fast pedestrians. In either case the intersection topology considers bicycle lanes the same as vehicle lanes. Therefore, bike lanes end at the junction border like for vehicle lanes, while pedestrians also have to cross the bicycle lane and a possible green verge through the traffic light. However, many intersections have a topology where the bicycle lanes follow the same principles as pedestrians and both only cross the vehicle lanes through a traffic light. This results in shorter crossings and has a significant impact on simulation results. This paper describes principles how to modify SUMO's netconvert for the latter topology.

Keywords: Bicycle, Safety, Modelling, Moving Bottlenecks

1.1 Introduction

With the recently released pedestrian modelling functionalities in SUMO (2), the path is open for full simulation of all urban modalities. For bicycles there are different options for modelling, either as a slow car or as a fast pedestrian according to SUMO's guideline.

In other traffic simulators bicycles can be modelled similar to cars. Vissim for example, has the possibility to define multi-class driving behaviour. The car following model encounters a different set of parameters for each vehicle class. Bicycles can be modelled as slow vehicles with a minimum speed of 1 km/h. Also, lanes can be used for mixed modalities which require a set of rules on how different transportation modes overtake each other. In Vissim, the user is completely free on where to place connectors and to shape the topology of network. However, in order to model bicycles in SUMO, more work is required to modify the automatically created networks with "netconvert".

Modelling cyclists as a slow class of vehicles in SUMO, necessitates to tune and calibrate the car following variables and parameters according to *fundamental diagram* of bicycle flow.

In literature there has not been a lot of research into bicycles as compared to other modalities. There are some attempts to define the fundamental diagram for bicycle flow (e.g. (11), (9), (3), (13)). Critical density, jam density and maximum lane capacity that are indicated in these studies could be applied by simulators to calibrate the car following parameters for the purpose of bicycle flow simulation.

The width of a bicycle lane has a high impact on its maximum capacity as demonstrated in (10). For example at $1.0m$ lane width the capacity is around 1200 bicycles per hour and for $3.0m$ capacity increases to 3200.

Following headway of bicycles also has impact on flow rate and capacity of lanes. The distribution of headways were investigated in a field trial in (4) and can be used in defining the car following parameters in SUMO. In SUMO's netconvert these variables and parameters can be combined to determine the amount of lanes and the car following parameters to best model the lane capacity of bicycle path and lane width.

SUMO has also the possibility to model the cyclists as faster pedestrians. Modelling the bicycles as pedestrians has the advantage that multiple bicycles can wait for a traffic light while standing next to each other, which they do in reality as well. However, a larger area is occupied by a bike which should be taken into account. Another consideration is the intersection topology, currently when specifying networks bike lanes end at the junction border like for vehicle lanes as is shown in figure 1.1.
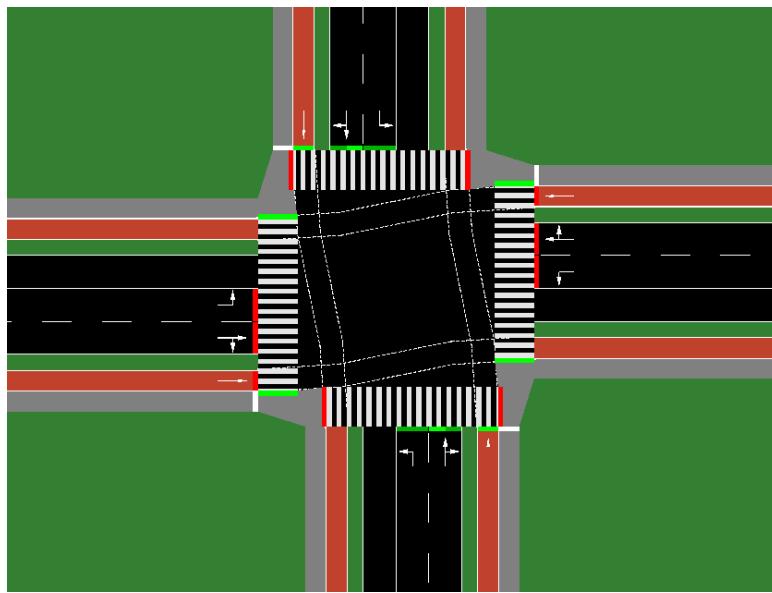
Figure 1.1: Traditional intersection layout in SUMO

However, like stated in the SUMO wiki, this does not enable the possibility to implement a bicycle left turn as two straight crossings in a row. Figure 1.2 shows a typical layout for the standard intersection topology in the Netherlands. Bicycle lanes are build up from the right turns, it can be seen as one continuous lane for example going from the east to north. At the beginning of the curve turning to the north, the exit of the crossing with traffic from the south merges with the lane. Near the end of the turn a lane leading to a crossing heading west branches off.

Conflicting bicycle and pedestrian streams are not controlled by the traffic light, right of way rules are sometimes shown on the road surface, but not always. By default traffic from the right has right of way and bicycles have priority over pedestrians. With larger traffic volume it can be interesting to prioritize traffic leaving the intersection to prevent non-safe spillback on the vehicle lanes.

The differences between figure 1.1 and figure 1.2 are not just visual but also have an impact on the simulation results. Pedestrian crossings will be shorter for not having to cross the vehicle lanes with a traffic light and the bicycle crossings will also be much shorter. This has an impact on the clearance time of the intersection and where the partial conflict between vehicles and pedestrians/bicycles occur.

The paper will first discuss a possible solution how to add a new option to netconvert which makes the generation of networks like figure 1.2 possible in SUMO. The number of lanes and capacity modelling also has a large impact on the intersection modelling, which is discussed in a separate section. This is followed by a section on impact of the topological changes. The paper finishes with conclusions and further research.

Figure 1.2: Satellite image of a real world intersection in Soest, the Netherlands

1.2 Netconvert Extensions

Before netconvert creates the simulation network, the network has to be defined. In this work it is assumed that the user has specified a network with bicycle lanes, green verges and pedestrians before using netconvert. Most likely this was done with node and edge files, but conversions from other network file formats should result in a similar topology.

Using netconvert, there are many different options to customize the conversion process to the specific situation of the network. Similarly, the proposed extensions should be user-configurable and therefore adding the option "–crossings.beyond-bicyles" is proposed.

When using the new option, the first change applied is to let pedestrians only cross vehicle lanes. Something similar was already implemented in the NBEdge.cpp class with a method to find the first non-pedestrian lane. A new method was added to find the first lane that is not for pedestrians, bicycles, pedestrians and bicycles or a green verge. If this method is used for the network of figure 1.1 it results in a crossing like shown in figure 1.3.

Figure 1.3: Result of adaptation to the crossing computation

From this picture the drawing layer order of GUIsim also clearly shows. The internal edges are completely on the background, with a correct node shape they are not visible. The second layer

has the node shape, which is a solid black area. On the foreground, the walking areas stand in grey colour. These area make internal edges and the node shape underneath invisible. This also identifies that bicycle lanes internal to the intersection should also be on top of the node shape as they are actually outside of the real intersection space and usually have an orange colour.

For the bicycle lane, the following changes are required: traffic light next to the pedestrian crossing, uncontrolled intersection with conflicting bicycles and pedestrians. To explain this in more detail consider the schematic overview in figure 1.4.
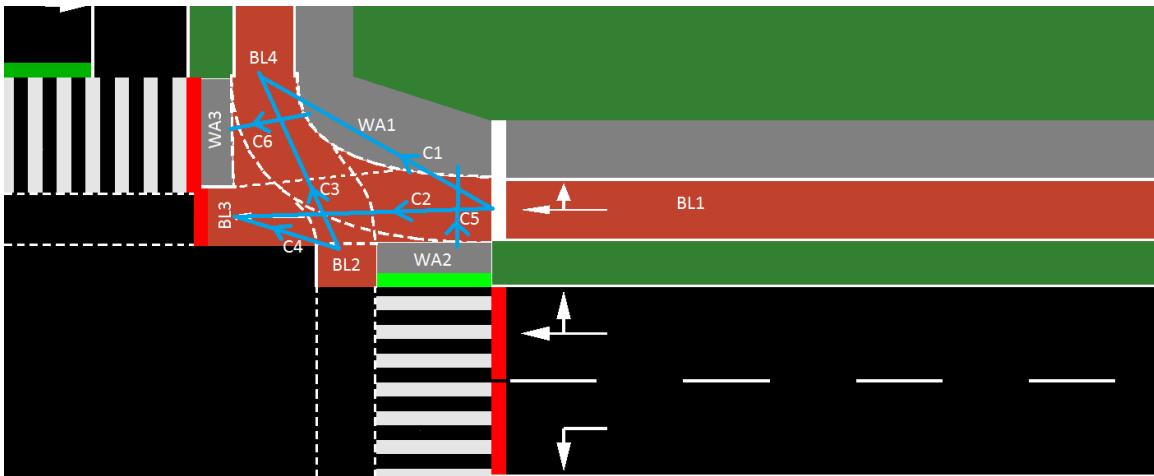
Figure 1.4: Schematic overview of bicycle crossing changes

Bicycles arrive on *BL1* which is part of the same edge as arriving vehicles and pedestrians. This lane connects to *BL3* to cross in westward direction and *BL4* to continue in a northward direction. Bicycles who just completed the crossing from the south arrive on *BL2* and can continue the second part of a left turn by using the internal edge of connector *C4* or continue northward with *C3*. The original right-turn of *C1* remains unchanged, while *C2* is the first part of the original connector for straight traffic. The walking area, which became very large in figure 1.3, is now divided in three separate areas. *WA1* connects pedestrians wanting to make a right turn. Pedestrians from the south arrive on *WA2*, have to cross the bicycle lanes via *C5* and can continue on *WA1*. Connector *C6* is required to continue in a westward direction.

To realize this topology netconvert should cut the internal lanes of each through direction for bicycle traffic in five parts. Two of these are normal edges, the first is just before the vehicle lane (*BL3*) and the second just after it (*BL2*), both have the same length as the width of the green verge. This implies that the edge input file for netconvert should have green verges for the "–crossings.beyond.bicycles" option to work. The walking areas *WA2* and *WA3* also have a length equal to the width of the green verge. With a non-existent green verge this will not be possible.

The connectors depicted in figure 1.4 represent the internal edges of a new uncontrolled intersection and are not part of the old controlled intersection. Each corner of the intersection will have a new uncontrolled intersection like that. The incoming lanes of the old controlled intersection do not need to be changed, but the internal lanes only contain the controlled crossings over the vehicle lanes. For example the east-west bicycle route will be made as follows: *BL1*, *C2*, *BL3*, crossing over vehicle lanes (controlled), new exit edge, connector to edge west of the intersection, edge west of the intersection. Following this example and the structure shown in figure 1.4, all connections and crossings in the net file are adapted to follow the new structure.

Regarding right of way rules there are many possibilities. By default netconvert should result in the safest and most common priorities. Therefore, the connections coming from *BL2* should be prioritized over the ones coming from *BL1*. All bicycle links should be prioritized over pedestrians. Table 1.1 shows the resulting right of way matrix like figure 21-3 of (6). The rows represent the

considered bicycle/pedestrian, while the columns represent an approaching potentially conflicting bicycle/pedestrian. So a bicycle approaching from *C1* has to stop for a bicycle on *C3* (red cell), while there is a conflict with a pedestrian going to *C5*, but it does not have to stop for it (yellow cell). The connections *C1* and *C4* may conflict depending on the curve radius, but here it is assumed in reality bicycles will solve this conflict by adapting their lateral positions on the lane.

Table 1.1: Right of way rules for uncontrolled bicycle-pedestrian intersection

Approaching connector	C1	C2	C3	C4	C5	C6
C1			🟥		🟨	🟨
C2			🟥	🟥	🟨	
C3	🟨	🟨				🟨
C4		🟨				🟨
C5	🟥	🟥				
C6	🟥		🟥			

Table 1.2: Legend

Stop	🟥
Right of way	🟨

1.3 Capacity Modelling

In the introduction it is already mentioned that bicycles can be modelled as slow vehicles or fast pedestrians according to SUMO's guideline. The focus of this section is on modelling cyclists as slow vehicles. Various bicycle models based on car-following models have been proposed in the literature(e.g. (4), (5), (1)). Without specification on the type ofthe car-following model that can best represent the bicycle flow, the research of modelling bikes as slow vehicles indicates that there is a good possibility for validating the cyclists behaviour in SUMO.

For netconvert, the work of (10) is very interesting when the user only specifies one bicycle lane with a certain width. Netconvert can then derive from 1.5 how many bicycles can cycle next to each other for the specified lane width.

A simple approximation would be to assume one lane of $1200\ b/hour$ is present for lanes $< 1.5m$, 2 lanes with a total of $2400\ b/hour$ for $1.5m < width < 2.8m$ and so on. However, as can be seen in the figure, the capacity increases gradually with increasing lane width. So for example with $1.5m$ not all bicycles cycle next to each other, but a considerable amount will do this, leading to $500\ b/hour$ extra capacity when compared to a $0.8m$ lane. This can be solved by modifying the car following distance on the second lane to have a lower capacity on it. This would lead to specific bicycle rules regarding the car following model, which take the lane width into account.

Another factor to take into account is spillback. It is possible that right turning bicycles may get blocked by a long queue for crossing the intersection. In reality bicycles take the footpath in such cases (and formally have to dismount their bicycle). This can be modelled by letting the bicycle change modality and use the footpath in such cases, but this will complicate the model. Another spillback possibility is for conflicting pedestrians or bicycles wanting to cross the queue. Unlike for vehicles, bicycles usually manage to reorganize the space to create a little corridor for crossing traffic.
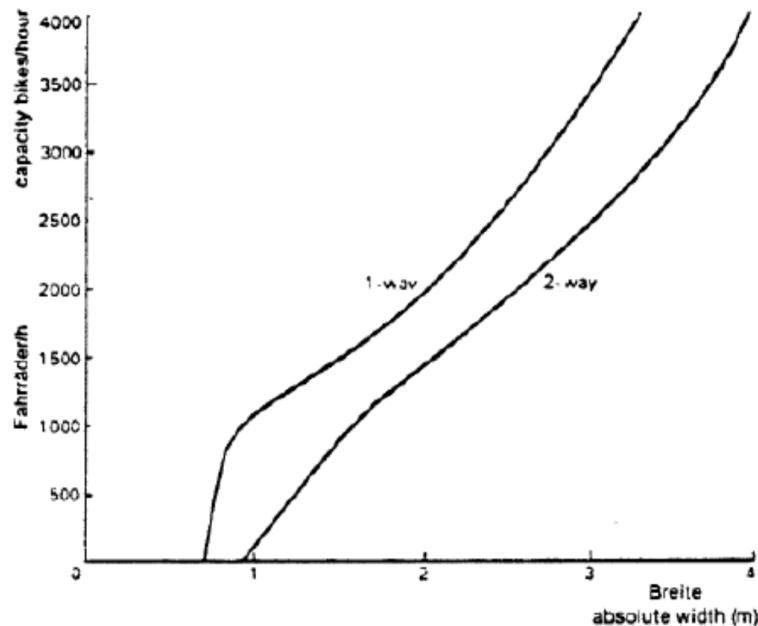
Figure 1.5: Capacity versus width, taken from (10)

Again this can be modelled as if the crossing bicycle changes to the pedestrian modality and slowly finds its way through the spillback.

A last factor to take into account is the jam density in b/m^2, which is most important for the areas close to an intersection. According to the literature, jam density has variation between 0.27 to 0.45 $b/m2$ [(9), (3), (13), (12)].

The variation of jam density could be explained as follows. Unlike cars bicycles stay closer to each other when standing at a signalized intersection. Not only the longitudinal distance between cyclists is much less than the cars but also their lateral distance could be quite small. In that respect the bicycle operation follows the pedestrian behaviour rather than vehicles. This phenomenon results to a much higher jam density specially near the intersections. Likely, bicycles start forming an extra lane when a jam occurs like can be seen in figure 1.6.

Figure 1.6: Bicycle queue at traffic light.

The width of the lane in the figure is not enough to cycle comfortably with three cyclists next to each other. However, when waiting at a traffic light, the density increases as there is enough space

to stand still with three cyclists. Netconvert can take this into account by using two approaches. In the next two subsections the proposed approaches to correctly model the bicycle jam density at the intersections are explained.

Creating Extra Lanes

One way to encounter a high jam density of cyclists at the intersections is to create an extra lane for all internal bicycle lanes in Netconvert. Extra lanes should also be added to the new edges required for creating the uncontrolled intersections at the corners of the original intersection.
Note that the capacity of the bicycle path should not be increased due to addition of extra lanes. These lanes are used by cyclists at the intersection to form lines beside each other and as a result have a more realistic queue length. Therefore, the lanes have to merge to their original number closely after the intersection.

Introducing Moving Bottlenecks

Adding an extra lane to the bicycle infrastructure in Netconvert results in increased total capacity for the bike path. To compensate for this effect, it is proposed to introduce a "dummy" slow bicycle to the added lane which is acting as the bottleneck while it moves.
This "dummy" bike that is riding significantly slower than the average speed of the bike flow, occupies the added lane as it moves. The other cyclists have to overtake it and use the remaining capacity of the path. As a result, this slow bike is acting as a *moving bottleneck* for the bicycle flow.
At the signalized intersection, the "dummy" bike also stops with the rest of the cyclists and the bottleneck is becoming *inactive*. The added lane in this situation could be used by cyclists to form a queue behind the "dummy" bike. The total jam density at the intersection increases whereas the capacity of bicycle path remains unchanged. The total jam density is resulted from addition of jam density of each lane so that the overal jam density of the bicycle path is increased due to added lane. Apart from that the speed of the moving bottleneck can also be adjusted to gradually control the capacity of the extra bicycle lane. With this method the capacity of 1.5 can be approximated more precisely for any lane width.
To know how the moving bottleneck is decreasing the link capacity (7), consider the relation between flow and density in bicycles (3).

$$k = \frac{q}{c.u.w} \tag{1.1}$$

Where k is density[b/km²],
q is flow[b/h],
c is a constant that is 1.00/m if the speed is provided in kilometre/hour,
u is the average speed,
w is the lane width.
Suppose that there is a 3 lane bicycle path like in figure 1.7.a. Bicycles can move with the free flow speed v_f on this path. The fundamental diagram of the bicycle traffic on this path could be like figure 1.7.b.
A bike is introduced on the right-most lane with a speed of (v_b), which is significantly slower than the average speed. Faster bikes have to overtake the "dummy" slow bike and they can no longer use the whole capacity of the road. The capacity of the road around this "dummy" bike is less than the total capacity and corresponds to point U in the figure 1.7.b.
The bikes upstream of the "dummy" bike have to slow down to merge into another lane and overtake it and have the speed v_u. This "dummy" bicycle is then acting as a bottleneck for the traffic flow
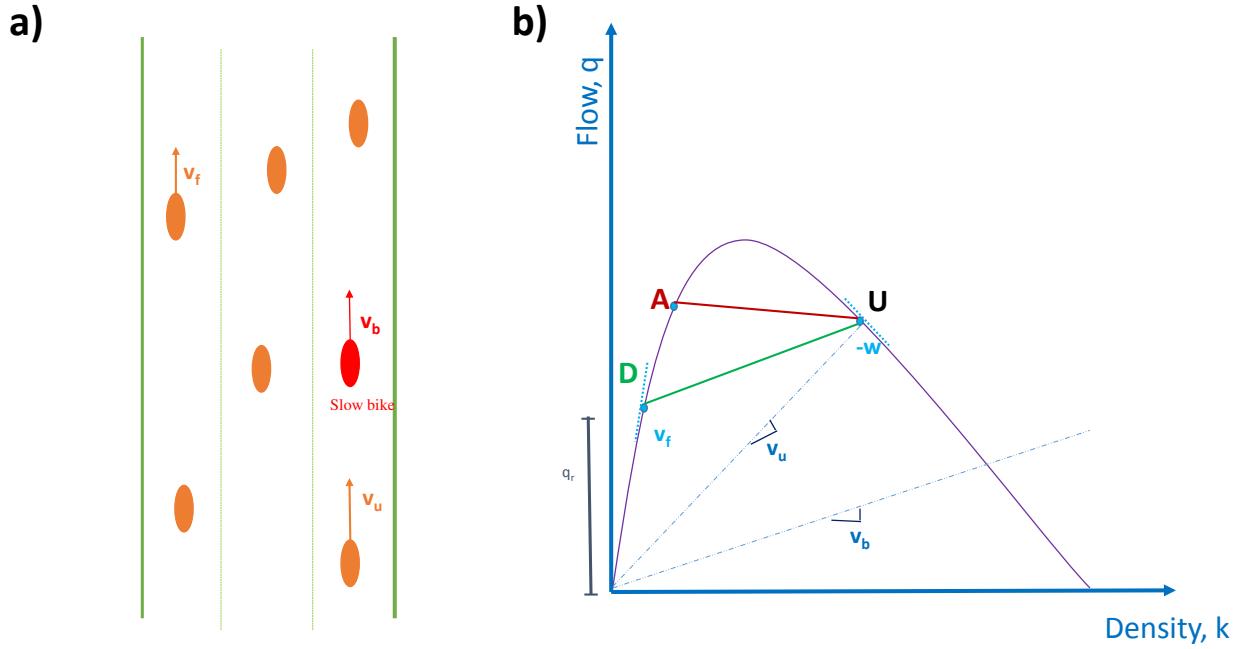
Figure 1.7: Fundamental diagram and moving bottleneck

around it. Such a slow bicycle is known as *moving bottleneck* and can reduce the capacity of the road as it moves.

Bikes that are far upstream of the "dummy" bike are in state A and can move with free flow speed v_f. Downstream of the "dummy" bike, the cyclists can again move with free flow speed. This state corresponds to point D. The density downstream of the moving bottleneck is always lower than the density of upstream of it due to the reduced capacity.

Note that the highest possible unqueued flow that can be achieved when the bottleneck exist is at point D. This point is generally known as *bottleneck capacity* and can be used in SUMO to accurately model the bike road capacity. If the bottleneck is not active the downstream and upstream condition have to coincide with the $k - q$ diagram.

As an observer that is moving with the speed of the "dummy" bike (v_b), it can count the bikes that overtake it. The passing rate (i.e. the number of overtaking bikes at time unit) can be calculated according to (8):

$$q_r = q - k.v_b.w \tag{1.2}$$

This passing rate could be used to define the capacity of the road around moving bottleneck in SUMO. The interval of introducing the "dummy" slow bike in the extra lane must keep the passing rate at around the designed road capacity.

1.4 Impact of changes

To estimate the impact of different intersection topologies, three simulation scenario's were defined. The first has the standard topology like defined in 1.1, the second only has the pedestrians crossing beyond the bicycle lane and green verge as shown in 1.3. This has the advantage that the pedestrians can cross faster and the right turn for bicycles is uncontrolled. However, this uncontrolled right turn can only be accessed by the first bicycle in the queue. The last scenario has the full adaptations

like shown in 1.4. Here bicycles can also cross faster and the right turn is accessible as long as the queue does not reach beyond the length of *BL3* and *C2* combined. The bicycle flow was set at 180 b/h for the right turn and 180 b/h for the straight movement crossing the intersection. Vehicles have a demand of 72 veh/h for both the straight and the right turn. With 26 seconds of green time in a cycle of 90 seconds, these signal groups are well below saturation. The pedestrian demand is not significant with 30 per hour. The results for the traffic directly affected by the topology change is shown in 1.3.

Table 1.3: Results of average delay time for all traffic affected by topology change.

scenario	Bicycle right	Bicycle straight	Vehicles right	Vehicles straight
Standard	30.2s	26.9s	32.9s	25.8s
Pedestrians small	19.7s	28.2s	33.6s	26.0s
Ped. and Bicycles small	5.9s	26.9s	26.0s	25.1s

Clearly, bicycles turning right have an advantage with the uncontrolled right turn, especially when the internal edge is easily accessible like in the third scenario. The through direction for bicycles is not significantly affected, which can be expected as the traffic control plan did not change. More interesting however, is the impact on the right turning vehicles. They have to yield for crossing bicycles and are therefore positively affected by their shorter crossing area. This could be observed very well at the start of a green phase, bicycles effectively get a headway for crossing and vehicles have to wait less for them. This resulted in a 27% decrease of delay time for right turning vehicles from 32.9 to 26.0 seconds. Straight going vehicles were also slightly affected due to less spillback from vehicles waiting to turn right.

1.5 Conclusion and Recommendations

In this paper, the possibilities of modelling the bicycle operations in SUMO is investigated. In particular the modification of Netconvert to serve the purpose of realistic infrastructure modelling is emphasised. In this respect, the intersection topology is modified to match the real network, by adding an option to only cross vehicle lanes with the traffic light. The simulation result of bicycle model with such modification showed a 27% shorter delay for vehicles due to the smaller area of conflict between cyclists and cars. The paper provided descriptions of the principles for the network modification, which should not be too complicated to implement in netconvert and demonstrated the significance of having an accurate model for the delay time of vehicles.

Another modification that is proposed in this paper is balancing the lane capacity and jam density at the controlled intersections. The jam headway of cyclists (both longitudinal and lateral) at signalized intersections are much smaller than for cars. The proposed method is considering adding extra lanes to the bicycle path to allow cyclists form lines at the intersections. To keep the total capacity of bicycle lanes unchanged a slow bicycle as a moving bottleneck is introduced. This slow bike is inactive at the intersections as it stops and allows for a high jam density due to a formed line of bicycles behind it.

The paper has shown two elements to improve capacity and jam density modelling. However, more research is required to calibrate the addition of extra lanes, the moving bottleneck and the car following model into a consistent complete bicycle model for SUMO. Additionally, the implementation into netconvert and SUMO is not completely clear yet and needs further work.

1.6 References

[1] BOTMA, HEIN and HANS PAPENDRECHT: *Traffic operation of bicycle traffic.* Transportation Research Record, (1320), 1991.

[2] ERDMANN, JAKOB and DANIEL KRAJZEWICZ: *Modelling Pedestrian Dynamics in SUMO*.

[3] HOMBURGER, WOLFGANG S: *Capacity of Bus Routes, and of Pedestrian and Bicycle Facilities.* Institute of Transportation Studies, University of California, 1976.

[4] HOOGENDOORN, SERGE and WINNIE DAAMEN: *Bicycle Headway Modeling and Its Applications.* In *Transportation Research Board 95th Annual Meeting*, number 16-2088, 2016.

[5] KNOFLACHER, HERMANN: *Fußgeher-und Fahrradverkehr: Planungsprinzipien.* Böhlau Verlag Wien, 1995.

[6] KRAJZEWICZ, DANIEL and JAKOB ERDMANN: *Road intersection model in SUMO.* In *1st SUMO User Conference-SUMO 2013*, volume 21, pages 212–220. DLR, 2013.

[7] LAVAL, JORGE: *Stochastic processes of moving bottlenecks: Approximate formulas for highway capacity.* Transportation Research Record: Journal of the Transportation Research Board, (1988):86–91, 2006.

[8] MUNOZ, JUAN CARLOS and CARLOS F DAGANZO: *Moving bottlenecks: a theory grounded on experimental observation.* University of California Transportation Center, 2004.

[9] NAVIN, FRANCIS PD: *Bicycle traffic flow characteristics: experimental results and comparisons.* ITE journal, 64(3):31–37, 1994.

[10] RICHARD, HEINRICH: *Handbuch für Radverkehrsanlagen und Radverkehr.* 1981.

[11] SCHACHT, HANS-JOACHIM: *Der Radfahrweg:(ein Beitrag zur Lösung des Radfahrverkehrproblems).* Keyser, 1934.

[12] YANG, JUN-MENG: *Bicycle traffic in China.* Transportation Quarterly, 39(HS-039 147), 1985.

[13] ZAMANOV, MS: *Development of a prediction model for speed limit violations on tangent road sections.* PhD thesis, TU Delft, Delft University of Technology, 2012.

2 Calibrating a Motorway Segment in SUMO Using Single Vehicle Data

Mirko Barthauer and Bernhard Friedrich
Technische Universität Braunschweig, Institute of Transportation and Urban Engineering
{m.barthauer, friedrich}@tu-braunschweig.de

Abstract

In the field of microscopic traffic flow simulation, the calibration and the validation of the model are important steps before it can be used for decision support and evaluation of measures. Simulation of Urban Mobility (SUMO) supports a number of model parameters to approximate the real vehicles' driving behaviour. As we look forward to use SUMO for traffic simulation, we develop a calibration method using single vehicle data.

Over the last few decades, more and more motorways have been equipped with loop detectors for use in traffic management systems. They record the speed, length and gap of the vehicles passing by and guess the vehicle type in a continuous way. A big sample of empirical single vehicle data from the German A2 motorway west of Braunschweig is used for calibration and validation.

First, the SUMO car following parameters are estimated taking into account the traffic characteristics given by the detector data. All parameters are described by distributions from which individual vehicles can be drawn. During the calibration, further changes in parameters are made to approximate the field measurements. The resulting gap distribution of the simulation is compared with the respective empiric one by means of an error metric. For validation, the speed distributions are compared in a similar way. The simulation cannot emulate the real gaps and speeds exactly. Some differences may be related to lane changes which have not been addressed yet.

Keywords: single vehicle data, motorway, calibration, validation

2.1 Introduction

Models are thought of as a replacement for real systems. In the field of traffic planning and traffic control, many questions cannot be answered directly by implementing them in the real world. Instead, a big number of experiments can be modeled in microscopic traffic simulations which reproduce single traffic participants. However, almost any model uses a number of parameters to adapt to different real world situations. In the past, a German research project (see (8)) has been conducted to gain insight into simulating motorway traffic in concordance with the German Capacity Manual ((2)). Here, we use the microscopic traffic simulation SUMO ((12)) to model a short German motorway segment including a calibration and validation process. Non-aggregated so-called single vehicle data measurements from some induction loop sections along the motorway have been available from a previous project. These include a timestamp, gap, local speed, lane and the deducted vehicle type and length. They are used as simulation input and for comparison during the calibration and validation.

First, the example site is introduced by giving an overview of the road network and the traffic characteristics identified with use of single vehicle data. For example, we have a closer look at the gap, speed as well as the lane flow distributions. Afterwards, the SUMO simulation is initiated by means of the traffic characteristics and some additional assumptions. Here, the desired speed distribution of the simulated vehicles is estimated by the real speed in approximately free-flow conditions. In the main part, a calibration and validation methodology is developed and applied to the SUMO example. This includes the number of replications to simulate due to stochastics and a specific target function.

2.2 Traffic Characteristics on Motorways

2.2.1 Single Vehicle Data

Many motorway stretches in Germany and other countries have been equipped with traffic management systems. They are used for dynamic speed limits, overtaking prohibition and to display warnings to the driver in order to avoid accidents and alleviate congestion. Besides other data input, the automatic control receives traffic statistics from several sections along the motorway. One section may consist out of a double induction loop per lane and continuously records every passing vehicle. Induction loops register a change in their magnetic field when a metallic item passes by. According to its inductance time curve (see (3, p. 15)), several characteristics related to individual vehicles can be deducted. We refer to them as single vehicle data:

- The shape of the curve indicates the type of vehicle.

- The time between the inductance amplitude of two subsequent induction loops indicates the vehicle speed.

- The time between the inductance amplitude of one induction loop indicates the headway between the vehicles.

- The lane the induction loop is located.

Further aggregations of this single vehicle data lead to speed, vehicle type, lane flow, and gap distributions as well as to the total vehicle volume.

2.2.2 Example Site

The motorway A2 runs from the west of Germany to Berlin passing by Hannover and Braunschweig. Positions along both directions of the motorway are indicated by a chainage based on a reference point at km 0 close to Berlin. Our example site is situated on the eastbound direction between Peine and Braunschweig starting at chainage km 181.250. It consists of three lanes and runs straightly through mostly flat countryside. Figure 2.1 shows a schema of the motorway including some important places. The start is determined by an induction loop set for single vehicle data. This way, we can transfer the counted vehicle volume directly to SUMO input flows. Some kilometers downstream, we use another induction loop set for the calibration and validation process. In most cases, dynamic overhead speed signs are positioned close by. Besides that, there is one exit which is not further used in the simulation. It is assumed that traffic flows entering or exiting the motorway are small enough to be ignored.
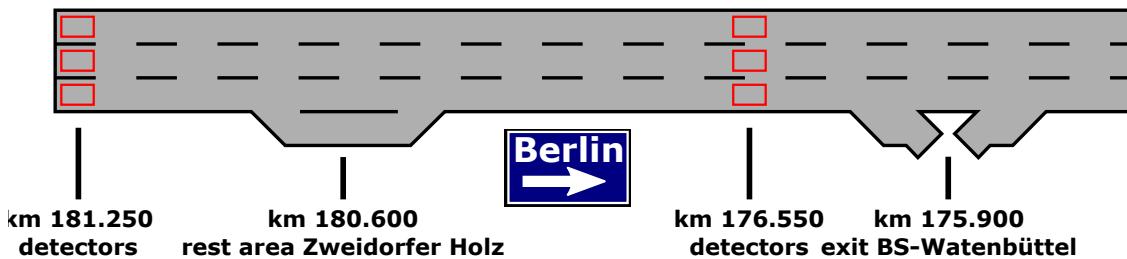
Figure 2.1: motorway schema (not to scale)

2.2.3 Evaluation of Characteristics

By targeting working day morning traffic, we choose a demanding scenario for modeling experience and for traffic management. According to the German survey guidelines ((4)), one should select a time period without intersecting school holidays nor any other singular events which usually have an effect on traffic. In our chosen period of September 2012, school holidays in most regions of Germany were over and no relevant singular event or weather condition was noted. Further on, we offer an overview of the traffic characteristics at the section MQ_A2_B_176,550 (see km 176.550 in Figure 2.1) situated along our field study segment. Later, part of this data will be compared with the simulation results.

To give a short overview of the analyzed time period, the volume/time graph (Figure 2.2) on a week day base is shown. During the working week (Monday to Friday), the traffic volume reaches a peak around 08:00 with 2700 to 3100 vehicles per hour. At the weekend, the traffic volume is lower and does not have a morning peak. Overall, there have been no exceptional peaks during the analyzed time period. With respect to (2, p.3-11) which assumes a capacity of around 5400 vehicles per hour for this kind of motorway, the capacity is not reached on an average workday morning. While the middle lane (ÜFS1) is preferred on working day mornings, the remaining are well used, too.
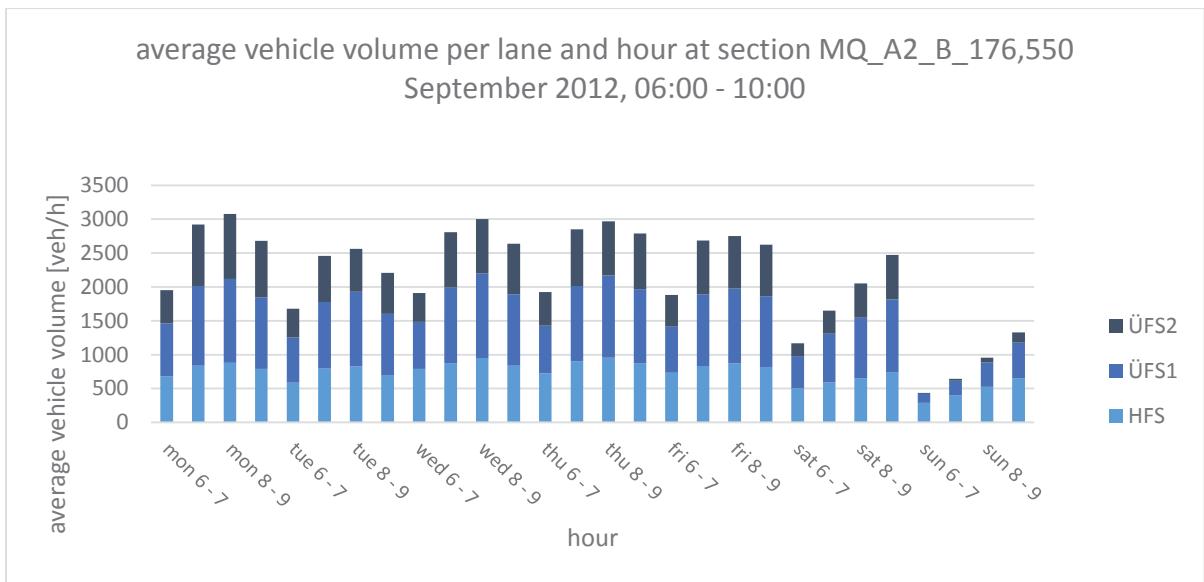
Figure 2.2: average vehicle volume

But the allowed lanes also depend on the vehicle type. Due to German legislation ((5, §2)), the inner lanes shall be used for overtaking. Heavy vehicles including (trailer) trucks are not allowed to use the innermost lane in any case while they may use the middle lane for overtaking. These rules are reflected in the lane distribution per vehicle type, as most heavy vehicles (Lkw, LkwA and SattelKfz) pass on the right lane (HFS) and only a small number of them is detected on the left

lane (ÜFS2). Passenger cars (Pkw) and delivery vans (Lfw) use all lanes but prefer the middle lane (ÜFS1). All in all, the left lane is the least used, especially in off-peak periods like Sunday morning. The encountered speeds at the section MQ_A2_B_176,550 (see Figure 2.3) may to some extent be influenced by dynamic speed limits. Heavy vehicles run mostly between 80 and 90 km/h, whereas passenger car speeds are rather broadly distributed around 130 km/h. As with the lane distribution, delivery vans and unclassified (Sonst) behave similarly to passenger cars.
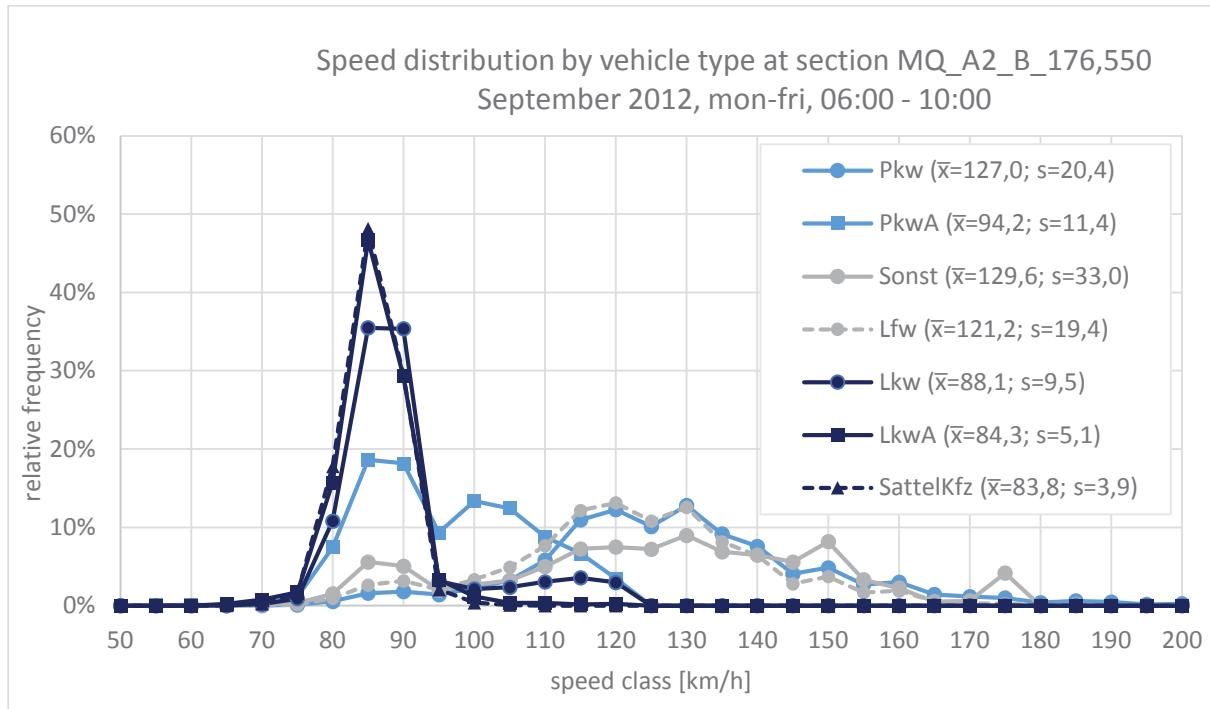
Figure 2.3: speed distribution

As far as gaps are concerned, mainly the short ones are of interest for the calibration. We want to focus on drivers consciously following a leader vehicle because they correspond to what the car following model in the simulation emulates. Only gaps below 8 s are taken into account to exclude most of the free-flow vehicles. Assuming a speed of 100 km/h, this results in a maximum distance of around 220 m between two subsequent vehicles. The distribution is biased as both approaching and following vehicles contribute to it. This is why we mainly look at the peak of the respective gap distributions, excluding the right tail of the curve. Globally, passenger car drivers and similar vehicles follow each with a gap around 1 s while heavy vehicles tend to follow each other with around 1.5 s. This might indicate that professional drivers behave differently.

2.3 Simulation Model

2.3.1 SUMO Initiation

First, the road network has to be build and adapted to the purpose of the study. Consequently, the vehicle flows are added and finally, the initial car following parameters are set. The purpose of the traffic statistics shown before is twofold: On one side, to provide input data for the simulation initiation, and on the other, to be used for comparison during the calibration process. Some parameters of the car following model can be estimated out of the traffic statistics or other sources.

The road network has been extracted from OpenStreetMap and reworked in the graphical network editor NETEDIT. Given the single vehicle data, only the main lanes vehicle volume is known, but
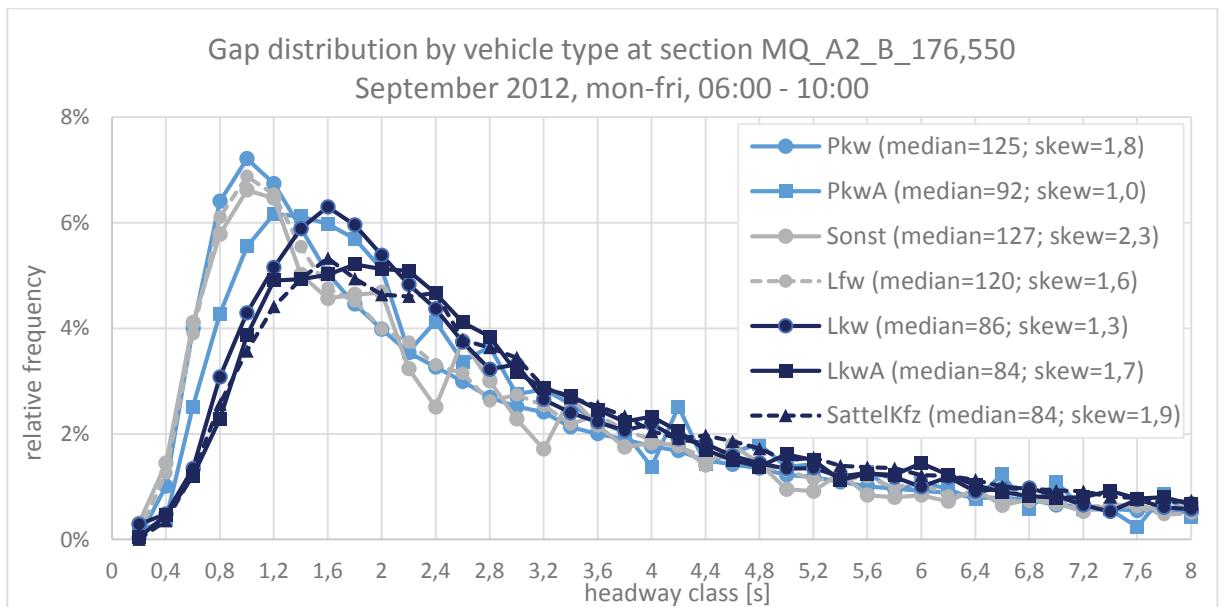
Figure 2.4: gap distribution

Table 2.1: Krauß' car-following model parameters in SUMO

parameter	description
accel	maximum vehicle acceleration
decel	maximum vehicle deceleration
maxSpeed	desired speed
minGap	minimum gap between two vehicles
sigma	driver imperfection
tau	driver reaction time / minimum gap

not how many exit or enter the motorway. However, the rest area is not considered that important to play a role here. Every vehicle entering the network is set to continue on a motorway main lane. A morning period of a working day between 06:00 and 10:00 is used to gather the vehicle volumes and the states of dynamic traffic management. The vehicle volume per lane and vehicle type at the induction loops of km 181.250 is aggregated to time intervals of 1 min and converted into SUMO input flows. During the recorded period, the dynamic speed limits are the same for all sections showing either 120 km/h or unlimited. Hence, all SUMO motorway links are controlled by a single variable speed sign. Additionally, heavy vehicles' overtaking movements are banned sometimes. However, there is no static way to predefine overtaking bans in SUMO before starting the simulation. We ignore it for the time being: The vast majority of heavy vehicles is inserted on the right lane, but is allowed to use the middle lane, too.

Several model parameters need to be initialized with a default value before the calibration can start. Table 2.1 provides an overview of some basic car following parameters in SUMO. We choose the Krauß car following model for our simulation being the standard in SUMO. As it can be seen in the motorway traffic characteristics, a point estimation for a vehicle type will not be able to reproduce the range of speeds and gaps. A subset of the available single vehicle data is used to estimate a car following model parameter distribution per type of vehicle. For this purpose, we take all measured vehicles at the section MQ_A2_B_176,550 during the morning hours of September 2012 into account.

Without further restrictions like speed limits or slow leaders, SUMO vehicles proceed with their desired speed (given by the parameter *maxSpeed*). From the point of view of single vehicle data,

large gaps are associated with a low vehicle density and a high probability for free-flow conditions. Consequently, normal distributions are fitted to the empiric speed density of vehicles having a gap larger than 10 s. For example, trucks' desired speed can be approximated by the normal distribution $N(\bar{x}; s^2) = N(86.0; 4.5)[km/h]$. On the one hand, we draw several vehicle types with their own desired speed out of this distribution. On the other hand, SUMO offers to define a distribution of the link speed limit compliance by providing the *speedFactor* and *speedDev* attributes: The allowed link speed may be exceeded by a factor drawn from a normal distribution with *speedFactor* mean and *speedDev* deviation ((6)). In our application, the *speedFactor* and the *speedDev* attributes are derived from others: Assuming that in general, slow drivers comply more with speed limits, the *speedFactor* is set to the ratio of their individual and the expected speed of the desired speed distribution. The *speedDev* attribute is set proportionally to the deviation of the desired speed distribution.

Even if the distribution of short gaps can be used as an estimate for combined driver reaction and vehicle reaction time, we rely on another study on human reaction times to estimate the parameter *tau*. Even if this parameter has a real meaning, it cannot be easily measured in experiments. Hugemann ((10)) outlines some pitfalls of different measurement designs like human beings accepting more risks in a driving simulator. Therefore, the same author refers to a measurement series done for the German conference on road traffic law ("Deutscher Verkehrsgerichtstag") in 1985 to derive a gamma distribution model for the human reaction time. This approximation is used as an initial estimation of the car following parameter *tau* in SUMO by fitting a gamma density distribution (Figure 2.5) to it.
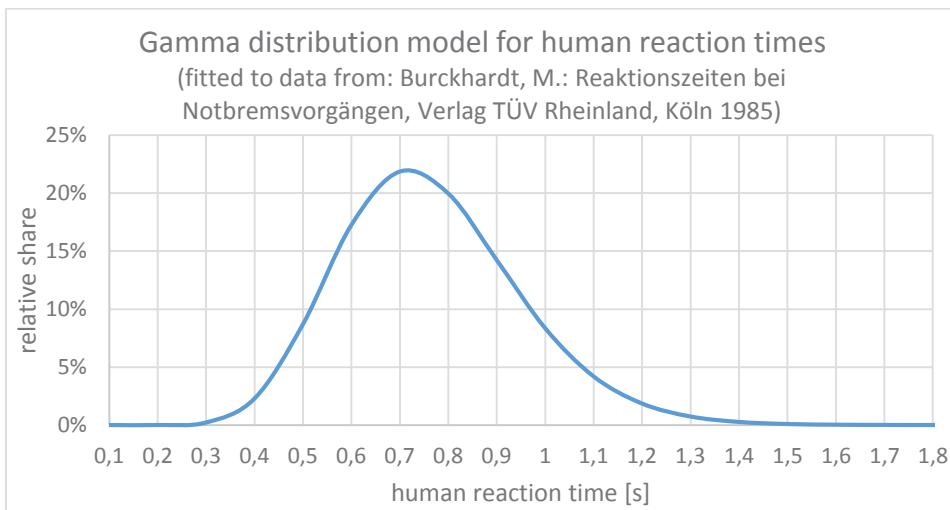
Figure 2.5: Gamma distribution model of human reaction time (shape a=15, scale=1/21)

The remaining parameters including *accel* and *decel* are set to default values drawn from distributions limited by useful limits. For example, *sigma* is defined as number between 0 and 1. The minGap shall be normally distributed around 2 m and the acceleration is set to a normal distribution around the SUMO default value for the respective vehicle type. At simulation start, for every general vehicle type like passenger cars or trucks, a high number of representative vehicle types with individual parameter values is drawn from the distributions explained beforehand.

2.3.2 Calibration and Validation

In general, fine scale models need to be parametrized to allow for an acceptable representation of the modeled object. As traffic flow results out of many interactions between drivers and their vehicles with different characteristics, a multitude of differently parametrized models for microscopic traffic

simulation has been developed. Each of them includes at least some parameters to adapt the model behaviour to the observed one.

Calibration is known as the process of adapting the model parameters to approximate the observed behaviour. Afterwards, the model validation indicates that different input still leads to an acceptable representation of the modeled situation ((1, p. 7)). This step has to be taken before using the model for decision support. Many organizations in the field of traffic modeling have issued guidelines on calibration and validation in the past. In Germany, the aforementioned is considered the main guideline in this topic.

To allow for a transparent calibration process, we have to define when the model may be considered acceptable. In general, it is defined as the critical value of an error metric. This metric takes a meaningful measure which is available in the model as well as from an external source, e.g. from motorway single vehicle data. There is the option to choose one measure for calibration and another for validation, or to evaluate different traffic scenarios by means of the same measure and metric. Our approach here is to use different traffic scenarios and different measures as both have an influence on the calibration process. In fact, interactions between vehicles following each other are more frequent in dense traffic. Hence, the same car following parameter set may have a different impact on the evaluated measures with change of the vehicle volume. Further on, the goal of modeling motorway traffic includes other measures than gaps, too. On the basis of these considerations, the calibration is carried out simulating early morning traffic between 06:00 and 07:00 looking at the gap distribution while the validation is done with rather peak traffic between 08:00 and 09:00 regarding the speed distribution.

In the past, there have been made multiple different efforts to calibrate a microscopic motorway simulation. Hollander and Liu ((9)) list several authors who calibrated their model against given flow, speed, density or travel time values. Kim, Kim et al. ((11)) have been among the first to apply nonparametric statistical tests on distributions of the simulation outcome. Common topics in almost all calibrations include a target function, a target value as well as to determine the number of replications. We explain those steps in the following lines and apply them to our example.

Before entering the calibration main part where parameter values are changed, it has to be defined which parameter should be varied to limit the search space. A sensitivity analysis gives an insight on which (combination of) parameters influence the studied measure to what extent. Several combinations of car following parameter values are evaluated for correlation. Even if due to model complexity, a one-at-time analysis would fail in identifying effects of interrelated parameters, the variation of one parameter with all others at constant value gives some hints on its influence on the vehicles' gap. For this task, the bounds of the search space are defined as equally distant from the mean explained in paragraph 2.3.1. Each parameter gets 10 equally distributed values assigned and the simulation results are evaluated for each of the resulting parameter sets. The car following parameter with the strongest correlation to the average gap is *tau*. Unfortunately, higher reaction times lead to a smaller gap variance. This is probably due to the resulting capacity reduction, when more vehicles closely follow a leader. However, the effect on the gap has to be considered in the calibration. Other parameters with a noticeable influence are *maxSpeed* and *decel*. When using distributions from which the parameter values are drawn, one has to consider the effect of the expectancy and the variance of the parameter distribution on the mean and the deviation of the gap measure.

Many traffic simulations introduce stochastic components to model random variations in driver behaviour. This is done using pseudo-random numbers which are initiated with a (deterministic) starting number. Because of different simulation results per starting number, a certain number of replications has to be run. Overall, the bigger the estimated variance of the studied measure x (e.g. speed) s^2 gets, the more number of replications are needed. Among other guidelines, (1, p. 38) suggests an iterative formula to estimate the number of replications n using an accepted relative

error e_r and t distribution value depending on the significance level α :

$$n \geq \frac{t(\alpha, n-1)^2 s^2}{e_a^2 = (e_r \bar{x})^2} \tag{2.1}$$

If the mean and the variance of the measure change, the iterative process is not guaranteed to converge after a few steps. Burghout ((7)) recommends a different approach where the number of replications is increased by one per algorithm iteration after an initial replication estimate similar to 2.1:

1. Run an initial number of replications $m \subset \{x \subset \mathbb{N} | x > 1\}$ and calculate $\bar{x}(m)$ and $s^2(m)$

2. Calculate n as in equation 2.1 and if necessary, run $m - n$ additional replications with $\bar{x}(n)$ and $s^2(n)$

3. Decide to run one more replication if equation 2.2 is true and recheck the condition with $\bar{x}(n+1)$ and $s^2(n+1)$

$$\frac{t(\alpha, n-1)\sqrt{\frac{s^2(n)}{n}}}{|\bar{x}(n)|} > \frac{e_r}{1 + e_r} \tag{2.2}$$

On one hand, a very small initial replication count m leads to a big variance and subsequent replication estimate. On the other hand, the variance decreases for most increments of the replication number. Therefore, we skip the second step and just choose a slightly bigger initial replication count. Our preferred measure to estimate the number of replications is the average gap at the section MQ_A2_B_176,550, aggregated in time intervals of 10 min. The maximum of the interval-related numbers of replication will be used as the final one.

During the calibration, several parameter sets are tried out. In order to compare them against each other, a target function metric is needed. It is used to assess the overall performance of the parameter set. In other calibration studies, several formulas which take the simulation result and the comparison data into account have been applied. Often, the root mean square metric is used. However, the main idea of our calibration process is to deal with distributions rather than single values. By this approach, we do not focus on time graph data but value a calibrated traffic simulation which has the same frequency of the studied measure. A small offset could easily make a time graph data based target function look worse, even if the vehicle behaviour itself was well calibrated. This is reflected in the target function:

$$Z = \sum_{veh} f_{veh} z_{veh} = \sum_{veh} f_{veh} \sum_i \delta \frac{|F_{sim,veh}(i) - F_{field,veh}(i)|}{max(x_{veh}) - min(x_{veh})} \tag{2.3}$$

The global target Z consists of a weighted average of vehicle type related components z_{veh}. These sum up the difference of the empirical sum curves of the simulated ($F_{sim,veh}$) and the field gaps ($F_{field,veh}$) per vehicle type using a common class width δ. If the simulation mimicked the field gaps exactly, the Z value would be zero. The difference sum is normalized by the maximum range of gaps in order to get a universally understandable metric which can be used for the validation measure, too. Altogether, the target function represents the average relative error in the simulated gap distribution compared to the field measurements. As one can guess from its definition, the target function has to be minimized during the calibration process. We use a target Z value of 1% as an acceptable threshold to stop the calibration process. Like in other calibration works including (1), a threshold is absolutely needed but it remains difficult to justify its value.

The main part of the calibration concerns the change of parameter values in order to decrease the target function value. Per iteration, one of the car following parameters is altered via its associated distribution parameters in order to make SUMO better approximate the gap distribution from Figure 2.4. Their order is as follows: *tau*, *maxSpeed*, *decel*, *accel*, and finally *minGap*. While normally distributed parameters can change the expected value and the variance, the shape of the Gamma distribution used for *tau* is updated. Both *maxSpeed* and *tau* are altered only once because they are considered prone for overfitting given their influence on gaps. The results of the sensitivity analysis done before indicate the direction to take: If the parameter had an effect on the gap mean, its expected value is shifted proportionally to the difference of means between the simulated and empiric gap distributions per vehicle type. If the parameter had an effect on the gap variance, its variance is scaled (inversely) proportionally. Used metrics here cover their peak ratio, their relative mean difference and others which have been tested during the development. While running a predefined number of iterations, parameter sets and target function values are recorded. The parameter set with the smallest target function value below the calibration threshold is passed to the validation and evaluated, too.

The German calibration and validation guideline (1) expects a valid model not to exceed a predefined error when it is used with a different dataset than used for the calibration. As we have used a gap distribution to calibrate the model, local speeds are used for the validation. Therefore, the target function is evaluated taking into account the speed distributions per vehicle type from the single vehicle data and from the simulation runs. Again, we consider the calibrated model valid if it does not exceed a Z value of 1%. Figure 2.6 shows an overview of the calibration and validation process explained above.
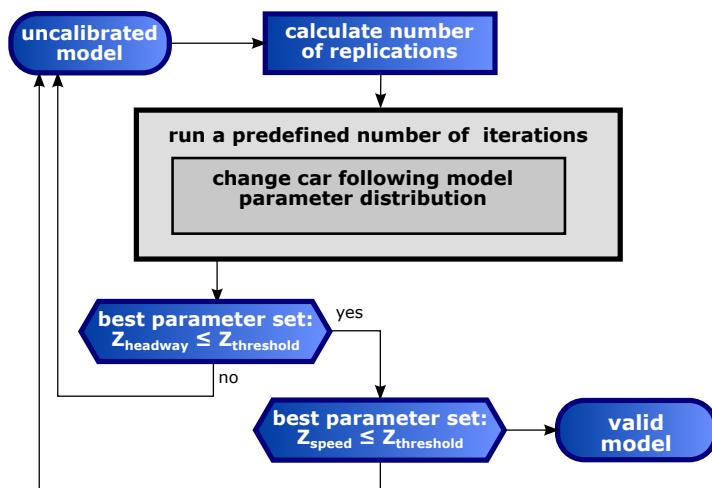
Figure 2.6: overview of calibration and validation)

2.3.3 Application of the calibration to SUMO

Further on, the calibration and validation process presented above has been implemented in a Python script and applied to the example scenario presented in paragraph 2.3.1. Before the first car following parameter changes, the number of replications has to be determined by applying the rules from in paragraph 2.3.2 to the initial simulation parameter set. As a result, six replications with different start seeds have to be run for each car following parameter set. They provide a sufficiently big sample of gaps to assume statistically significant results. The calibration process has been run for 9 more iterations. Each iteration contains a set of car following parameters per vehicle type given by the used distribution model, e.g. a specific normal distribution. Finally, the best iteration results

in a global calibration target function value of 1.2% and a validation target function value of 4.9%. Both target function values are weighted by the vehicle type frequency. In this paragraph, we focus on passenger cars' results but also give an overview on the modeled heavy vehicle characteristics. On the left, Figure 2.7 shows the cumulative distributions of the passenger car gaps up to 8 s used for the calibration. The continuous line represents the single vehicle data source and the dashed one is generated out of the SUMO results. In this case, z_{car} evaluates to 0.5%. To the right, the cumulative speed distributions for the time period 08:00 to 09:00 are shown. Even if the calibration goal is reached when taking only passenger cars into account, other vehicle types outweigh this good performance. The simulation fails to match the validation goal even for passenger cars. The calibration and validation cannot be considered successful as the target function values exceed the predefined threshold. In comparison to real traffic out of peak time, short gaps occur more often in SUMO. When using the same parameter set for the validation time period, high passenger car speeds are underestimated by around 5 km/h. This turns out to be influenced directly by the calibration heuristic which changes the *maxSpeed* parameter as well as the *speedFactor* and *speedDev* value. But anyway, other parameters also have an impact on speed by looking at the speed distribution changing from one iteration to the next.
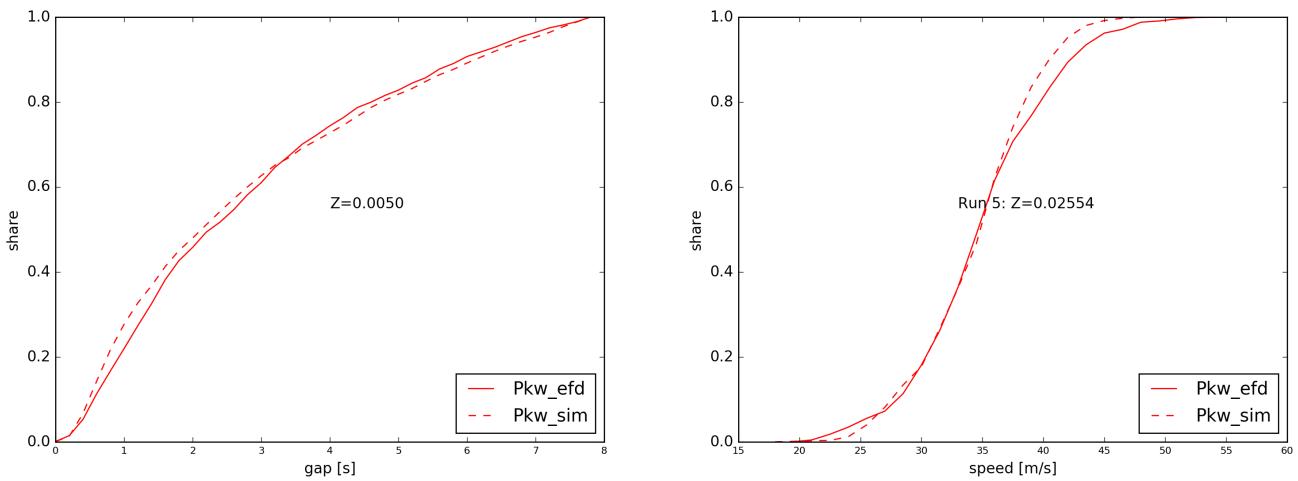
Figure 2.7: cumulative gap and speed distributions of passenger cars

In contrast to the passenger cars result, the behaviour of heavy vehicles like trucks has not been reproduced that well in SUMO. Figure 2.8 shows the corresponding cumulative distributions resulting out of the calibration process. With respect to reality, the amount of short gaps is clearly overrepresented in the simulation. Most of the simulated trucks drive faster than in reality, but there is also a share of trucks clearly exceeding the legal speed of 80 km/h in the single vehicle data. During the development, several variants of the proposed calibration heuristic have been tried out with different simulated vehicle volumes. Unfortunately, in some cases the heuristic has not been able to reach the same level of the target function as presented here. In extreme peak traffic, short gaps have been overestimated even more than shown in 2.7 and 2.8. Very low traffic still resulted in a clearly peaked gap distribution compared to the broad one from single vehicle data. Another traffic characteristic considered before is the lane flow distribution. Still, the lane-change model has not been altered in this study, but it depends on the longitudinal movements created by the car following model. Figure 2.9 shows to which extent the lane usage in SUMO differs from the average single vehicle data based one of the validation time period. Due to normalization by the total volume of the section, the maximal difference depends on the vehicle type frequency.
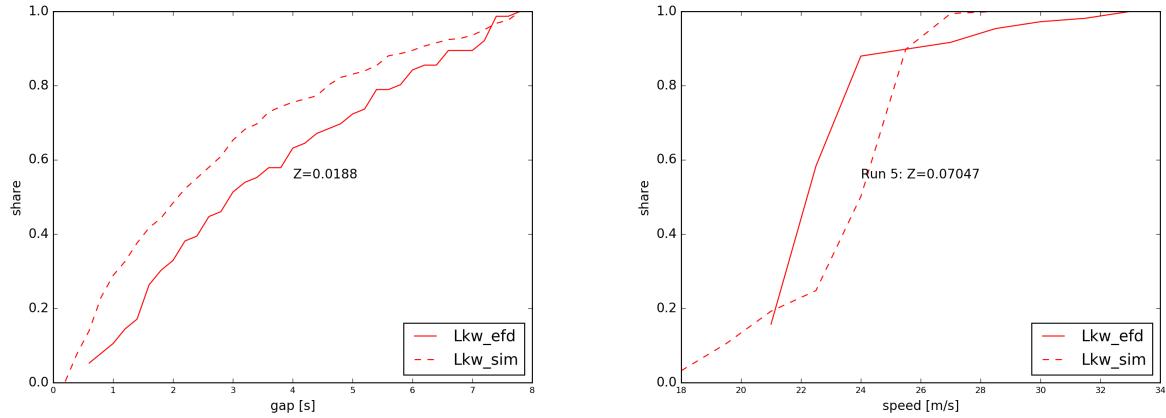
Figure 2.8: cumulative gap and speed distributions of trucks

Nonetheless, passenger cars and delivery vans in SUMO clearly use the third lane more often instead of the second lane. As the differences of the heavy vehicles concern only the rightmost lane, they just indicate a fluctuating heavy vehicle share in comparison to the September 2012 average.
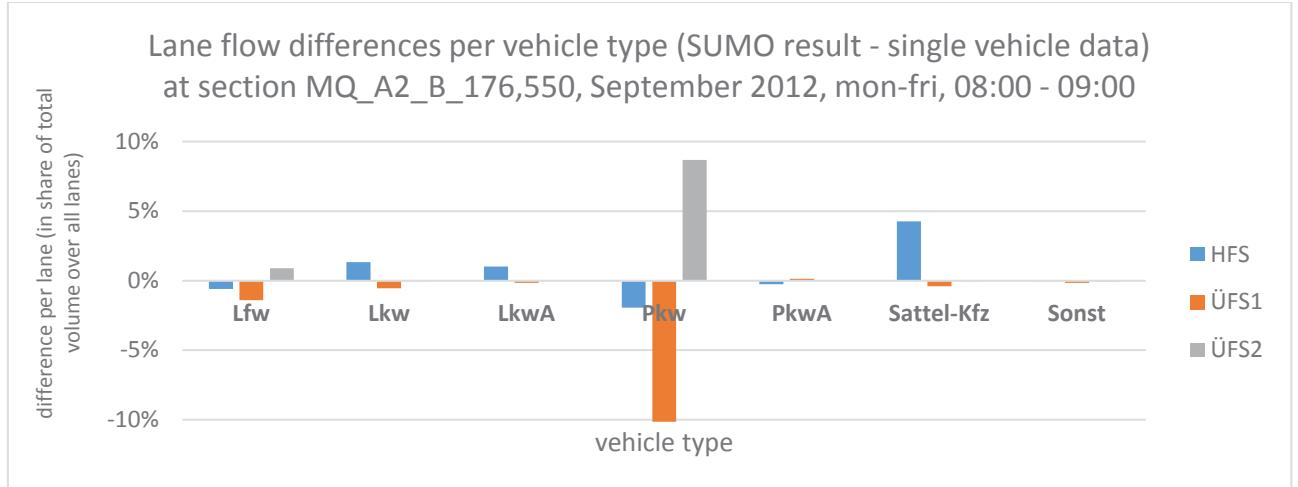
Figure 2.9: lane flow differences per vehicle type

2.4 Conclusion and Outlook

For this work, single vehicle data from motorway induction loops have been used during the calibration of the microscopic traffic simulation SUMO. The simulation offers an output interface which corresponds well to the single vehicle data measures. That is why single vehicle data can be easily transferred to SUMO or used as comparison data. Especially in a motorway use case, where several induction loops are used for traffic management, they represent valuable input data.

During the development of the calibration methodology, sensitivity analysis has helped to determine the most influent parameters on the gap distribution. Among the car following parameters of the Krauß model, *tau*, *maxSpeed* and *decel* have a big impact. But a further look into the lane change model might give more specific guidance for the simulation of multi-lane roads.

By default, SUMO vehicle types use scalar values for car following parameter values instead of distributions. A vehicle type sample had to be drawn out of predefined distributions by the Python

script. Even if the calibration target to match both gap and speed closely has not been reached, the resulting measure distributions are similar to the empiric ones in shape and width. Mainly, there are some over- or underestimations of limited gap and speed intervals which need to be addressed. This represents a big progress compared to the SUMO standard case. Additionally, the lane flow differences between single vehicle data and SUMO second further studies which take the lane change model into account.

The calibration heuristic itself may be subject to change in the future, because the parameter variation did not always result in a shrinking target value. An extended sensitivity analysis may clarify the reasons for that. Unfortunately, the findings of the research project related to (8) concerning SUMO motorway simulation could not be considered in this paper as they have not been published yet. Other characteristics like the motorway capacity could be used for validation as long as enough traffic breakdowns can be observed in single vehicle data.

2.5 References

[1] *Hinweise zur mikroskopischen Verkehrsflusssimulation : Grundlagen und Anwendungen*. FGSV ; 388. FGSV-Verl., Köln, 2006 edition, 2006.

[2] *Handbuch für die Bemessung von Strassenverkehrsanlagen : HBS*. FGSV ; 299. FGSV-Verl., Köln, 2001 edition, 2009.

[3] *Hinweise zur kurzzeitigen automatischen Erfassung von Daten des Straßenverkehrs*. FGSV ; 120 : W1, Wissensdokumente. FGSV-Verl., Köln, 2010 edition, 2010.

[4] *Empfehlungen für Verkehrserhebungen : EVE*. FGSV ; 125 : R2, Regelwerke. FGSV-Verl., Köln, Ausg. 2012 edition, 2012.

[5] *Straßenverkehrsordnung*. http://www.gesetze-im-internet.de/stvo_2013/, 2016. accessed 16.02.2016.

[6] *SUMO Wiki*. http://sumo.dlr.de/wiki/Main_Page, 2016. accessed 16.02.2016.

[7] BURGHOUT, W.: *A note on the number of replication runs in stochastic traffic simulation models*. http://www.ctr.kth.se/publications/ctr2004_01.pdf, 2004. accessed 16.02.2016.

[8] GEISTEFELDT, J., S. GIULIANI, P. VORTISCH, U. LEYN, R. TRAPP, F. BUSCH, A. RASCHER and N. CELIKKAYA: *Assessment of Level of Service on Freeways by Microscopic Traffic Simulation*. Transportation research record, (2461):41–49, 2014.

[9] HOLLANDER, YARON; LIU, RONGHUI: *The principles of calibrating traffic microsimulation models*. Transportation, 35(3):347–362, 2008.

[10] HUGEMANN, W.: *Driver Reaction Times in Road Traffic*. In *Driver Reaction Times in Road Traffic*, Portoroz, 2002. Jahrestagung des Europäischen Vereins für Unfallforschung und Unfallanalyse.

[11] KIM, S.-J., W. KIM and L.R. RILETT: *Calibration of Microsimulation Models Using Nonparametric Statistical Techniques*. Transportation research record, (1935):11–16, 2005.

[12] KRAJCZEWICZ, D., J. ERDMANN, M. BEHRISCH and L. BIEKER: *Recent Development and Applications of SUMO - Simulation of Urban Mobility*. International Journal On Advances in Systems and Measurements, (5):128–138, 2012.

3 Beyond OSM − Alternative Data Sources and Approaches Enhancing Generation of Road Networks for Traffic and Driving Simulations

Andreas Richter, Hartmut Friedl, Michael Scholz; German Aerospace Center
{Andreas.Richter, Hartmut.Friedl, Michael.Scholz}@DLR.de

Abstract

Often OpenStreetMap data is used in the creation of the anderlying road networks for traffic simulations; sometimes it can be enriched with navigation data. Due to the fact that traffic simulation plays a growing role for the development of driver assistance and automation systems more detailed road networks are necessary because traffic simulations are coupled to driving simulators. This detailed information on lane-level and road infrastructure is not available in OSM or navigation data or modelled in an inappropriate way.

This paper will discuss alternative data sources like cadastral or mobile mapping data, their advantages and disadvantages compared to OSM and navigation data and will give an idea on how inappropriate data could be improved. We will show approaches of how to fuse this data to a road network using the OpenDRIVE format resulting in a suitable output for the use in traffic and driving simulations.

Keywords: traffic and driving simulations, road network generation, data sources

3.1 Introduction

The development of driver assistance and automation systems is still the next big thing in the automotive domain. It is important to differ from other car manufactures but furthermore to shape the future of mobility (1). These systems are currently getting more and more complex due to their growing functionality and connection between other vehicles and to the outer environment of the vehicle. To test all this complexity the driving simulation is an essential tool to reduce time and cost effort. Therefore a driving simulator needs a detailed and highly precise road network often accompanied by an appropriate environment model.

Connecting driver assistance and automation systems to each other and interchanging data with other participants raises the need for testing of applications and systems in traffic simulations with larger road networks and more traffic participants, too. Examples are multimedia systems with global and micro navigation functionality or the upcoming vehicle-2-X communication. Therefore the coupling of driving and traffic simulations is a good idea to combine both testing purposes. Due to the different requirements on the anderlying road network the coupling of driving and traffic simulation shows some issues. For example, positioning of the traffic participants differs due to different ways of calculating the movement. Also different ways of linking the road networks can cause a

varying behaviour in route finding. Challenges in the coupling of a driving simulation with a traffic simulation are described in detail using the example of connecting SUMO with VirtualTestDrive (2).

3.2 State of the Art

3.2.1 Traffic Simulation

OpenStreetMap (OSM) is one of the favourite data sources for road network generation for the traffic simulation SUMO because of its extensive world-wide coverage; it is more or less continuously updated and provided for free. Besides OSM also road networks from other traffic simulators like VISUM, Vissim and MATSim are used (3), which often base on OSM data, too.

OpenStreetMap

All these networks have in common that they describe a logically linked road network with some attribute data such as speed limits or other traffic rules but also integrates interesting information such as the number of lanes and additional road furniture. To understand OSM it's good to know, that geometry and all descriptions are stored in an xml formatted text file. Each feature is described with tags. A tag consist of a key and a value, e.g. highway=residential. In most cases tagging follows guidelines to describe defined features, but in general tagging is free and therefore good and easy for crowd sourcing. To process OSM for other application data is often translated in tables good for reading in relational databases such as PostgreSQL. This is easy for single features but can get complicated for larger data extractions. The following figure 3.1 shows a JOSM screenshot, displaying a complex intersection in Braunschweig, Germany. It shows road edges, lanes including traffic rules, traffic signs, additional road furniture and the description of the way highlighted in red in the included table.
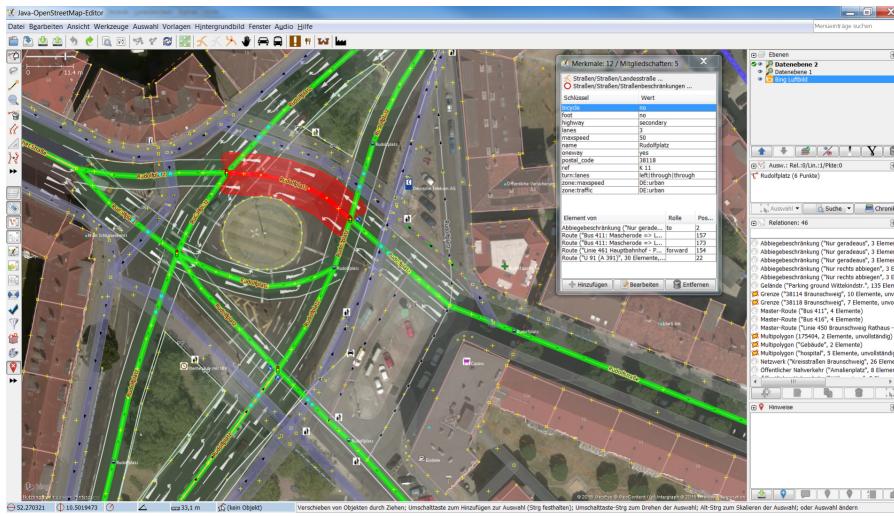
Figure 3.1: complex intersection with information about road edges, lanes including traffic rules, traffic signs and additional road furniture

The amount of additional data about road infrastructure and lanes grew in the last years. Data was added especially for intersections. Nevertheless OSM data has some major disadvantages for driving simulators. The following sections are going more into the technical details of the data. All following statistics were collected with http://taginfo.openstreetmap.org on 24th of February 2016. Crossroads are often simplified for map representation. The above pictured intersection is a good

exception due to the street island. Less sophisticated intersections are not modelled in such a precise way as a second intersection in Braunschweig shows in figure 3.2.
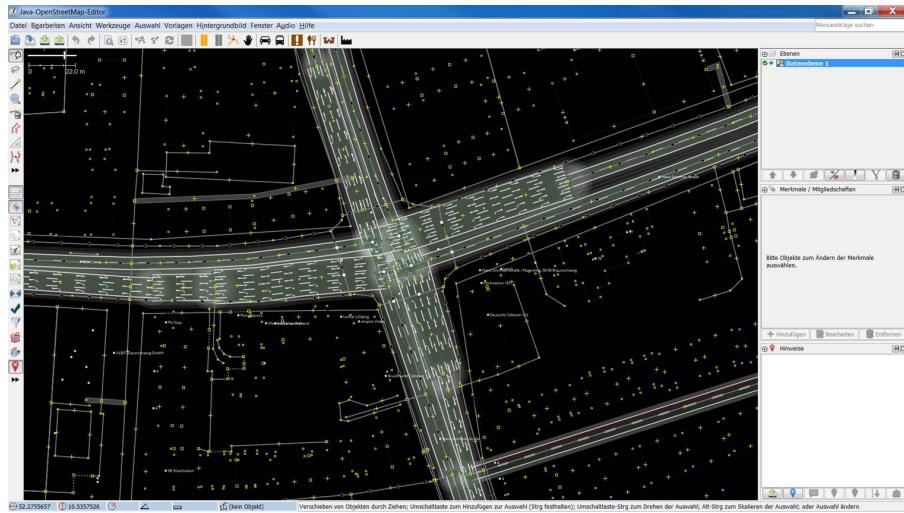
Figure 3.2: larger intersection with simplified layout

In general there is mostly a defined way of tagging and adding additional data to the OSM database but practically there are still different ways of tagging that information. Especially for new kinds of data a common way has to be established first, e.g. proposed for railway infrastructure (4). Concerning the tags for [lanes] or [turns] mostly more than 99% of the tags are correct, but tags like [width] must be cleaned because of information on measurement units being often included. Additionally, it can be confusing how to derive the information from different tags. Tags such as [turn], [turn:lanes], [turn:lanes:forward], [turn:lanes:backward], [turn:lanes:both_ways], [turn:forward], [turn:backward] or, in typically rare but existing exceptions [lanes:turnleft], come up with the layout information of an intersection. The [turn:lanes:forward] and [turn:lanes:backward] tags have the disadvantage that they can only describe the linkage at the beginning and at the end of a way and all intersections in between are not considered. That is a drawback especially for long roads (5).

Moreover the latter tag [lanes:turnleft] follows a different logic, describing the amount of lanes turning left, while the previously mentioned tags are describing the lane turn types like [left|through| through|right][1].

Additionally, typos and careless editing lead to a network that is not usable out of the box, in particular for the road network (5; 6). It has to be cleaned and corrected. The following figure 3.3 shows a complex intersection in Stuttgart, Germany, where some lanes in the north-west were tagged with a future validity date, which has already been reached at the time of browsing. Furthermore, these lanes are connected to nowhere.

A driving simulator depends on the topographical representation of the roads. But just 1% of all highways provide a [width] tag, or 4% of [highways] with a [lane] tag also define a [width] tag. Only 0.03% of all [lanes] provide a [width:lane] tag. Information on connections in-between lanes does not exist but can be estimated/constructed in most cases. There is also no information on detailed road markings (transverse markings, no passing zones) but may be constructed from the [change:lanes] tag. There is not always a need for this tag, so it is hard to say how complete this information is. At least more than 20% of [turn:lanes] are tagged.

There are lots of approaches to detect, enhance or correct locations of road networks like OSM (7). For driving or traffic simulation there is little gain in correcting locations, even though locations in OSM offer sufficient quality. For simulation approaches enhancing topographic data, such as the

---

[1]http://taginfo.openstreetmap.org/tags/turn%3Alanes%3Aforward=left%7Cthrough%7Cthrough%3Bright
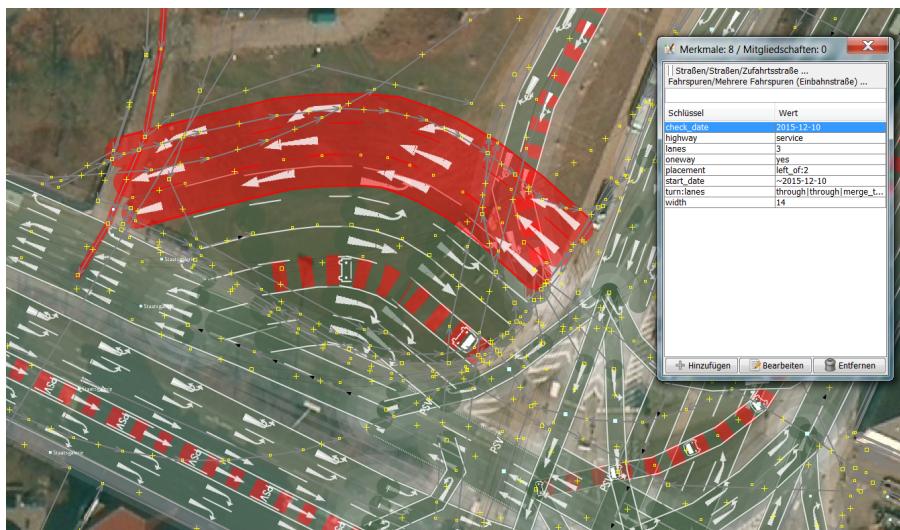
Figure 3.3: intersection with incorrect lane information

width of roads, is more prospering. Road width could be calculated from very detailed town plans including curbstones (8) or from ortho images (9).

Navigation Databases and Cadastral Data

The pre-processing can be minimised by using official data provided by map makers such as HERE or TomTom and already prepared road networks from other traffic simulation tools. The standard navigation data lacks information about the amount of lanes and lane width, too. The following figure 3.4 shows a HERE road network with underlying road topography loaded as Shapefile. It clearly shows the mismatch of lane widenings, which leads to a wrong representation of the capacity of turning lanes.
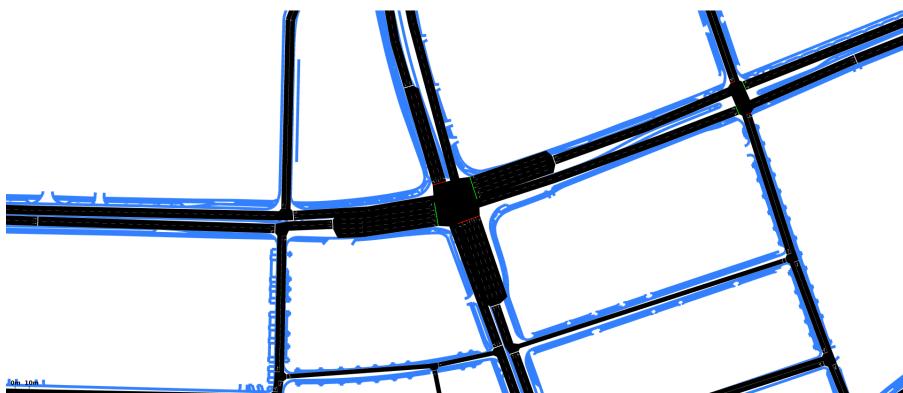
Figure 3.4: road network based on navigation data compared with topography data

But there exists so called High Definition or ADAS data, which is enriched with elevation profiles and road curvatures as additional information for assistance and automation systems. Actually they contain recently tagged information about lane widening and linkage. But again, most of these data lack topographical information about width of the lanes (5). For example, in the case of a lane widening for a left-turning lane, it is of importance at which point this new lane offers enough space for vehicles not to block the neighbouring lane any more.
An additional drawback is that these navigation data sets have to be purchased and sometimes are not available nationwide.

Official German cadastral data DLM5 does contain information about the lane width for some road categories. The following figure 3.5 shows the same intersection as above, highlighting the roads with lane width information. Only major roads are represented in detail, but there is also no information about lane widening available and intersections are generalized.

Figure 3.5: highlighted roads with information about lane width in cadastral data

3.2.2 Driving Simulation

Data bases for driving simulations are getting bigger and bigger due to their usage in model- or system-in-the-loop tests. Additionally, more and more real world data is needed in the driving simulator to close the gap between simulated and real world test (10).

A road network for a driving simulator does not only contain the logical linkage of the roads and their lanes. The major part is its topographical description of the geometrical road appearance. A third component is the description of road infrastructure such as road signs, traffic lights and important road objects. Therefore specified formats are used, for instance OpenDRIVE (11), which evolved as a de-facto standard in the driving simulator domain.

Currently, suitable data is poorly available. Most of such OpenDRIVE data bases are manually built based on areal images as guidance or even totally artificially built. Both approaches are time-consuming processes. A mobile mapping road survey is able to deliver exactly the needed data in a very accurate way. The following figure 3.6 depicts the same intersection as in the images above, modelled with the help of mobile mapping data.

However, if the area of interest is growing and contains many complex situations, the surveying of roads can get very expensive and time consuming as well. The use of cadastral data can fill the gap because it can deliver a good coverage, e.g. city-wide, but sometimes does also lack interesting data for lane-level description or information about the road infrastructure (12).

3.3 Road Network Generation

The project "Virtual World" builds upon the experience of the "SimWorld$^{\text{URBAN}}$" project, which fused survey data with cadastral and land use data as well as areal images to generate a detailed road network (13). To process the cadastral data a computer graphics approach combined with geographic information systems functionality was used to generate the basic information, such as reference lines and lane borders, and to add road infrastructure.

The following paragraph gives a brief overview about how a road network can be generated. A more
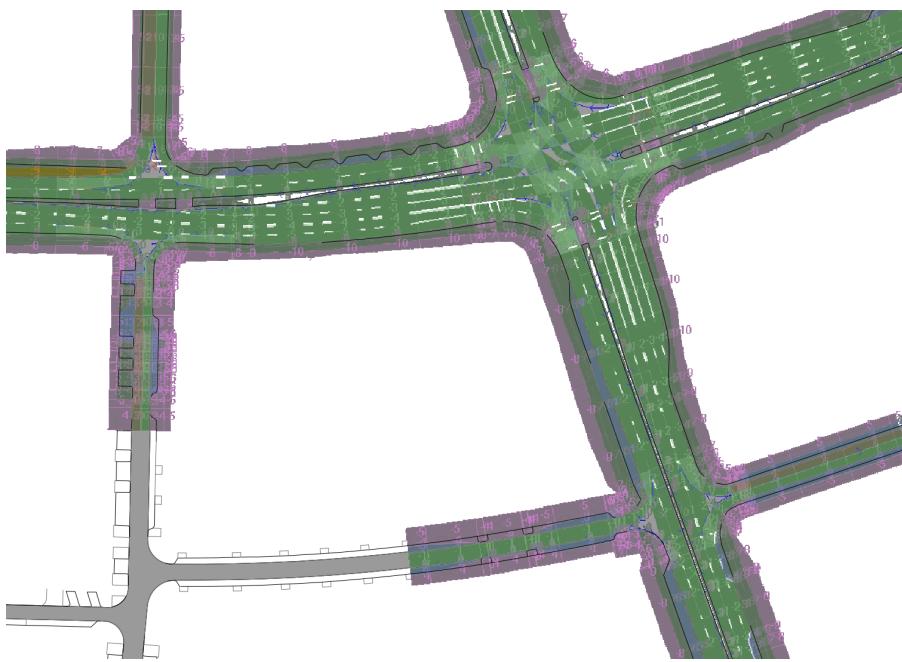
Figure 3.6: intersection based on mobile mapping data

detailed description can be found in (14).

As starting point a geo-dataset of curb stones from cadastral street topography is used. The data comes with geometrical inconsistencies and has to be cleaned and integrated in the first step. At this point there is also no information about driving lane properties present as this dataset consists of plain linear geometries without any semantics of the road surface area other than its boundary. The whole, cleaned line network is polygonised to determine potential asphalt zones for motorised traffic. The appropriate area can be selected with a combination of cadastral road centrelines and centrelines from OSM/HERE. Reference lines serve as anchoring of all further street information such as lanes, signals, and elevation. A reference line is ideally located in the centre of a road. It describes the road's course in detail as a mathematical function. From the polygonised asphalt zones such reference lines are derived as skeletons of the polygons. The outcome is a linked line network which describes the course topographically correct. A potential road junction exists where the skeletonised reference lines join in one or multiple neighbouring points. The linkage in such junction areas is still geometrically inappropriate and does not describe the real bend of curves. Therefore line segments within these junction areas are removed and exchanged by more realistic arc-, spline-, and spiral-based connection paths. For broad streets with centre islands and complex lane layout the reference lines are translated onto the centre street islands for correct lane reference. Information about these centre islands is also gripped from the polygon dataset. For single-lane roads an appropriate translation of the reference line is performed as well. One-way roads can be considered also if appropriate meta-information is extracted from OSM or HERE, for example. Each reference line is associated a polynomial elevation model derived from digital raster-based elevation data. Such digital terrain models are available from photogrammetric or radar-based remote sensing or can be generated from laser scanning. Based on the corrected reference lines the road lanes are generated. A rule-set interpolates the count and width of lanes incorporating distances between reference lines and asphalt borders and width thresholds. At junctions logical relationships between incident lanes are added generically and can be corrected with metadata from OSM or HERE again. Those rudimentarily generated lanes already serve well for basic representation of the street network and can automatically be furnished with markings. Also grass borders, bike lanes and pedestrian walks could be estimated from cadastral data to enrich the road network.

If there is more information available in the input data, e.g. lane markings, the approach works better

and generates a more realistic road network (8; 15). A similar approach can be used to generate the environment, e.g. the city model (14; 16) or road infrastructure. The following figure 3.7 shows the same intersection as before in a fully equipped driving simulator environment.

Figure 3.7: 3D driving simulator environment

As descripted the road network generation can be improved by introducing OSM or HERE data to support the generation of road axes and to enrich these roads with road furniture. Road infrastructure, such as signs and traffic lights, is added with the help of extracted data from appropriate cadastral sources. At the end the road network is generated in the OpenDRIVE format.

Even though a lot of information, such a city models, road topography, road infrastructure, is already available in a digital way, in most cases this data is not machine-readable and thus cannot be processed in a proper way. Additionally, these data can also lack accuracy due to careless editing and different ways of gathering of the raw data, such as digitalisation of analogue maps or processing of aerial surveys.

Nevertheless the DLR is working together with major German car manufactures on guidelines to survey road networks in a simplified but enhanced way to meet cadastral and simulation requirements. The guidelines will also cover the gathering of attribute data, such as the relation and the type of lines and points, in order to support a later transformation of these data into specific driving simulator or navigation formats like Navigation Data Standard (NDS). In the future cadastral data should be streamlined for additional usage and not just for digital visualisation. Therefore a growing database should be available to generate detailed road networks out of cadastral data.

3.4 Dual Use

SUMO's netconvert application is able to read OpenDRIVE and transform it into SUMO's internal road format (3). It takes over all necessary information like lane linkages and infrastructure, such as traffic lights (2). Therefore the OpenDRIVE network is fully functional and can be used by SUMO and the SUMO suite tools. The following figure 3.8 shows the OpenDRIVE road network from the two previous pictures loaded in SUMO with underlying road topography loaded as Shapefile.

The representation is more or less equal and differs only because of the less detailed internal road description format of SUMO. Most important is that all identifiers and connections stay unchanged and make a coupling between traffic and driving simulation possible. The driving simulator and the traffic simulator can work on the same road network without the need to translate road network elements. Still the positioning of vehicles in the driving simulator has to be interpolated with the
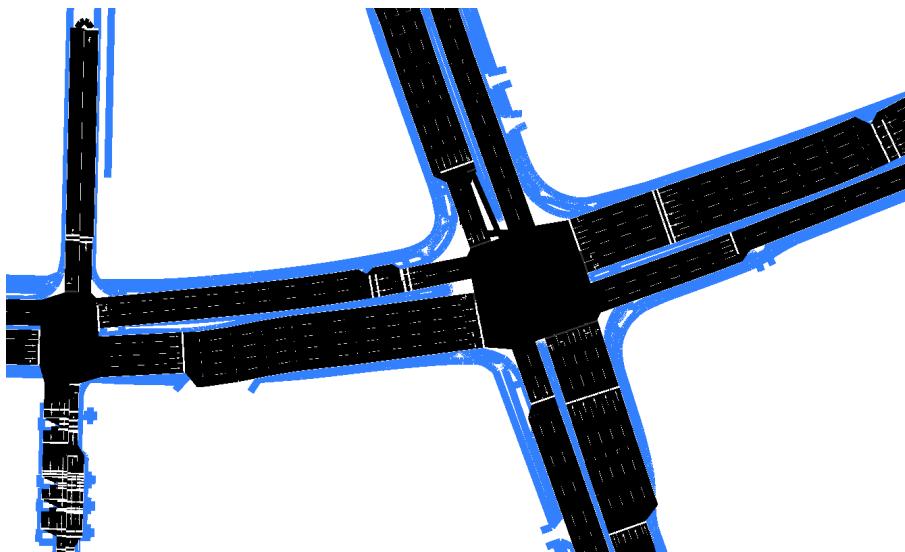
Figure 3.8: OpenDRIVE road network compared with topography data

help of the road identifiers.

As side benefit SUMO's netconvert can also export OpenDRIVE networks. Therefore it is possible to create a huge road network database for driving simulators. Nevertheless, it will show all the drawbacks of SUMO's internal road network format, for instance, the mathematical road curvature being discretised as line strings.

3.5 Summary

We showed that it is possible to generate road networks based on different kinds of geodata. In general the fusion of heterogeneous sources is necessary to reach the goal of a precise road model. OpenStreetMap can be such a data source but does not have to remain the only one. However, OSM currently cannot fulfil all requirements of upcoming use cases. For example, for the test of emergency vehicle rescue lanes or continuous lane change, it is necessary to model the width change of lanes in a proper way. For the coupling of driving simulation and traffic simulators more detailed information will be required. OSM cannot provide this information, but it can be derived from cadastral data. To model this information in a suitable manner DLR will announce guidelines for a generally accepted way of representing all required information.

3.6 References

[1] Maurer, Markus and Gerdes, J. Christian and Lenz, Barbara and Winner, Hermann (Hrsg.), (2015) *Autonomes Fahren. Technische, rechtliche und gesellschaftliche Aspekte.*, DOI 10.1007/978-3-662-45854-9

[2] Krajzewicz, Daniel and Richter, Andreas and Behrisch, Michael and Erdmann, Jakob (2013) *Abbildung des Umgebungsverkehrs in einem Fahrsimulator.* VDI Verlag. 4. Berliner Fachtagung Fahrermodellierung, 13.-14. June 2013, Berlin, Germany. ISBN 978-3-18-303522-9. ISSN 1439-958X

[3] Krajzewicz, Daniel and Erdmann, Jakob and Behrisch, Michael and Bieker, Laura (2012) *Recent Development and Applications of SUMO - Simulation of Urban MObility.* International Journal On Advances in Systems and Measurements, 5 (3&4), (pp. 128-138). ISSN 1942-261x.

[4] Rahmig, Christian and Simon, Augusto (2014) *Extracting Topology and Geometry Information from OpenStreetMap Data for Digital Maps for Railway Applications.* 10th ITS European Congress 2014, 16.-19. June 2014, Helsinki, Finland.

[5] Sämann, Robert (2014) *Bestimmung einer Bewertungsmetrik zum Vergleich digitaler Straßennetze für Verkehrsflusssimulation und Routing von Einsatzkräften.* master thesis, Leibniz University Hanover.

[6] Krumnow, Mario (2014) *Modeling Mobility with Open Data* http://www.sumo.dlr.de/2014/Presentation_Keynote_Open%20data%202014_MarioKrumnow.pdf SUMO - Simulation of Urban Mobility 2014, 15.-16. May 2014, Berlin, Germany.

[7] Butenuth, Matthias (2010) *Geometric refinement of road networks using network snakes and SAR images.* In Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International (pp. 449-452). IEEE.

[8] Richter, Andreas (2015) *Geodaten: Vom Kataster in die Fahrsimulation.* 1. Fachtagung Fahrsimulatoren, 07. July 2015, Stuttgart.

[9] Mattyus, Gellert and Wang, Shenlong and Fidler, Sanja and Urtasun, Raquel (2015) *Enhancing Road Maps by Parsing Aerial Images Around the World.* Computer Vision (ICCV), International Conference (pp. 1689-1697). IEEE. International Conference on Computer Vision, 13-16 Dez. 2015, Santiago de Chile, Chile.

[10] Richter, Andreas and Fischer, Martin and Frankiewicz, Tobias and Schnieder, Lars and Köster, Frank (2015) *Reducing the gap between simulated and real life environments by introducing high-precision data.* Driving Simulator Conference 2015 Europe, 16.-18. Sep. 2015, Tübingen, Germany. ISBN 978-3-9813099-3-5.

[11] *OpenDRIVE® Format Specification, Rev. 1.4 Issue H* (2015) http://www.opendrive.org/docs/OpenDRIVEFormatSpecRev1.4H.pdf

[12] Friedl, Hartmut and Richter, Andreas (2012) *Fusion heterogener Geodaten zur Erstellung realer 3D-Welten am Beispiel einer Fahrsimulation.* Shaker. GEOINFORMATIK 2012, 28.-30. March 2012, Braunschweig. ISBN 978-3-8440-0888-3. ISSN 1618-1034

[13] Richter, Andreas and Friedl, Hartmut and Guraj, Vitalij and Ruppert, Thomas and Köster, Frank (2011) *Developing a toolchain for providing automatically highly accurate 3D database.* Verlag der Bauhaus-Universität Weimar. ConVR2011, 03.-04. Nov. 2011, Weimar, Germany. ISBN 9783860684580.

[14] Richter, Andreas and Scholz, Michael and Friedl, Hartmut and Ruppert, Thomas and Köster, Frank (2016) *Challenges and Experiences in Using Heterogeneous, Geo-Referenced Data for Automatic Creation of Driving Simulator Environments.* Transactions of The Society for Modeling and Simulation International, Special Issue Driving Simulation 2016. Prepublished 08. April 2016, DOI: 10.1177/0123456789123456

[15] Richter, Andreas (2015) *Master the Challenge of Generating OpenDRIVE Road Networks based on Real World Data.* 8. OpenDRIVE User Meeting, 15. Oct. 2015, Stuttgart, Germany.

[16] Richter, Andreas and Friedl, Hartmut (2015) *Parametrisierte Stadtmodelle für Fahrsimulatoren.* GIS Talk 2015, 19.-21. May 2015, Unterschleißheim, Germany.

4 SUMO as part of a testing platform for highly automated driving in urban areas

Mario Krumnow; Technische Universität Dresden
Mario.Krumnow@tu-dresden.de

Abstract

Constantly new Advanced Driver Assistance Systems (ADAS) are being developed. The estimation of the reachable benefit of such complex ADAS systems mainly depends on the particular components for instance internal car sensors, radio communication units, localization devices and of course the used algorithms. To improve this process, a funding project has started in 2015. The goal of the project is the support of the whole developing process. Which contains the simulation part, the planning and execution of test rides and the analysis of the recorded sensor data. This paper will give a first overview about the concept of this research project.

Keywords: SUMO, TraCI, TraaS, automated traffic, V2X

4.1 Introduction

The connection between vehicles and their surrounding infrastructure is currently an active research topic, see (3). Prior new driver assistance systems tries to benefit from such V2X communication technologies to make the driving task more efficient and even more convenient. As a consequence the automated driving shifts more and more into the focus of many car manufactures. One of the great challenges is the automation in urban areas because of the complex scenarios (e.g. traffic lights, public transport) that has to be managed, see (10). To improve this process, a funding project named REMAS (Resource management system for automated driving) has started in 2015 with the goal to support the whole developing cycle for new ADAS systems.

Starting with the simulation, the planning of test rides and the analysis of the obtained results. In order to consider numerous kinds of influences, a couple of simulations suits were used (e.g. traffic flow, communication, localization) to manage a specific scenario. The fact that all the unique simulators should work together to see the real effects of new ADAS systems requires to build a system, where all simulators get connected and exchange their information. For microscopic traffic simulations SUMO(2) will be used together with a predefined maneuver catalogue which includes intersections, bus stops, interaction between public transports and so on.

4.2 Related Work

Generally, traffic simulations are used to analyze the traffic flow under specific conditions. Different car following models are used to represent the normal driver with his usual behavior. In order to

simulate highly automated cars it is necessary to manipulate the default behavior of the cars in the microscopic traffic simulation. Therefore vendors of simulation software generally offer software API's to support this approach. For examples PTV Vissim uses an external interface to support also traffic light controllers of other providers. Furthermore the reachable level of detail in a single traffic simulation is usually restricted due to performance issues, see (6).

For the open source traffic simulation suite SUMO there are numerous plugins and extensions available. One of the most famous tools for evaluation of the network protocols during the V2X communication is Veins (8) which is based on OMNeT++. This tool can simulate the communication process of automated cars with support for some common protocols like IEEE 802.11p and the cellular networking with LTE.

The simulation of the internal processes of the vehicle (e.g. electronic control unit, drive train) are mostly done by specific Matlab simulations, see (7). Similar approaches are used to simulate different kinds of car devices like GPS receivers or car sensors. Due to the fact a high-level architecture will be used to handle the communication between the different simulation suits.

4.3 REMAS Project

Within the REMAS project a huge workbench will be established to support many specific tasks in the field of automated driving. This begins with the definition of the content for the required standards, for instance Decentralized Environmental Notification Messages (DENM) or Cooperative Awareness Messages (CAM), and ends with the post processing of different test rides. A Brief overview of the most important parts of the REMAS System is shown in Fig 4.1.
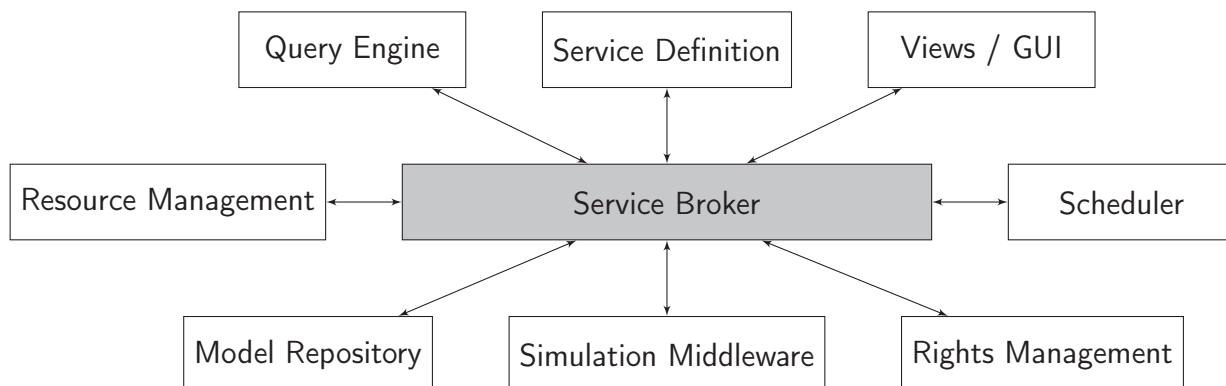
Figure 4.1: REMAS Structure

The backbone of the system is a service broker which connects all the different components. It follows a modular concept which is extensible very easily. One of the most important things is the graphical user interface that gives access to all the shown components. Currently, this is a content management system (CMS) which works like other famous WIKI applications. Furthermore the CMS implements some restful services that are reachable with the HTTP protocol. Due to this fact users can interact with the platform as well as other software services. For the last one there exists an information exchange based on JSON[1] objects.

For the activities around the developing and evaluation of automated cars a lot of information exchange is necessary. For example to evaluate a new localization device the knowledge of the specification is mandatory. Typically an important part is to simulate the behavior of the device in some accurate and comparable simulation scenarios in order to see the benefit of the new device as part of the entire ADAS system. For this reason a maneuver catalogue is implemented which

---

[1]JavaScript Object Notation

contains typical traffic situations especially designed to see the benefit of new ADAS components. The initialization and calibration of the models will be done with real traffic data from the local traffic management center, see (4).

After a successful simulation phase the results will be verified on real roads. In order to reproduce the simulation results a defined traffic scenario is necessary. For example the chosen traffic scenario requires a 4-way intersection with a traffic actuated traffic light controller and a traffic flow of 400 vehicles per hour. The scheduler and the resource management component of REMAS can find such conditions by analyzing the road network and the data of the traffic detectors. The produced data during the test rides are stored in the REMAS system and gets accessible with the query engine.

4.4 Simulation Environment

The focus of this paper is the simulation middleware of REMAS. Here, multiple simulations get connected and exchange their information. Figure 4.2 illustrates the main concept. A high-level architecture (HLA) based on the IEEE standard 1516 (1) is used. In order to facilitate interoperability and reusability, HLA distinguishes between the simulation functionality provided by the members of the distributed simulation and a set of basic services for data exchange, communication and synchronization.

Federate	Federate	Federate	Federate
Traffic Simulation	Vehicle Simulation	Radio Communication Simulation	Positioning Simulation

Run-Time Infrastructure
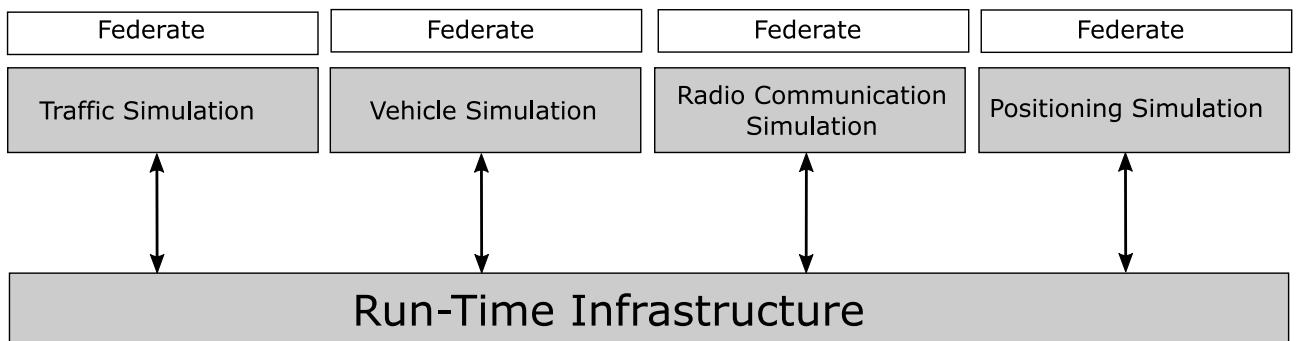
Figure 4.2: HLA - Simulation Middleware

In a HLA, every simulation suite is called federate, and these entities can interact with each other within in a so called federation. A federation is a set of federates acting together. This concept provides functionality for macro-, micro- and nanoscopic traffic simulations at the same time, see (5), (9). The inter-process communication is managed by a Run-Time Infrastructure (RTI) which follows the specification rules of the IEEE standard. Each federate communicates with the RTI using an interface called ambassador. Generally, the ambassador is the API of the simulation suits which allows external bidirectional access.

For REMAS the microscopic traffic simulation suite SUMO is used. This is possible because of the TraCI[2] interface which acts as the ambassador of the federate. The ambassador of the RTI is an implementation of the TraaS[3] library which is written in java and allows two kinds of usage. On the one hand it can be used as a java TraCI Client which connects directly to SUMO and on the other hand it is possible to run as a web service which provides all the TraCI functions via the Web Services Description Language (WSDL). The advantage of a web service is the fact that it can handle multiple clients at the same time but the disadvantage is the increasing network traffic used for the communication. For other connected simulators the ambassador has to be specified as well.

---

[2]Traffic Control Interface
[3]TraCI as a Service

For instance for the vehicle simulations in Matlab the implementation of TraCI4Matlab seems to be an appropriate choice.

To get a first impression of the necessary data flow between SUMO and the connected simulation suites a first simple code example with TraaS is given below.

```
SumoTraciConnection sumo = null;
sumo = new SumoTraciConnection(sumo_bin, config_file);
sumo.addOption("step-length", "0.1");
sumo.runServer();
for (int i = 0; i < 3600; i++) {

//example B
if (i % 20 == 0)
 sumo.do_job_set(Vehicle.add("v" + i, "car", "r1", i, 0, 13.8, (byte) 0));

for (int i1 = 0; i1 < 10; i1++) {
sumo.do_timestep(); //example A

//example C
SumoStringList sl = (SumoStringList) sumo.do_job_get(Vehicle.getIDList());
for (String vehID : sl) {
    SumoPosition2D pos = null;
    pos = (SumoPosition2D) sumo.do_job_get(Vehicle.getPosition(vehID));
  }
 }
}
sumo.close();
```

The code snippet shows a simple java TraCI client which handles a simulation with a total time of 3600 seconds and a time-step length of 100 milliseconds. There are 3 different kinds of TraCI communication commands implemented in this code. In *Example A* only a method is used to control the next simulation time step manually. In this case 0.8 MB were used for the communication with SUMO. *Example B* extends that simulation with a method to insert a vehicle every 20s into the running simulation. The additional used bytes for this new method are minimal so that the total data consumption is 0.97 MB. In the last *Example C* the code shows a typical approach for the simulations done in REMAS. Basically in every simulation time step the existing vehicles get listed. And the vehicle positions is gathered and transferred to the other simulators for instance to the vehicle simulator or the radio communications simulator. The data consumption is significantly higher with an usage of 10.6 MB. In Figure 4.3 the data consumption for the requests and responses for the 3 examples are illustrated.
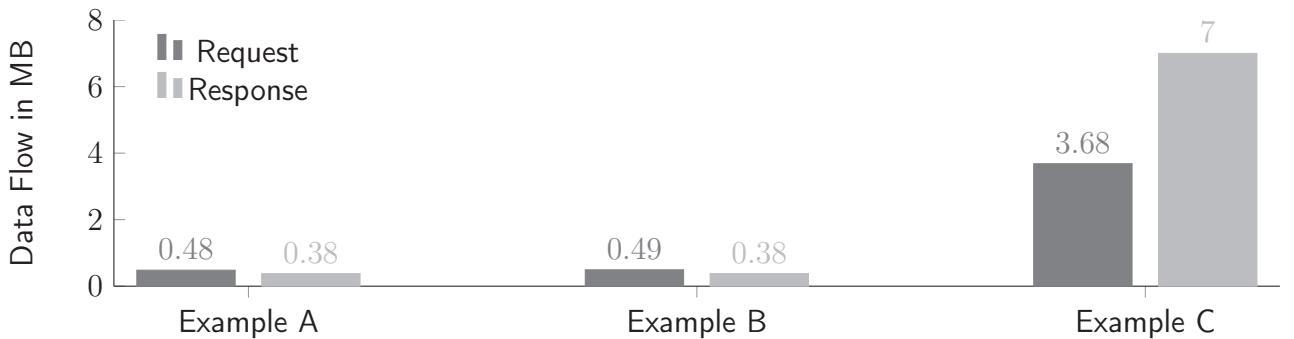
Figure 4.3: Date Flow within TraCl

4.5 Conclusion and Future Work

This paper presents a concept to support the developing and evaluation of new ADAS components for automated driving. We are still at the beginning of the project which means that there are a lot of work for the specification and implementation needs to be done. Several questions has to be answered for instance how fast the overall simulation process will be. How does the physical network should be designed to support that mass of communication messages between all the simulators. Keeping in mind that not all simulators will be operate on the same computer. To support the requirements of realistic simulations for automated driving in urban areas a lot of effort is necessary. In fact other non-equipped vehicles will influence the automated car as well as the accessibility of V2X networks or the accuracy of the positioning system will determine the effectiveness.

4.6 References

[1] *IEEE Standard for Modeling and Simulation High Level Architecture (HLA)– Federate Interface Specification*. IEEE Std 1516.1-2010 (Revision of IEEE Std 1516.1-2000), Aug 2010.

[2] D. KRAJZEWICZ, J. ERDMANN, M. BEHRISCH AND L. BIEKER: *Recent Development and Applications of SUMO - Simulation of Urban MObility*. International Journal On Advances in Systems and Measurements, 5(3&4):128–138, December 2012.

[3] HOBERT, L., A. FESTAG, I. LLATSER, L. ALTOMARE, F. VISINTAINER and A. KOVACS: *Enhancements of V2X communication in support of cooperative autonomous driving*. IEEE Communications Magazine, Dec 2015.

[4] KRIMMLING, JÜRGEN, RALF FRANKE, RALF ENGELMANN and MATTHIAS Kï¿½RNER: *Erfahrungen mit dem vollautomatischen baulastträgerübergreifenden Betrieb der Dynamischen Wegweisungskomponente im Operativen Straßenverkehrsmanagementsystem VAMOS*. In *HEUREKA 2011 – Optimierung in Verkehr und Transport*, 2011.

[5] MACEDO, JOSÉ LUÍS PEREIRA: *An Integrated Framework for Multi-Paradigm Traffic Simulation*, 2013.

[6] RIEBL, R., H. J. GUNTHER, C. FACCHI and L. WOLF: *Artery: Extending Veins for VANET applications*. In *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015 International Conference on*, June 2015.

[7] SCHUBERT, TOSRTEN, MARIO KRUMNOW, BERNARD BÄKER and JÜRGEN KRIMMLING: *Using Nanoscopic Simulations to validate the Benefit of Advanced Driver Assistance Systems in*

complex Traffic Scenarios. In *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2013 International Conference on*, June 2013.

[8] SOMMER, C. and F. DRESSLER: *Progressing toward realistic mobility models in VANET simulations*. IEEE Communications Magazine, 46(11):132–137, November 2008.

[9] TIMÓTEO, I.J.P.M., M.R. ARAÚJO, R.J.F. ROSSETTI and E.C. OLIVEIRA: *TraSMAPI: An API oriented towards Multi-Agent Systems real-time interaction with multiple Traffic Simulators*. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, Sept 2010.

[10] WILLE, J. M., F. SAUST and M. MAURER: *Stadtpilot: Driving autonomously on Braunschweig's inner ring road*. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010.

5 Experiment Study on the Evacuation of Bomb Alert with SUMO

Yun-Pang Flötteröd and Jakob Erdmann; German Aerospace Center
{Yun-Pang.Floetteroed, Jakob.Erdmann}@DLR.de

Abstract

In this paper, an evacuation scenario, based on the real bomb alert case in Brunswick on 20. July 2015, is modelled with SUMO. This is done to assess the capabilities of SUMO for modelling traffic during an evacuation with a special focus on dynamic route choices. Two influence factors (1) time to get information (2) probability to cancel a trip are used to examine the rerouter function in SUMO and to further understand the respective impacts on traffic during the evacuation. The experiment results show that the rerouter function in SUMO can in general deal with the proposed scenarios. Some issues related to route choice are discussed and further improvements are suggested.

Keywords: evacuation, rerouting, bomb alert, SUMO, DRIVER, AIM, VABENE

5.1 Introduction

The global weather change effects and human-caused major incidents such as terrorist and nuclear incidents have made crisis management become one of today's important issues. Lots of attention has already been paid to it. Many research studies have also been performed (1) to model different evacuation situations (2) to establish and develop the corresponding frameworks, (3) to examine and solve existing problems and obstructions, and (4) to develop, merge and evaluate possible solutions. Different simulative and operational tools for crisis management are already developed as well. In order to efficiently and effectively execute and manage different activities during crises it is necessary to coordinate existing crisis management tools and to have exercises and trainings as preparation for successful crisis management.

Several European research projects have also aimed to achieve the above mentioned goal for some years. The European project Driver (Driving Innovation in Crisis Management for European Resilience) is one of them. In this project, experiments/exercises for different crisis situations are designed and a part of them is already executed. The respective test-beds to coordinate tools for supporting the Driver experiments are planned and currently under development. One of the Driver experiments handled the bomb alert case in Brunswick, Germany, on July 20th 2015. Such cases happen several times in Germany every year and often results in an immediate evacuation which impacts the corresponding traffic system and the evacuation activities. Therefore, it is needed to understand the possible traffic impacts of the proposed evacuation and traffic management strategies for decision support. In this experiment, a simulation environment is established with SUMO to demonstrate the impact of the road closure for evacuation on traffic. Based on this experiment, two influence factors (1) time to get information (2) probability to cancel a trip are used to examine the current rerouter function in SUMO and to further understand the respective impacts on traffic during the evacuation.

5.2 Data Preparation

In order to simulate a bomb alert scenario in Brunswick, Germany the respective digital road network, traffic demand, the information about bus and the related traffic infrastructure are needed. With the consideration of the rerouting possibilities the analysis network covers not only the evacuation area but also the major roads, highways as well as the corresponding ramps in Brunswick. Based on the results of the DLR's fundamental research projects AIM (5; 4) and VABENE++ (6; 2) the analysis network is adjusted and the corresponding traffic demand, i.e. vehicular routes, in the analysis network are extracted from the original traffic demand in Brunswick. Figure 5.1 shows the layout of the analysis network. Furthermore, the information of the bomb alert case on July 20th 2015 is collected and summarized in Table 5.1. All the information is considered in the simulation as much as possible in order to closely reproduce the traffic situation during the evacuation of the bomb alert case.

To sum up, the evacuation area is 1 kilometer around the found bomb close to the main railway station. Some railway services were therefore adjusted or cancelled during the evacuation. Buses and trams were allowed to run in the evacuation area without stopping. The evacuation began at 17:05 and the total evacuation duration is 6.5 hours (from 17:05 to 23:36). More than 11.000 residents are evacuated. The evacuation area shows in Figure 5.1 as well.
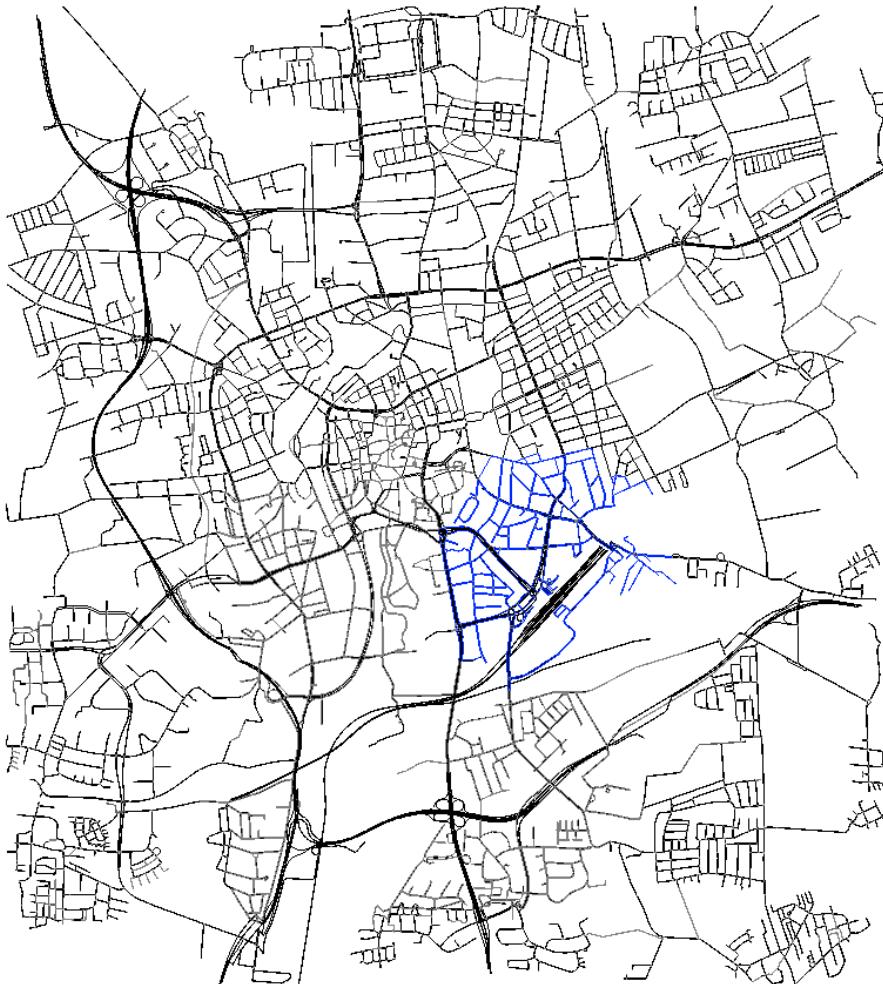
Figure 5.1: Analysis network of the bomb-alert scenario in Brunswick, Germany (The blue area is the evacuation area.)

5.3 Scenario Description

Beside the normal case (Base) five scenarios are developed to analyze the impacts of the following two factors
(1) the time when the road closure information is available to the drivers and
(2) the probability to cancel a trip in order to avoid the expected traffic congestion.
on traffic with SUMO.

Base (the normal case):

a common Monday without the bomb alert (The bomb alert in Brunswick happened on Monday.)

Scenario 1 (S1):

It is based on the normal case. All drivers receive the road closure information right after the road closure. The probability to cancel a trip on the way is 100% only for drivers who cannot reach their destinations when they are on the defined rerouting roads. Furthermore, some drivers may cancel their journeys in order to avoid the expected traffic congestion. In this case, the probability to cancel a trip on each rerouting road is defined as 10%.

Scenario 2 (S2):

It is based on Scenario 1. The probability to cancel a trip on each rerouting road is defined as 15% for avoiding the expected traffic congestion in this scenario.

Scenario 3 (S3):

It is based on Scenario 1. All drivers receive the road closure information right after the bomb alert, i.e. 15 minutes before the road closure, in this scenario.

Scenario 4 (S4):

It is based on Scenario 1. However, all drivers receive the road closure information right after the bomb alert, i.e. 15 minutes before the road closure, in this scenario. The probability to cancel a trip on the way is 50% only for drivers who cannot reach their destinations when they are on the defined rerouting roads. The probability to choose a new destination is 50% on each defined rerouting road for drivers who cannot reach their destinations. Drivers, who decide to change their destinations, choose a new destination from a pre-defined destination set.

Scenario 5 (S5):

It is based on Scenario 4. However, the probability to cancel a trip on each rerouting road is defined as 15% for avoiding the expected traffic congestion in this scenario.
Moreover, some common phenomena during the road closure as well as the evacuation are also considered in order to establish the simulation environment close to the reality. First, drivers in the evacuation area are allowed to leave the evacuation area in the first 10 minutes of the evacuation period. Second, drivers, who cannot leave the evacuation area in the first 10 minutes of the evacuation period, need to park their vehicles in the evacuation area and leave the area by foot or by shuttles provided by the rescue team. The later one will not be considered in the simulation. People, living in the evacuation area, are also not allowed to use their vehicles to leave the evacuation area

after the first 10 minutes of the evacuation period. Thus, no vehicle should move in the evacuation area after the first 10 minutes of the road closure.

The applied data in the simulation are summarized in Table 5.2.

5.4 Current Rerouting Function in SUMO

SUMO supports four mechanisms for location based rerouting of vehicles. When vehicles pass a pre-defined set of edges (referred to as *rerouting roads*) during a pre-defined time interval, they can take one or more of the following actions with the respective user-defined probabilities.

1. Pick a new route from a predefined distribution of routes.

2. Pick a new destination from a predefined distribution of destinations and then take the fastest route to the new destination.

3. Terminate their routes immediately.

4. Compute and use the fastest route to their original destination that avoids a pre-defined set of closed roads (if the original route does not include the closed roads, no action is taken).

Mechanisms 2, 3 and 4 can be combined and then vehicles which are affected by the closed roads can either pick a new destination and take the fastest route that avoids the closed roads or terminate their routes. According to the given probabilities, decisions can be made at each time when drivers are on a certain rerouting road. This method is employed to model road closure situations. Currently, the fastest route for a given vehicle is computed according to the traffic state without traffic loads, if users do not specify the traffic state, i.e. no information about edge weights. More corresponding information can be found at the SUMO-Rerouter website (9).

5.5 Implementation

To implement the above mentioned scenarios and phenomena in the simulation, the rerouter function in SUMO (9) is used. The evacuation begins at 17:05. In order to capture the traffic phenomena of the evacuation the simulation starts at 15:00 so that the traffic state corresponds to the normal traffic situation when the evacuation starts. Furthermore, different roads are defined as "rerouting" roads at different time intervals according to the scenarios. Once a driver is on a rerouting road, she receives the road closure information and has the opportunity to reroute the trip. When a driver decides to cancel her trip, the respective vehicle will be immediately removed from its current location. According to the scenarios, two rerouter styles "Closing a Street" and "Assigning a new Destination" in SUMO are applied.

In order to describe the situation in the simulation that drivers, who cannot leave the evacuation area in the first 10 minutes of the evacuation period, have to park their vehicles and leave by foot, the method terminateRoute in the rerouting function is used. With this method, the respective drivers terminate their routes immediately and are removed from the simulation.

Although SUMO can simulate traffic closely to the given reality, there are currently still some limitations when simulating drivers' "flexible" behaviors, for example standing in a wrong lane, reverse driving or driving to the roadside in order to provide enough gap at intersections so that dead-lock situations can be prevented. Such "flexible" driving behaviors are especially needed for evacuation. Otherwise many dead-locks will appear at the borders of the evacuation area or some major intersections with heavy traffic loads. Therefore the sumo-option "teleporting" is used in addition to allowing U-turns at intersections. 150 seconds and 15 seconds are set as the time to

teleporting on local roads and highway respectively in order to avoid the above mentioned issues and, at the same time, be still able to capture the overall traffic congestion phenomenon during the evacuation.

Table 5.3 shows the applied data and definitions in the simulation for each scenario.

5.6 Result analysis

The proposed scenarios with the above mentioned parameter settings are executed with SUMO (Version: dev-SVN-r20166). Based on the data in the output file tripinfo.xml the simulated travel information is firstly analysed and summarized in Table 5.4. Due to the nature of the proposed scenarios, i.e. the possibility to cancel a trip, the total number of vehicles varies in each scenario. Naturally, this variation also influences the average travel duration and the average travel length. In comparison to the normal case (Base) all other scenarios have less average travel times and shorter average travel lengths. With a trip cancellation the route length and the corresponding travel time is as a matter of course shorter than the original trip, since the vehicle is immediately removed from the location it is once a trip cancellation is decided. A new destination selection can also result in a shorter travel time and travel length. In addition, the whole traffic load with trip cancellations becomes less. The overall travel condition in all scenarios is therefore better than that in the base case. Thus, the negative impact of the road closure in the scenarios on traffic cannot be seen directly when only observing the changes in travel time and travel length.

In addition, it is noticed that some vehicles have a route length of zero in all evacuation scenarios. Their travel duration is 1 second. It means that they cancel their entire trip for avoiding the expected traffic congestion. It also explains why the total numbers of vehicles with a zero route length in S2 and S5 (with 15% of probability to cancel a trip) are considerably higher than those in other scenarios (with 10% of probability to cancel a trip). Such trips are not considered in the result analysis.

It is known that there is a apparent relationship between travel speed and the amount of emissions (1; 3). The more the driving speed deviates from the respective ideal driving speed, the larger the amount of emissions will be. Traffic jams occur during the road closure. Drivers who decide to continue their trips often experience lower travel speeds compared to the base case. The amount of the respective emissions should increase accordingly. The emission analysis, shown in Figure 5.2, supports this thought. Three emissions CO_2, CO and HC are chosen as indicators here. The amounts of the observed emissions for passenger car and trucks in all scenarios are higher than those in the normal situation (Base). When observing each scenario, S2 produced more emissions then S3, although the probability to cancel a trip (15%) in S2 is higher than that in S1. The similar result also appears when comparing S5 and S4. The amounts of emissions in S3 are slightly higher than those in S2, where drivers get the information before and after the road closure in S3 and S2 respectively. More vehicles in the network are affected by the evacuation when the respective information are earlier available. Therefore there is no improvement on the emission reduction. Moreover, S4 produced only slightly more emissions than S3, where the probability that driver who cannot reach their destinations cancelled their trips on each rerouting road is 100% and 50% in S3 and S4 respectively.

5.7 Conclusion and discussions

The human behaviors during evacuations are quite complex. It is thus a challenge to model such behaviors properly, especially when empirical data are not available all the time. Such behaviors influence many aspects in traffic modeling and simulation, such as traffic demand, selected traffic modes, trip-making decision, departure decision, route choice decision, driving behaviors and so
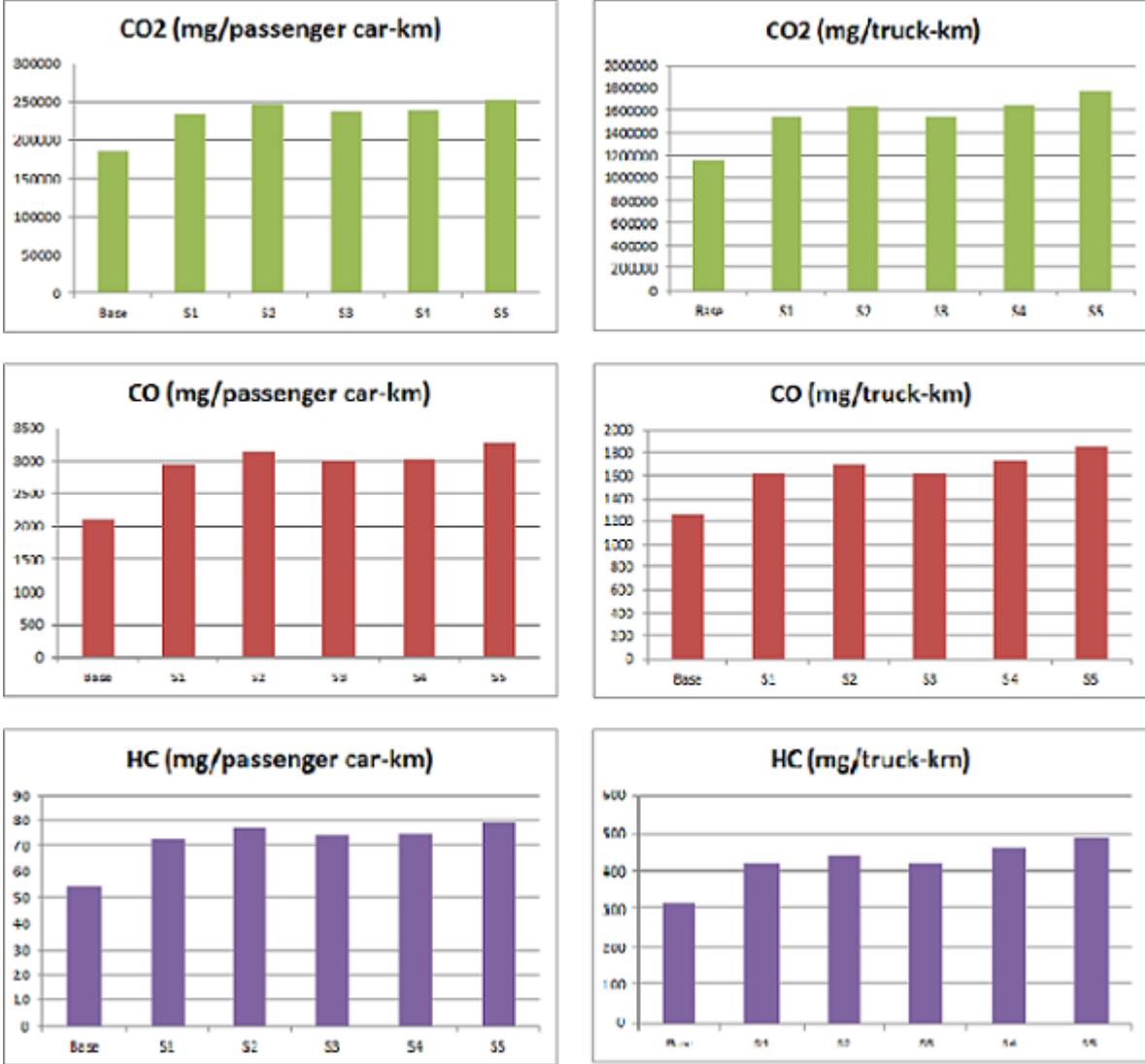
Figure 5.2: Simulated emissions for all scenarios

on. This study simplified the analysis situation and some assumptions are made in order to show a simple example of an evacuation situation with the existing functions in SUMO and to examine the rerouting function in SUMO. The experiment results show that SUMO can basically simulate the proposed scenarios properly. However, some issues were discovered during the experiment. It seems that a higher probability to cancel a trip results in a higher emission production when comparing Scenario 1 and 2 as well as Scenario 4 and 5. It does not correspond to the expectation. Further investigation is needed. Vehicles which already stand on the defined rerouting roads instead of driving into these rerouting roads are not properly affected by the pre-defined actions, such as route termination or U-turn making. It makes the existing traffic congestions worse. Moreover, vehicles with the destinations outside of the evacuation area will also be rerouted when combing the methods closingReroute and destProbReroute in the rerouter function. Some of vehicles can indeed change their routes due to the expected traffic congestion even though their routes are not really affected by road closure. However, it is not appropriate that all of such drivers change their routes. These issues are taken care of currently. Sometime, a certain amount of shelters and the corresponding routes to the shelters are given during evacuations, especially during serious disasters, such as nuclear disasters. In this case, the combination of the rerouter function and a given route set is necessary in the simulation. Therefore, such feature is suggested to be implemented in SUMO.

Last but not least is the routing issue. The route choice modeling can basically be divided into

the pre-trip route choice modeling, the en-route route choice modeling and the hybrid route choice modeling which is based on the combination of pre-trips and en-route route decisions. The pre-trip route choice modeling is to determine routes based on the current or expected route utilities, i.e. travel time in this study. The principle to determine such routes is nowadays often based on the Wardrop's user equilibrium(10). In our example, these routes are already given. In the en-route model, drivers can make a new route choice according to the available traffic information when they approach an decision-making point, for example intersection and ramps. In general, the hybrid route choice model takes the unfamiliarity of the traffic information into consideration. Drivers choose their routes according to their departure times. They can then adjust their routes according to the available information. More information about these three models and the review of the dynamic evacuation models can be found in (8). Currently, the routing concept of the evacuation in SUMO is similar to the hybrid route choice model. However, some factors, such as the unfamiliarity and the reliability of road situations and road classes, drivers' time perceptions as well as departure time choosing are not taken into consideration in SUMO yet. In SUMO, users can define the available new routes and the corresponding usage probabilities so that drivers can choose a route from the given route set. Moreover, drivers can also check for a new route, i.e. the fastest route to his/her destination, when they are on a pre-defined rerouting road and the new route will be used. The new route calculation is based on the traffic state in a empty network if no additional information is given by users and no TraCI function in SUMO is used. It results in a static route search, i.e. the route to the same destination will not be changed when searching for a new route. Therefore, the respective route choice cannot be modeled properly. This drawback will be improved in the future. In the current SUMO-version, it is possible to define a specific set of vehicles or a certain percentage of vehicles equipped with navigation devices so that vehicles can adjust their routes with the traffic state in the pre-defined intervals. The combination of the use of navigation devices and the rerouter function can improve the static route search problem.

5.8 References

[1] Behrisch, Michael, Yun-Pang Flötteröd, Daniel Krajzewicz and Peter Wagner: *Ecological user equilibriumin traffic management?* In *4th International Dynamic Traffic Assignment Symposium*, Martha's Vineyard, Massachusetts, USA, June 2012.

[2] Detzer, Sandra and Melanie Weber: *Case study: Simulation of Transport Systems in a critical situation in Brunswick, Germany.* In *ISCRAM2015 Conference Proceedings*, Kristiansand, Norway, May 2015.

[3] Flötteröd, Yun-Pang, Peter Wagner, Michael Behrisch and Daniel Krajzewicz: *Simulation-based Validity Analysis of Ecological User Equilibrium.* In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, Berlin, Germany, December 2012.

[4] Institute of Transportation Systems of the German Aerospace Center: *Application Platform for Intelligent Mobility web site.* Accessed 2016, 2015.

[5] Institute of Transportation Systems of the German Aerospace Center: *Anwendungsplattform Intelligente Mobilität web site.* Accessed 2016, 2016.

[6] Institute of Transportation Systems of the German Aerospace Center: *VABENE++ Traffic Management for Large Scale Events and Disasters web site.* Accessed 2016, 2016.

[7] NDR: *Live-Ticker: Bombe in Braunschweig entschärft web site.* Accessed 2016, 2015.

[8] Pel, Adam J., Michiel C. J. Bliemer and Serge P. Hoogendoorn: *A review on travel behaviour modelling in dynamic traffic simulation models for evacuations*. Transportation, (39):97–123, 2012.

[9] SUMO: *SUMO: Simulation/Rerouter web site*. Accessed 2016, 2016.

[10] Wardrop, J. G.: *Some theoretical aspects of road traffic research*. In *Proceedings of the Institute of Civil Engineers*, volume 1, Delft, The Netherlands, 1952.

Table 5.1: Basic data of the bomb-alert case in Brunswick, Germany, on July 20th 2015

Item	Content	Note
Bomb location	in the BraWo-area next to the main railway station Brunswick	
Evacuation period	from 16:50 to 21:50	It is planned to evacuate the area until 19 O'clock. The delay was due to some residents refusing to leave their places.
Closed area	1 Km around the found bomb	
Period of the area closure	from 17:05 to 23:36	
Bus and trams	ran without stopping in the evacuation area	
Number of the evacuated people	11.000 people	
Trains	The regional trains and the long-distance trains stopped to run from 17 O'clock and from 22:05 on respectively.	
Shelter	Volkswagenhalle at the Europaplatz	capacity: 1.000-1.200 residents
Action force	around 600 people from the Fire Department	
Begin of the bomb deactivation	22:34	
End of the evacuation	Residents could go back home from 23:36 on	

source: (7)

Table 5.2: Factors considered in the scenarios

Scenario	Time to get the information right after		Probability to cancel a trip with a unreachable destination		Probability to cancel a trip in oder to avoid the traffic jams	
	road closure	bomb alert	50%	100%	10%	15%
Base (normal case)	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
S1	X			X	X	
S2	X			X		X
S3		X		X	X	
S4		X	X		X	
S5		X	X			X

Table 5.3: Overview of the applied data in the simulation

Item	Content	Note
Simulation begin time	15:00	
Traffic demand	146.976 vehicles from 15:00 to the end of the day	The respective vehicle routes are extracted from the original vehicle routes.
Time to start the evacuation and the road closure	17:05	
Time to disallow vehicles to drive within the evacuation area	17:15	no moving vehicles in the evacuation area
Time to start rerouting	See Table 5.2	
Rerouting roads	all roads except the roads in the evacuation area and the mainlines on the highways	
Possibility to cancel a journey during a trip on each rerouting road (for drivers who cannot reach their destinations)	See Table 5.2	
Possibility to cancel a journey on each rerouting road	See Table 5.2	
Time to teleporting	150 seconds for local roads and 15 seconds for highways	

Table 5.4: Simulated traffic times and lengths for all scenarios

	Base	S1	S2	S3	S4	S5
total number of vehicles in the network	146976	127151	122396	125780	126517	121473
total number of vehicles with a route length = 0	0	17865	22620	19060	18323	23367
average travel duration (minutes/veh)	4.94	3.38	3.49	3.17	3.13	3.22
average travel length (km/veh)	3.37	2.26	2.32	2.13	2.10	2.14

6 Distributed simulations using SUMO

Dídac Busquets, Raymond Francis, Christos Tsotskas, Robin North, Fabio Galatioto, Patrizia Franco, Roland Guichard, Richard Tusting, Ecaterina McCormick, Graham Ravenscroft, Graham Fletcher, Eifion Jenkins
Transport Systems Catapult, MK9 1BP Milton Keynes, United Kingdom
{firstname.lastname@ts.catapult.org.uk}

Abstract

The increasing availability of multiple sets of data and the need to solve complex problems that involves different focus and wide areas allow urban and traffic planners to choose from a variety of different modelling tools, ranging from transport modelling suites to standalone simulation models. Each of them is highly specialised but works independently with a considerable increase of running time when large scenarios are simulated. This issue could be overcome by looking at the integration of different tools as well as the ability to simulate large scenarios in less time. While some existing software packages provide some of the means of integration, this is usually a bespoke integration with another package, which limits the interoperability with diverse tools and models. Also, so far no one has attempted to run in an integrated way several micro-simulators for a unique larger area simulation. This is because the integration and redistribution over several simulators is a major issue. In this paper we present an architecture and methodology to distribute simulations across several simulators to address this issue, which can be used to simulate large scale urban scenarios.

6.1 Introduction

The need for large scale simulation over an entire city or region has been addressed so far to some extent at a macroscale level, by distributing model processes or blocks of one or more independent modelling steps (10; 6). This has been done by assigning the processes across several available processors, but always using the same simulation software package or suite. While using macroscale simulation is useful to get aggregated data, there might be a need to simulate individual agents at a microscale level in large areas (e.g. to analyse end-to-end journeys). However, due to the computational complexity and large scale of the scenarios to be modelled, the distributed technique has seen only few attempts almost a decade ago, implemented with traditional microsimulators (4; 9). Also, at microscale and sometimes even at macroscale, run time of large simulations can easily go over days and thus the possibility of distributed simulation at that scale would be highly appealing and may represent a new frontier of the traffic modelling simulation. Even at macroscale level, the distributed simulation cannot be done at present using different simulation packages modelling independently different modes, networks, time periods, etc.

With the increasing demand of addressing complex problems in transport modelling, the necessity of integrating different software tools comes together with the ambition to execute them in the most appropriate and efficient way. Most of the time commercial tools have to be integrated with bespoke software to allow large scale distribution to be simulated. Several approaches have addressed this. More recently, (8) proposed in the MERGE project to build a complex simulation program: a component-based client-server programming model which couple a simulation component with a

Data coupling library (SCDC). Performance results demonstrated that the distributed execution of simulation applications on several compute nodes is a quicker solution.

In this paper we present an architecture and methodology to distribute large scale simulations across several simulators by breaking down the area being simulated into smaller ones and assigning each of these areas to individual simulators. This reduces the computational workload of each of the simulators, allowing for faster computation.

The paper is organised as follows. In the next section we briefly discuss existing work on integration tools and platforms as well as methods to distribute simulations. We then present our proposed architecture, based on the High Level Architecture (HLA), and how the simulation is actually distributed by means of a data ownership manager. We present the results of some performance tests on the scalability of SUMO, and a simple test case to illustrate the functionality of our platform. We finally draw some conclusions and discuss future work we need to address.

6.2 Related work

6.2.1 Integration tools

Transport modellers have a variety of options when choosing tools for modelling urban environments. This includes vehicle micro-simulators such as Vissim (13), Aimsun (20), or Paramics (19), available as commercial products, or SUMO (5) as an open-source alternative; vehicle macro-simulators such as Saturn (2) or Visum (16); pedestrian simulators such as Viswalk (17), Legion (12) or MassMotion (15), just to name a few commercial ones; or activity based modelling tools such as MATSim (14). While all these tools are scientifically validated and widely used by modellers and transportation bodies, they are usually focused on some specific transportation mode (e.g. either vehicle or pedestrian) and used as stand-alone products, with only limited integration between them. When possible, such integration is usually bespoke and just one-to-one (e.g. Vissim and Viswalk, both from the same company; or Aimsun-Legion, from different companies), thus making it very difficult to integrate three or more components, which might be needed for modelling realistic and complex simulation scenarios.

Nevertheless, integration platforms have been developed to allow for the combination of several components. In particular, the VSimRTI platform (18) uses a similar approach to ours, using ideas based upon the High Level Architecture (HLA), which will be briefly described in the next section, to provide a flexible way of integrating different components. However, the platform is limited to some of the functionalities of HLA, while our approach utilises the full HLA standard, thus making it fully compatible with any other HLA-enabled platform, as well as making any HLA-compliant component compatible with our platform, being a host for any HLA-compliant component.

Over the last 2 decades, only few examples of HLA implementation in the Transport modelling field have been attempted (10; 11; 7). However, so far no large scale implementation or real-world testing with this novel method has been performed.

6.2.2 Distributing simulations

Another limitation of standard micro-simulation tools used by transport modellers and practitioners is that they simulate relatively small scenarios such as junctions, or corridors of limited length, but lack scalability to simulate large areas such as whole cities, encompassing far too many entities (e.g. vehicles and/or pedestrians) and for which the simulation is then computationally very intensive and, consequently, expensive to run. A solution to this problem is to break the simulated area down to smaller ones, each being described by an individual simulator and running simultaneously and in synchronicity with the others. A platform as the one we present in this paper enables such a
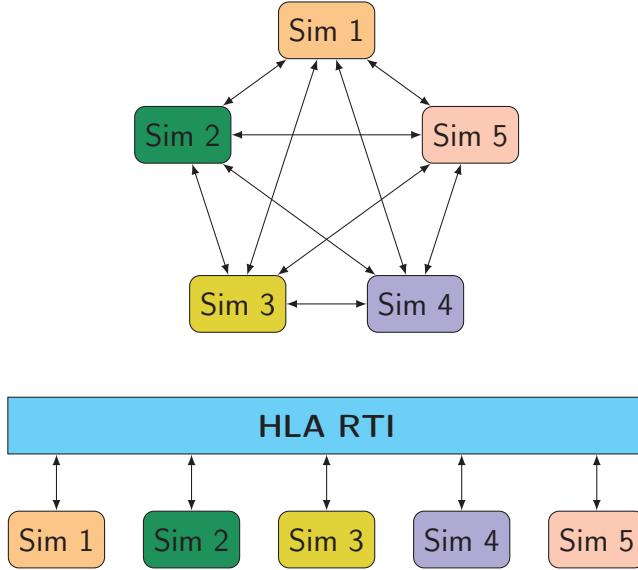
Figure 6.1: Connecting several simulators with one-to-one interfaces (top) or through HLA (bottom)

distributed simulation, since the simulators in charge of all smaller areas can easily be run in an integrated manner.

Scaling up, in terms of number of individual simulations, an assembly of simulations that is based on HLA with multiple components is very advantageous, as the number of components grows. This is clearly illustrated in Figure 6.1. In the case of an assembly with five components, establishing an all-to-all communication (top diagram in the figure) requires to develop several bespoke individual connections. Note that by "connection" we refer to having the ability of components developed by different providers (e.g. SUMO, Aimsun, VISSIM, etc.) to speak to each other, not to the actual connections between the simulators for a specific area, which might depend on neighbouring relationships. Moreover, should we consider adding a new component, we should then develop five new connection interfaces. Conversely, under an HLA federation (bottom diagram in the figure), each federate just connects to the RTI (Run Time Interface), and adding a new component requires the development of only one connection interface. This has a great impact on the development and operational costs, also in the long term. Considering the workflows of a typical software development life cycle and structuring a policy, it is easy to notice a clear reduction in time, human and physical resource utilisation. Regarding connectivity and quality of services, it is important to scale a simulation over HLA even from 4 simulators, which will be even more obvious in large scale simulations.

There has already been some development on distributing SUMO, dSUMO (3). Although the main concept of our approach is very similar to that of dSUMO, the latter has been developed solely for SUMO, while ours, being based on HLA, is independent of the micro-simulator being used. In fact, with our approach we could distribute simulations over micro-simulators from different providers.

6.3 Our approach: HLA architecture

6.3.1 Overview of HLA

A common approach to deal with integrated simulations in military and aerospace industries is to use High Level Architecture (IEEE standard 1516e) (1). HLA is based on the premise that no single monolithic simulation can satisfy the needs of all users and that it is not possible to anticipate the combinations of model components that users will find useful. HLA therefore seeks to establish a common development and execution architecture to facilitate the interoperability of all types
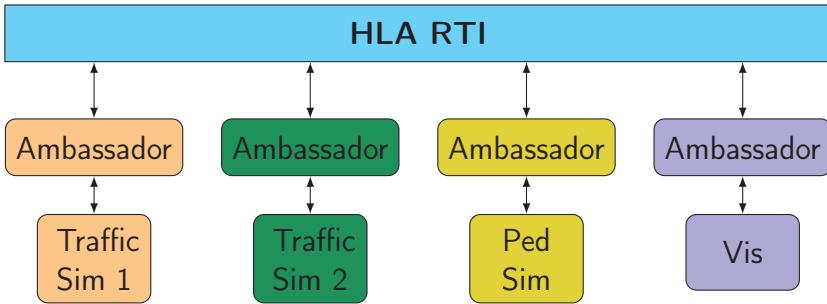
Figure 6.2: Integration of two traffic simulators, one pedestrian simulator and a visualisation component through HLA

of models and simulations including real-time, faster than real-time, event-driven simulations and supply-side management systems.

The whole system is called a federation in HLA terminology, and each component is a federate. A federate is connected to the Run Time Interface (RTI) through an Ambassador (see Figure 6.2), which is responsible for handling all communications between the federate and RTI. Internally, there are actually two ambassadors, the RTI Ambassador, which handles outgoing communication (i.e. from the federate to RTI), and the Federate Ambassador, which handles incoming communication (i.e. from RTI to the federate).

HLA provides a "Publish-Subscribe" data exchange between model components using a shared networked Run Time Interface (RTI). By placing the communications intelligence in the low level communications module rather than in each simulation component, the simulation components may publish all the data that may be required in any application, incurring a low overhead to do so. Only when another simulation component subscribes to it, is the data transferred, which allows each component to be prolific in the data it publishes leading to more flexibility in how data is then used in the simulation. Moreover, this keeps each component independent of each other, since they only have to be able to send and receive information to the RTI, regardless of what other component will receive or send such information.

6.3.2 Our HLA platform

In order to use simulators within the HLA platform, we need to connect them to the RTI. We achieve this connection by "wrapping" each simulator into a component that will then be part of a federate representing that simulator. This micro simulator wrapper is in charge of communicating with the actual simulator (usually through some API that is provided by the simulator's developer) as well as connecting to the Federate Ambassador so that the data coming out from the simulator can be published to the RTI, and any data coming from the RTI can be passed to the simulator (see Figure 6.3).

An important aspect when integrating different simulators is to define a common representation of the involved entities (e.g. vehicles, pedestrians, etc.). In HLA this is achieved by the Federation Object Model (FOM), which defines how each entity or object is represented (e.g. a vehicle is defined by an identifier, its current speed, its current position, etc.). Each micro simulator wrapper is then in charge of converting the specific representation of objects in the simulator into the FOM representation. This common representation of the objects provides interoperability between different simulators, since they do not need to know how a particular simulator represents an object, but they all rely on the common representation.

More details on the architecture of our HLA platform can be found in (21).
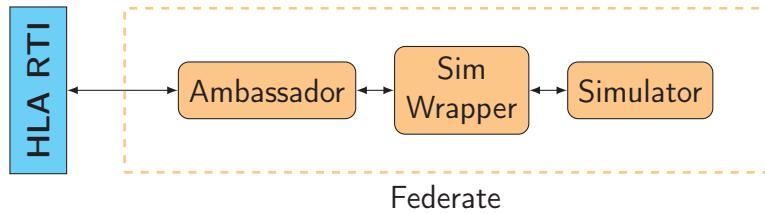
Figure 6.3: Connecting a simulator to HLA through a wrapper
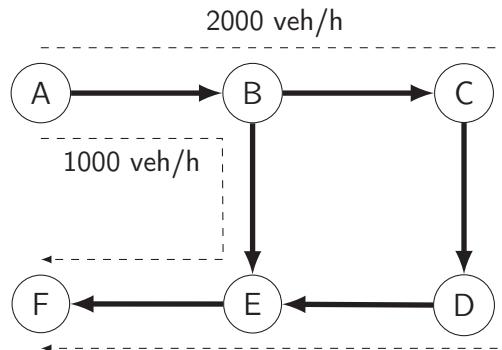
Figure 6.4: Base network used for performance tests

6.4 Performance tests

To better assess the need of distributing large simulations into smaller ones, we have run some performance tests on networks of different sizes. We have used a base network (see Figure 6.4) composed of 6 nodes and 6 edges, with two flows of vehicles, one driving along edges $\langle A, B, E, F \rangle$ (1000 vehicles in 1 hour) and the other along $\langle A, B, C, D, E, F \rangle$ (2000 vehicles in 1 hour).

The base network has been used to generate larger networks by simply replicating it a number of times (and shifting it so that they do not overlap). Although the resulting network is a collection of disconnected networks, and so not representative of an actual city scale network, it is valid for our performance tests.

We have replicated the base network starting with only one replica (i.e. the base network) up to 100 replicas. We have run 10 simulations (of one simulated hour) for each number of replicas, and computed the average time to finish the simulation. Figure 6.5 shows the results, where we can observe that there is a linear increase in time as a function of the size of the network.

These results are an indication that by splitting the simulation into smaller ones and running them concurrently a considerable speed up in the time taken to simulate the whole simulation could be achieved.

6.5 Managing the simulations: handing-off entities

By using HLA we can introduce the concept of ownership of entities and entity attributes within a simulation. Ownership is the idea that one given simulator takes control of an entity or a subset of its attributes. In any HLA federation, at most one simulator can own any given attribute of an entity at any given time. For simplicity, in our initial implementation, a simulator can only own all attributes of a given entity. This means only whole entities will change ownership from one simulator to another.
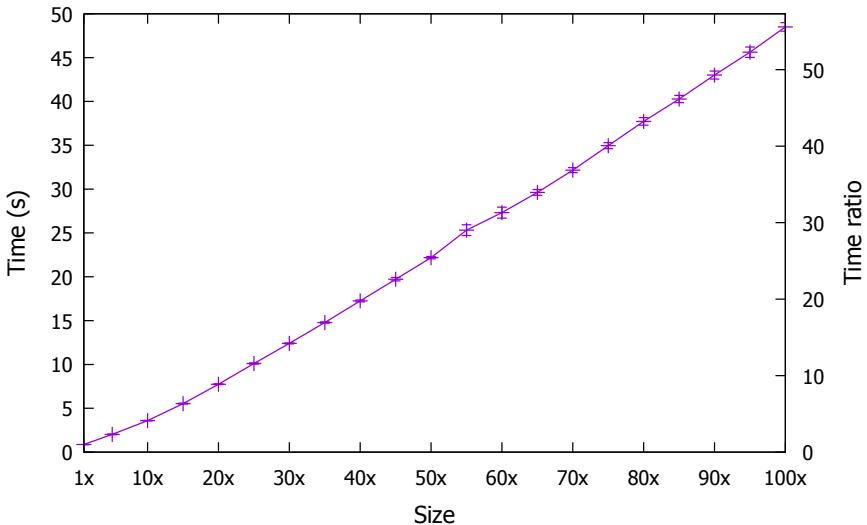
Figure 6.5: Simulation time (left vertical axis) and ratio (right vertical axis) as a function of the size of the network

6.5.1 Determining ownership

To determine when an entity is owned by a given simulator, each simulator needs to define a set of conditions such that when the conditions are satisfied by an entity, that simulator is a potential owner of that entity. To ensure that an entity is only owned by one simulator we need the following concepts:

- A single component to manage which entity is owned by which simulator (Ownership Controller).

- A unique identifier within the federation for each simulator (so that the Ownership Controller can identify a specific simulator in ownership handover).

- A unique identifier within the federation for each entity of a given type (so that the Ownership Controller can identify a specific entity in ownership handover).

Each simulator is configured with a set of ownership conditions (such as "within an area of interest"). These conditions are then broadcasted (via HLA) to the Ownership Controller, informing the controller of the simulator (identified by its federation wide unique identifier) and the conditions required on what type of data that means the simulator can potentially take ownership (for example, a vehicle simulator is only interested in ownership of vehicle data within a square kilometre of the origin).

To ensure a smooth handover, ownership conditions will be overlapping within the federation and provide three responses:

1. **Priority ownership**: the entity is owned by a simulator (where the ownership condition is exclusively satisfied by the entity or is satisfied and takes priority in overlaps).

2. **Low priority ownership**: the entity is owned by a simulator, but can be given over to another simulator if available (where the ownership condition is satisfied by the entity, but should not take priority in overlaps).

3. **No ownership**: the entity cannot be owned by the simulator (ownership condition is not satisfied by the entity).
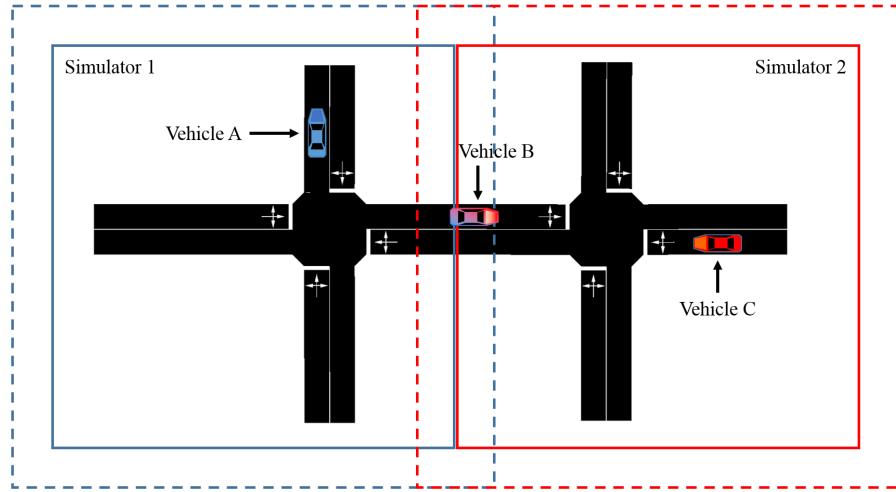
Figure 6.6: Example of areas of interest in ownership: Simulator 1 (left hand side, in blue) is in charge of part of the network, while Simulator 2 (right hand side, in red) is in charge of the other part. The border (in dashed lines) of each area of interest overlaps with the other simulator's area

So, if, for example, ownership of an entity is determined by areas of interest, each ownership condition is defined by an exclusive area of interest (such as cells in a grid) with an additional border area around the cell (see Figure 6.6). Entities falling in the cell satisfy condition 1 above (e.g. Vehicle A in Figure 6.6, with respect to Simulator 1). Entities that fall outside the cell but inside the border region satisfy condition 2 above (e.g. Vehicle B in Figure 6.6, with respect to Simulator 1). Entities that fall outside the cell and the border region fail to satisfy the condition, point 3 above (e.g. Vehicle C in Figure 6.6, with respect to Simulator 1).

The Ownership Controller maintains a list of all ownership conditions for simulators and searches that list for conditions that are satisfied for each owned entity it receives over HLA. If there is any condition that gives priority ownership of an entity to any simulator that is different from the entity's currently owning simulator, an ownership change request is sent over HLA to the federation for the currently owning simulator to release the ownership of the entity. The new owning simulator then receives information from HLA that the entity is available for ownership and claims it. The new simulator can then publish the updated entity values to the federation.

6.5.2 Ownership conditions

Defining ownership conditions is not limited to areas of interest (as shown in the example in Figure 6.6), but it can include any attribute of the entity that should fulfil the condition. For instance, if the entity is of type vehicle, we could use conditions on speed, so we could define a given simulator to own vehicles driving above a certain speed, while slower vehicles could be owned by a different simulator. Moreover, a condition can be a logical combination of other conditions (e.g. being in a given area of interest and not exceeding a given speed).

Figure 6.7 shows the different type of conditions and their relationships. A data condition can be of one of the following types:

- **Within shape**: the condition will be satisfied if the object location is within a given shape (represented as a polygon). This would be the appropriate condition to define that a simulator owns any object within a given area (e.g. as in Figure 6.6).

- **Field condition**: it is defined as $\langle field\ name, comparator, value \rangle$. The condition will be satisfied if the comparison (according to $comparator$, e.g. $>, \geq, <, \leq, =, \neq$) between the
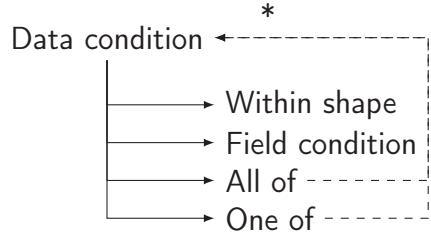
Figure 6.7: Relationships of data conditions for onwnership
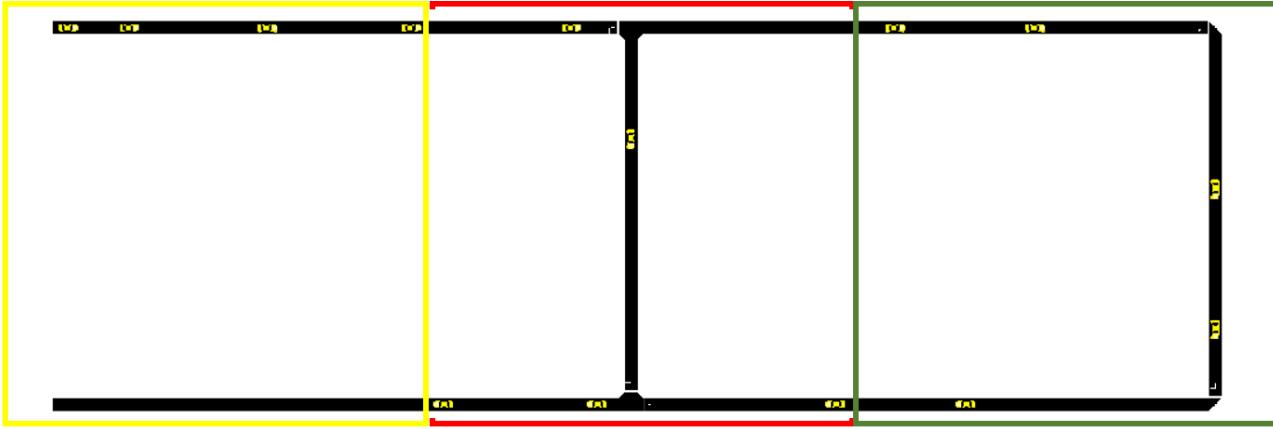
Figure 6.8: Network used for the proof of concept and the three regions, each of which is controlled by an independent SUMO instance

value of the attribute $field\ name$ of the object and $value$ evaluates to true. For example, we would use $\langle speed, \leq, 20\ m/s \rangle$ to indicate that the condition will be satisfied by those objects with a speed of 20 m/s or less.

- **All of**: this condition is represented as a list of conditions (of any type), and it will be satisfied if all its subconditions are satisfied by the object. For instance, we could have an *all of* condition composed of a *within shape* condition and a *field* condition.

- **One of**: this condition is also represented as a list of conditions, but it will be satisfied if at least one of its subconditions is satisfied by the object.

6.6 Proof of concept

We have developed the presented platform as well as a wrapper for the SUMO micro simulator. To test our platform we have used a simple network as shown in Figure 6.8. We have defined a federation composed of 3 instances of SUMO as well as the ownership controller component. Each of the 3 SUMOs is in charge of a region of the test network (the square regions depicted in Figure 6.8).

Each of the 3 simulators is only simulating the vehicles corresponding to its control region, thus the computational load of each of them is much lower than that of a stand-alone simulation controlling all vehicles. While in this example we would not gain much by distributing the simulation, in a large scenarios this might be quite significant. If we split a large network into N regions each controlled by one simulator, on average each simulator should take care of only 1/N of the vehicles, which would speed up the computation.

However, there are a few issues that should be further studied to better understand the potential gains of distributing the simulation over HLA. Firstly, we should study whether the overhead due to the use of the HLA architecture (both in terms of communication and computation) cancels out the speed up obtained by the lower number of entities being simulated by each simulator. Secondly, in the example shown above each simulator only takes into account the vehicles under its control. However, they should also consider the vehicles in the adjacent areas, so that vehicles near an area border can react to vehicles in the other side of the border. This would imply having a neighbour-to-neighbour communication, so that each simulator is aware of the vehicles around it. While this would make each simulator to represent vehicles outside its control area, this would only be limited since it should not need to know about vehicles far away, and so the reduction in number of vehicles being simulated (compared to a stand-alone simulation) would still be significant.

6.7 Conclusions and further work

With the increase of urban population and the advance in technology in the area of mobility, the standard practice of simulating small areas is not the most appropriate to answer the new questions and challenges posed by the new urban environment. One of the key points is to be able to simulate larger areas in order to consider a bigger picture instead of localised problems such as junctions or corridors.

In this paper we have presented an approach to distribute a large simulation amongst a set of simulators to balance the computational load. Consequently, the whole simulation can run faster than as a stand-alone one. In order to do so we have used the High Level Architecture, taking inspiration from the military and aerospace fields, which provides an integration platform where different simulators can be easily connected. Having proved that our platform can be used to distribute a simulation, our next steps are to make a detailed performance analysis to better understand what are the quantitative gains, as well as limitations, of splitting a simulation into smaller ones.

Moreover, such an integration platform not only allows distributing a large simulation, but also combines several independent components into the same simulation. This is going to be a requirement if we want to model and simulate the forthcoming challenges in urban transportation. The current approach of using a specialised simulator to model a particular aspect of urban transport (e.g. only vehicles, only pedestrians) will no longer be useful to deal with the increasingly complex interactions among the different actors (e.g. private vehicles, pedestrians, autonomous vehicles, bicycles). Furthermore, non-transport related systems (e.g. energy, weather) will also play a crucial role in accurately modelling the environment and so they should be easy to integrate within the platform.

6.8 References

[1] *IEEE Standard for Modeling and Simulation High Level Architecture (HLA)– Federate Interface Specification*. IEEE Std 1516.1-2010 (Revision of IEEE Std 1516.1-2000), Aug 2010.

[2] ATKINS: *Saturn manual.* http://www.saturnsoftware.co.uk/saturnmanual, 2016.

[3] BRAGARD, QUENTIN, ANTHONY VENTRESQUE and LIAM MURPHY: *dSUMO: Towards a Distributed SUMO*. In *The first SUMO User Conference (SUMO2013)*, Berlin, Germany, 05/2013 2013.

[4] CETIN, N., A. BURRI and K. NAGEL: *A large-scale agent-based traffic microsimulation based on queue model*. In *IN PROCEEDINGS OF SWISS TRANSPORT RESEARCH CONFERENCE (STRC), MONTE VERITA, CH*, 2003.

[5] D. Krajzewicz, J. Erdmann, M. Behrisch and L. Bieker: *Recent Development and Applications of SUMO - Simulation of Urban MObility*. International Journal On Advances in Systems and Measurements, 5(3&4):128–138, December 2012.

[6] Fujimoto, Richard M.: *Parallel and Distribution Simulation Systems*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1999.

[7] Gao, Peng, Chun Jin and Gang Wang: *HLA-based distributed simulation model for multimodal operation system on container terminals*. In *System Simulation and Scientific Computing, 2008. ICSC 2008. Asia Simulation Conference-7th International Conference on*, pages 338–343. IEEE, 2008.

[8] Hofmann, M. and G. Rünger: *Sustainability through flexibility: Building complex simulation programs for distributed computing systems*. Simulation Modelling Practice and Theory, 58:65–78, 2015.

[9] Klefstad, R., Y. Zhang, M. Lai, R. Jayakrishnan and R. Lavanya: *A Scalable Synchronized, and Distributed Framework for Large-Scale Microscopic Traffic Simualation*. In *Proc. of the 8th Int'l IEEE Conference on Intelligent Transportation Systems, Vienna, Austria*, 2005.

[10] Klein, Ulrich, Thomas Schulze, Steffen Strassburger and Hans peter Menzler: *Distributed Traffic Simulation based on the High Level Architecture*. In *In Proceedings of the Simulation Interoperability Workshop*, 1998.

[11] Lee, Jong-Keun, Min-Woo Lee and Sung-Do Chi: *DEVS/HLA-Based Modeling and Simulation for Intelligent Transportation Systems*. Simulation, 79(8):423–439, 2003.

[12] Legion: *Legion Science in Motion*. http://www.legion.com, 2016. [Online; accessed November-2014].

[13] Lownes, N. E. and R. B. Machemehl: *VISSIM: a multi-parameter sensitivity analysis*. In *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pages 1406–1413. IEEE, 2006.

[14] MATSim development team: *MATSIM-T: Aims, approach and implementation*. http://www.matsim.org/uploads/MATSim_overview_0.pdf, 2007.

[15] Oasys Software: *MassMotion*. http://www.oasys-software.com/products/engineering/massmotion.html, 2016.

[16] PTV: *Visum*. http://vision-traffic.ptvgroup.com/en-uk/products/ptv-visum, 2016.

[17] PTV: *Viswalk*. http://vision-traffic.ptvgroup.com/en-uk/products/ptv-viswalk, 2016.

[18] Schünemann, Björn: *V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems*. Computer Networks, 55(14):3189–3198, 2011.

[19] SIAS Ltd: *Paramics*. www.sias.com/paramics, 2016.

[20] Transport Simulation Systems: *Aimsun*. https://www.aimsun.com, 2016.

[21] Tsotskas, Christos et al.: *The design and development of an HLA-based open platform to model urban environments in SUMO*. In *SUMO User Conference (SUMO 2016)*, Berlin and Germany, 2016.

7 Distributed Simulation in SUMO Revisited: Strategies for Network Partitioning and Border Edges Management

Andrés Acosta, Jairo Espinosa; Universidad Nacional de Colombia
{afacostag, jespinov}@unal.edu.co
Jorge Espinosa; Politécnico Colombiano Jaime Isaza Cadavid
jeespinosa@elpoli.edu.co

Abstract

Since 2013 the project Modelling and Control of Urban Traffic in the City of Medellin (MOYCOT), has produced valuable results regarding the modelling, simulation and control of multimodal urban traffic using the SUMO simulator. Moreover, several software tools that simplify the setup of simulation scenarios and by taking advantage of the Matlab scientific computing environment have been developed. This time, the interest is on achieving a distributed road traffic simulation in SUMO. For this purpose, an approach for network partitioning and border edges management in free flow and congestion scenarios, aimed to the implementation of a distributed simulation platform is introduced. A detailed description on how to perform these tasks using software tools included in the SUMO package is presented. Furthermore, three strategies for simulating border edges in congestion scenarios are investigated.

Keywords: Distributed traffic simulation, network partitioning.

7.1 Introduction

This paper describes ongoing work in the Modelling and Simulation of Urban Traffic in the city of Medellin (MOYCOT) project, focused on strategies for network partitioning and border edges management in distributed traffic simulations. The goals of the second stage of the MOYCOT project are focused in the following topics:

- **Model validation and model reduction:** Models for multimodal traffic have been proposed in the previous phase of the project. These models are being validated and simulated in Matlab taking advantage of the object-oriented implementation of the sumolib library for the Matlab environment (1), which allows to parse a sumo network. Note that nowadays one can build a realistic SUMO network easily by importing it from Open Street Maps (OSM) and editing it, or creating it using NETEDIT.

- **Traffic control strategies:** An extensive study on traffic control is being carried out, from the most simplistic approaches, like static signal plans based on demand studies, to optimal control strategies and methodologies.

- **Parallel/Distributed simulation:** Since the interest is on achieving a large-scale urban traffic simulation capable of dealing with centralized and distributed control strategies, it is worth to evaluate the advantages of performing distributed simulations and the trade-offs in performance and accuracy.

In this paper, two important steps for achieving distributed simulations are presented: Strategies for network partitioning and simulation of border edges. This article is organized as follows: Section 7.2 reviews related work done on distributed road traffic simulation. Section 7.3 describes the methodologies and tools used for partitioning SUMO networks and performing distributed simulations, focused on strategies for managing the boundaries of network partitions. Section 7.4 shows simulation results and a discussion. Finally, section 7.5 presents the conclusions.

7.2 Previous related work

7.2.1 Distributed SUMO (dSUMO)

dSUMO (3) allows to run simulations for SUMO in a distributed manner, either in multiple cores of the same machine or in multiples machines. Each of these simulations are called nodes, i.e. each node contains an instance of SUMO. Communication between nodes is performed by means of sockets. Additionally, a simulation scenario is divided into several partitions, each one assigned to a node and the vehicular flow is synchronized between them. This work focuses on methods to partition the network communications between nodes and management of borders. dSUMO works in such a way that whenever a simulation step is done, every node sends messages to its neighbors about vehicles that leave them, once the neighboring nodes receive these messages, they insert the vehicles properly. However, although the authors reported an excellent accuracy of data validating dSUMO with the centralized version of SUMO, the authors clarify that, in terms of the time required for the synchronization of the nodes, its value is an average of 0.32 seconds for partitions on a single machine (each partition running in a processor core), and an average of 0.38 seconds when the partition is made on different machines. Regarding these values, the authors remark that they are larger than expected and they can be improved by optimizing the communication libraries or other schemes of communications such as IPC distributed shared memory type, which further complicate the solution, given that TraCI could not be used in this case. These findings are supported by the fact that when comparing only times of execution, the authors reported an improvement in dSUMO vs. SUMO. Another important aspect is related to the accuracy of the distributed simulation. In this work, the authors demonstrated that it is possible to achieve a 100% accuracy in free flow conditions. On the other hand, they report a 99.38% accuracy in a congestion scenario. It is worth noting that the accuracy metric used is the "percentage of cars that are at the same position at the same time". Moreover, they argue that there is an "unexpected safety check", in congestion scenarios that prevents the insertion of vehicles in the destination partition. We consider that this situation deserves special attention, and it is explained in more detail in section 7.3.

7.2.2 Distributed-Parallel Road Traffic Simulator for Clusters of Multi-core Computers (DUTS)

As in dSUMO, DUTS (7) considers clusters composed of multiple computers, each with multiple cores. Two versions of the simulation are compared: one purely distributed, where the cluster running processes in a single thread and other distributed/parallel, where a smaller number of processes is used but each one with multiple threads. It uses the simulator *Distributed Urban Traffic Simulator* (DUTS), a distributed version of the simulator *Java Urban Traffic Simulator* (JUTS), which has not yet been released. It can be concluded that the scheme of distributed/parallel simulation has better

performance, thanks to its shared memory feature.

The article describes a spatial method for the partition of the network, the communication between nodes is given by means of protocols and synchronization is done via special nodes, called barriers. Nodes correspond to a computer. In this way, the simulation distributes partitions of the network and assigns a subnet to each node, while the parallel/distributed simulation extends the purely distributed schema to use multiple threads programming, related to a core of each computer. It is important to mention two aspects in relation to the parallelization. The first is that the communication is minimized thanks to the shared memory. The second is that parallelization also requires synchronization through a barrier, since a set of network elements is assigned to each of the program threads (the elements that make up the network are lanes, intersections, etc.).

Several simulations were perform for networks of different sizes in computers of quad-core, finding that the use of the four cores in parallel mode represents average a speed of 3.32 times higher than the use of a single-core. This relationship is not linear due to the additional processing required for communications. Moreover, it was found that it can be accomplished in a scheme parallel/distributed improvements in performance up to 52% for the case of two-node and four cores compared to purely distributed schema.

7.3 Distributed Simulation in SUMO

Achieving a distributed road traffic simulation has to consider tasks for partitioning the system, in this case the SUMO network, and tasks related to the distributed simulation itself. These tasks and the involved SUMO tools are described in the following subsections.
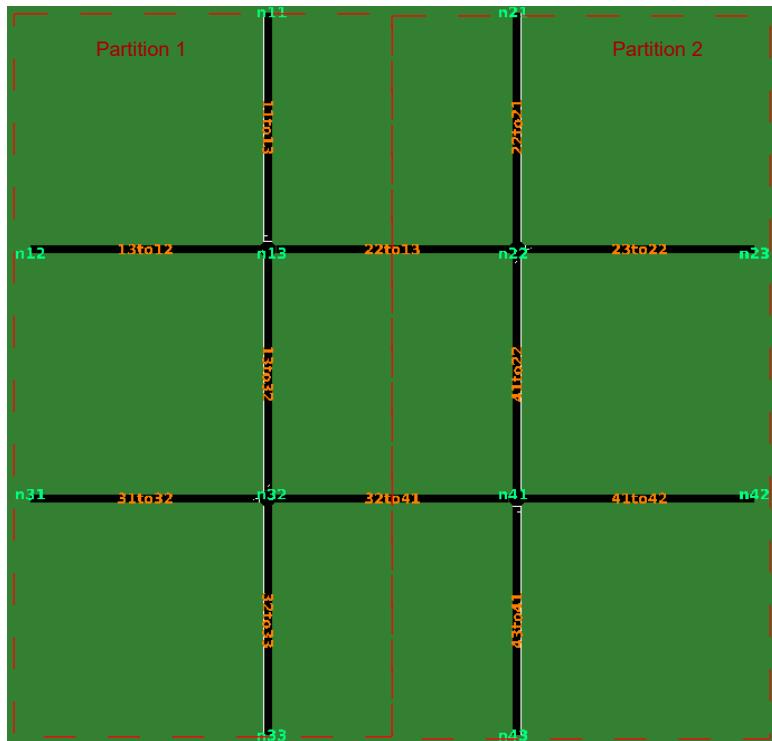

Figure 7.1: Test scenario for distributed simulation in SUMO

7.3.1 Network Partitioning

Road networks can be represented, in general, as a weighted directed graph. A distributed simulation requires this graph to be properly partitioned so that each subsystem is assigned to a computer.

The problem of graph partitioning has been widely studied and it is not a subject of this article. Readers interested in this fascinating topic can refer to (2; 5; 4; 8; 6). Instead, a description on how to partition a SUMO network will be made, to illustrate the method. It is assumed that the inputs for that module are a SUMO network with a given traffic demand and a definition of the network partition. In this case, the definition of the network partition corresponds to the edges that belong to each partition, initially with border edges belonging to the pair of partitions it spans.

To elaborate, consider the network showed in figure 7.1. As explained above, each partition is defined by the edges it contains, including the border edges, which in this case are defined by ids 22to13 and 32to41. Therefore, the two partitions showed in figure 7.1 can be defined as shown in the following code listing[1]:

```
partition1 = {'11to13','13to12','22to13','13to32','31to32','32to41','32to33'
    };
partition2 = {'22to21','22to13','23to22','41to22','32to41','41to42','43to41'
    };
```

Using NETCONVERT along with the option `--keep-edges.explicit`, the network partition defined above can be obtained. However, as noticed in (3), in the network building process NETCONVERT makes junctions' geometries to influence edges' lengths. Hence, whenever a junction changes its geometry because it has been turned into a dead end by means of the partition process, the lengths of the resulting edges are not the same, thus affecting the behavior of vehicles in the distributed simulation. This can be solved by patching the network either with additional edges, like in (3), or by properly modifying the position of the nodes, as showed in the following code listing:

```
net = sumolib.NetReader.readNet(originalNetID);
for i = 1:length(partitions)
    partitionNet = sumolib.NetReader.readNet(partitions(i).getID());
    inBoundEdges = partitions(i).getInBoundEdges();
  for j = 1:length(inBoundEdges)
        patchNodePosDiff(net, partitionNet, inBoundEdges(j));
    end
end
```

where the function `patchNodePosDiff` computes the difference in the positions of the starting nodes of the boundary edges in the original SUMO network `net` and those in the networks corresponding to each partition `partitionNet` and applies these patches accordingly. Note that the sumolib library is used for parsing the original network and the network partitions. Figure 7.2 shows the results obtained partitioning the test scenario, the red Points of Interest (POI) denote the actual boundaries.

Demand Partitioning

As stated previously, it is assumed that one of the inputs to the network partitioning module is the traffic demand associated to the road network. Like in the network partitioning process, vehicle routes have to be partitioned, so that there is one route file per partition. For example, in the scenario shown in figure 7.1, consider a vehicle whose route starts in partition 1 and ends in partition 2. The resulting route from the demand partitioning should end in the boundary edge 32to41 for the routes file of partition 1 and start in the same edge for the routes file of partition 2.

The routes partitioning strategy explained above can be achieved by a suitable extension to the tool `cutRoutes.py`. This script extrapolates the departure times of routes that start outside the specified subnetwork according to the number of edges outside it and their speeds. However, in distributed simulation these departure times are delegated to the algorithm for border edges management, which is in charge of removing vehicles that leave a given partition and inserting them in the destination

---

[1]The code listings showed herein follow a Matlab-like syntax, since the actual implementations relied on the TraCI4Matlab and SUMOLib4Matlab libraries

(a) Partition 1   (b) Partition 2
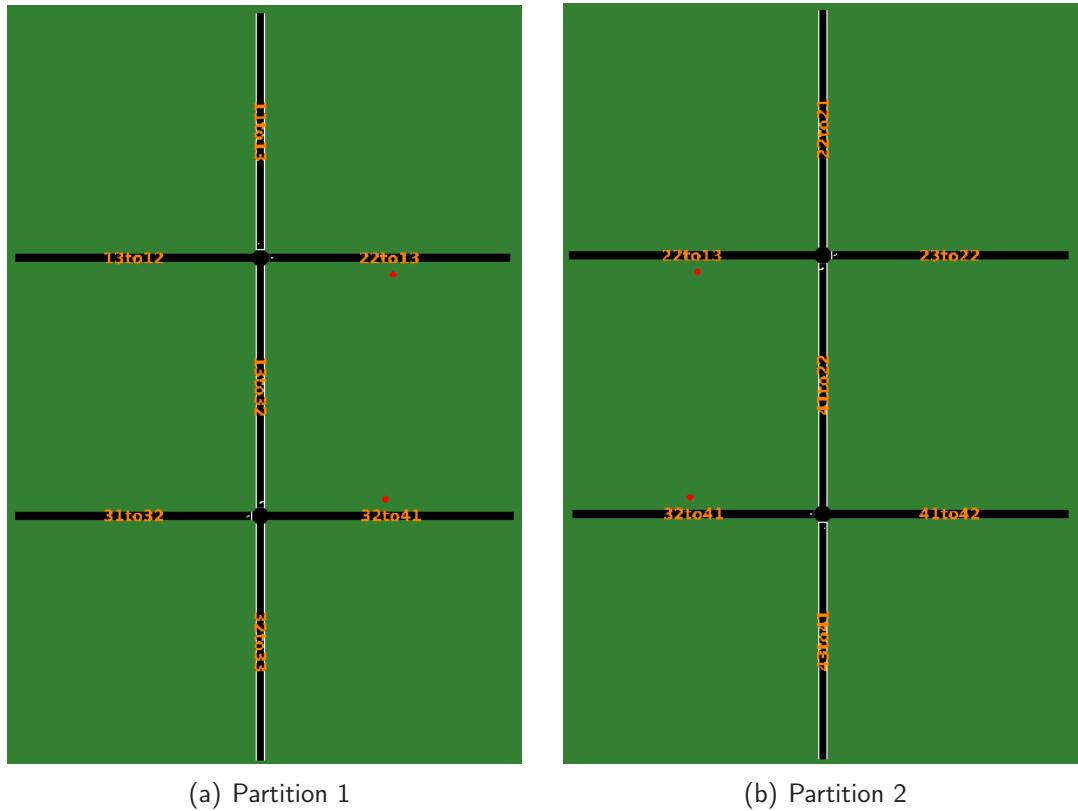
Figure 7.2: Results from the partition of the test scenario

partition. Nonetheless, the extrapolated departure times computed by `cutRoutes.py` can serve for comparative analyses, specially in congestion scenarios.

The extension made to `cutRoutes.py` creates routes for every vehicle entering the subnetwork, so that they can be referenced in the `traci.vehicle.add` command. Additionally, this approach guarantees that the entire route for every single vehicle is preserved after the partition.

Recall the example of a vehicle whose route starts in partition 1 of figure 7.1 and ends in partition 2. The route of this vehicle in the original SUMO routes file would look like the following:

```
<routes>
    <vType id="zeroSigma" vClass="passenger">
        <carFollowing-Krauss sigma="0.00"/>
    </vType>
    <vehicle id="31to421_zeroSigma.0" type="zeroSigma" depart="50400.00">
        <route edges="31to32 32to41 41to42"/>
    </vehicle>
</routes>
```

Here, the `zeroSigma` vehicle type was defined to implement the Kraus car-following model, with $\sigma = 0$, which means that there is no variability in the acceleration of vehicles (3). Using `cutRoutes.py` and the included extension, the obtained routes file for partition 1 would be:

```
<routes>
    <vType id="zeroSigma" vClass="passenger">
        <carFollowing-Krauss sigma="0.00"/>
    </vType>
    <vehicle depart="50400.00" id="31to421_zeroSigma.0" type="zeroSigma">
        <route edges="31to32 32to41"/>
    </vehicle>
</routes>
```

and the routes file for partition 2:

```
<routes>
  <route id="31to421_zeroSigma.0" edges="32to41 41to42"/>
</routes>
```

With this information, a distributed simulation can be performed using TraCI. The vehicle removal and insertion and the exact position where vehicles shall be inserted depend on the strategy for managing the border edges, which will be discussed in the next section.

7.3.2 Border Edges Management

One of the main components for the distributed simulation in SUMO is responsible of transferring vehicles that leave one partition to the next one. The simplest scenario has free flow conditions in border edges. On the other hand, the most complex scenario involves queue formation along partitions, bus stops and lane changes in border edges, to mention just a few. The following subsections describe these scenarios starting from the simplest. Specifically, in scenarios with queue formation, several strategies are described along with their advantages, disadvantages and challenges.

Free Flow Scenario

The free flow scenario described here considers the following assumptions:

- All vehicles travel at the maximum speed allowed in the border edges.

- There are no lane changes in the border edges.

- There are no bus stops in the border edges.

The border edges management module has to verify vehicles that leave a given partition so they can be inserted in the next partition using the `traci.vehicle.add` command. However, vehicles are inserted one time step after, thus causing an undesired error. Due to the simplicity of the free-flow scenario, this problem can be solved by predicting if the closest vehicles to the boundaries are leaving the partition in the next time step, according to the maximum speed of the border edge.

Scenario With Queue Formation

When free flow conditions are not satisfied in the boundary edges, vehicles close to them have to interact with those in the next partition and adapt their speeds accordingly. However, like in the case of free-flow, if the distributed SUMO implementation relies on the TraCI API, one has to take into account that the `traci.vehicle.setSpeed` command takes place in the next time step, when the leader vehicle's speed may have changed. Furthermore, the insertion of vehicles through the `traci.vehicle.add` command is effectively performed in the next time step only if the insertion speed does not cause a crash with the leader vehicle. Otherwise, the vehicle will be held until this condition is met. If no insertion speed is provided to the `traci.vehicle.add` command, vehicles are inserted with zero speed and adapt it to their leader vehicle, which is also undesirable since this vehicle becomes the new leader and the new follower would decelerate unexpectedly. Ideally, the solution would require to predict the movements of all vehicles in the boundary edges in the next time step, but this solution comes at the cost of replicating car following models and other models such as those for interacting with traffic lights and stops.

With these considerations, three strategies for border edges management can be identified:

- **Replicating the car following model.** The car following model is implemented in the distributed simulation module and is applied to the vehicles approaching the boundary.

- **Using the** `traci.vehicle.stop` **command.** Using this command, one has to make sure that the specified stop does not violate the braking distance restriction. Hence, this solution checks if the leader vehicle is stopped and tries to stop the follower at a distance given by the minimum safe gap. If the braking distance exceeds this distance, then the vehicle stops at a distance given by the braking distance and is not removed from the incoming partition in order to avoid the crash with the leader, but introducing an error.

- **Inserting a virtual traffic light.** The propagation of the queue can be done through a traffic light placed in the outgoing boundary edge and controlled by the queue length of the boundary edge of the next partition. Note that, in this case, the partition strategy has to split the boundary edges and insert the required traffic lights.

7.4 Results and Discussion

This section describes results obtained for the border management strategies for free flow and queue formation along partitions explained in subsection 7.3.2 and applied to the test scenario shown in figure 7.1. These results are focused on analyzing individual trajectories, corresponding to the first trajectory found of a vehicle that interacts with a leading vehicle in the destination partition, in each case through a tool that parses the netstate dump output, similar to `vehLanes.py`. This tool constructs the trajectories of the vehicles from the partitions dumped by netstate and compares them against the trajectories obtained from the netstate dump of the centralized SUMO simulation. Errors include the differences in input and output times to the edges and differences in the position on the lanes.

7.4.1 Free Flow scenario

As explained previously, it is relatively simple to achieve a 100% accuracy in a free flow scenario by solely using the TraCI interface. Figure 7.3 shows the trajectory of a test vehicle in the free flow scenario.
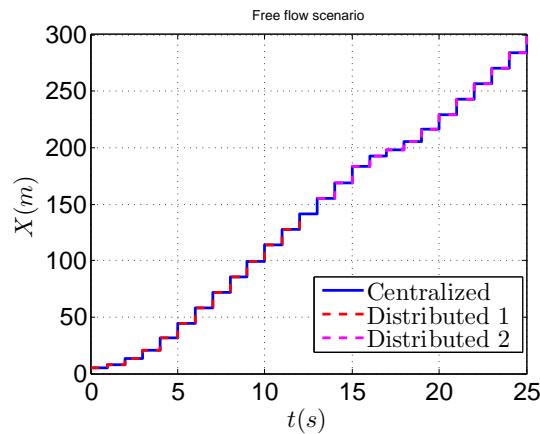
Figure 7.3: Trajectory of a test vehicle in the free flow scenario

7.4.2 Scenario with queue formation

Queue formation along the two partitions of the test scenario, shown in figure 7.1, could be achieved by inserting traffic lights at junctions `n13` and `n41`. In general, the biggest problem regarding the distributed simulation of scenarios with congestion in boundary edges, is the adaptation of

the velocity of vehicles that are leaving a given partition, in relation to leading vehicles found in the next partition, which cannot be performed instantaneously because the simulation is time-discrete. Additionally, it was found that the strategy of predicting the position of vehicles close to the boundaries in the next time step according to their current speeds so that they are inserted in the proper time step in the next partition can cause problems in the accuracy of the simulation. Recall that two forms of using the `traci.vehicle.add` command can be considered: one that specifies the insertion speed and one that does not. Their impacts in the strategy for managing border edges in congestion scenarios are explained again, as follows:

- **Specifying insertion speed:** In this case, there is the risk that the insertion speed causes a collision, then the vehicle is not inserted until this restriction is satisfied. In this case, the speed of the vehicle leaving the partition cannot be the same used in the insertion into the next partition. The speed should be adapted to the behaviour of the leading vehicle therefore the speed of the leader needs to be predicted in the next step making the border edges management strategy more difficult.

- **Without specifying insertion speed:** In this case, vehicles are inserted with zero speed and properly adapted in the next time step. However, zero speed would cause an undesired deceleration of vehicles in the previous partition.

Addressing this issue demands to ignore the insertion speed parameter, and try to insert vehicles in the proper time step and position.

Now, the details and results of the three strategies for managing border edges in congestion scenarios mentioned in section 7.3.2 are discussed. These strategies were analyzed considering the trajectory of a selected test vehicle, making sure that it interacts with the formation of the queue.

(a) Partition 1   (b) Partition 2
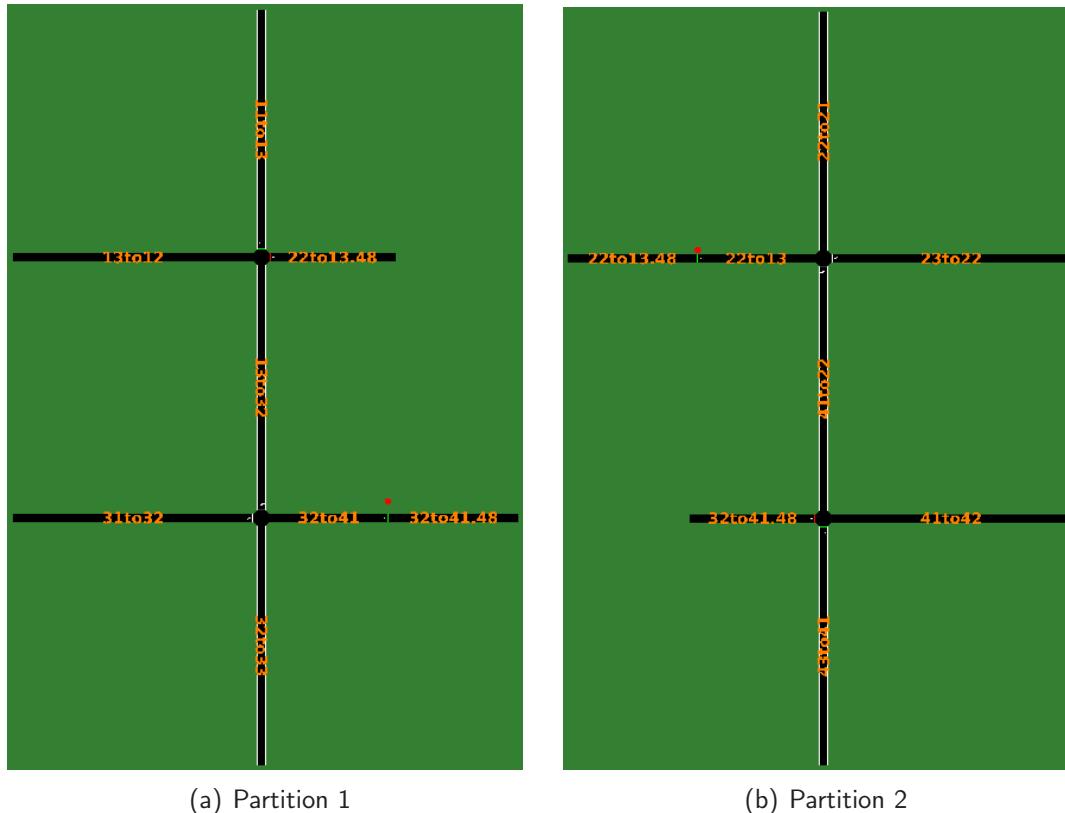
Figure 7.4: Results from the partition of the congestion scenario using virtual traffic lights strategy

Figure 7.5 shows the trajectories of the test vehicle using the aforementioned strategies. Surprisingly, for this test vehicle, the best results were not obtained with the car-following replication strategy,

but with the virtual traffic lights strategy. Furthermore, the first simulation with the car following replication strategy resulted in a larger error, because of the length of the lanes, the computed following speeds never resulted in zero. Therefore, the strategy was adapted to detect if the leader was stopped. In this case the follower is not removed from the partition until the leader's speed is different from zero. The `setStop()` command strategy works similarly: once it detects that a leader vehicle in the next partition is stopped, the `setStop()` command is executed on the follower and it is not removed from its partition until the leader starts to move. In the virtual traffic lights case, recall that the network partition has to split the border edges and insert the required traffic lights. Since one cannot define traffic lights in dead-ends, both outgoing edges resulting from the split have to exist in the partition, as shown in figure 7.4. These traffic lights are controlled by the queue lengths of the border edges in the next partition. If these queue lengths equal the capacity of the border edge, the virtual traffic light changes its state to red. For changing the state back to green, it is not enough that the queue length of the border edge is smaller than its capacity, but also that the leader vehicle starts to move.

(a) Using the car following replication strategy

(b) Using the setStop() command strategy

(c) Using the virtual traffic lights strategy
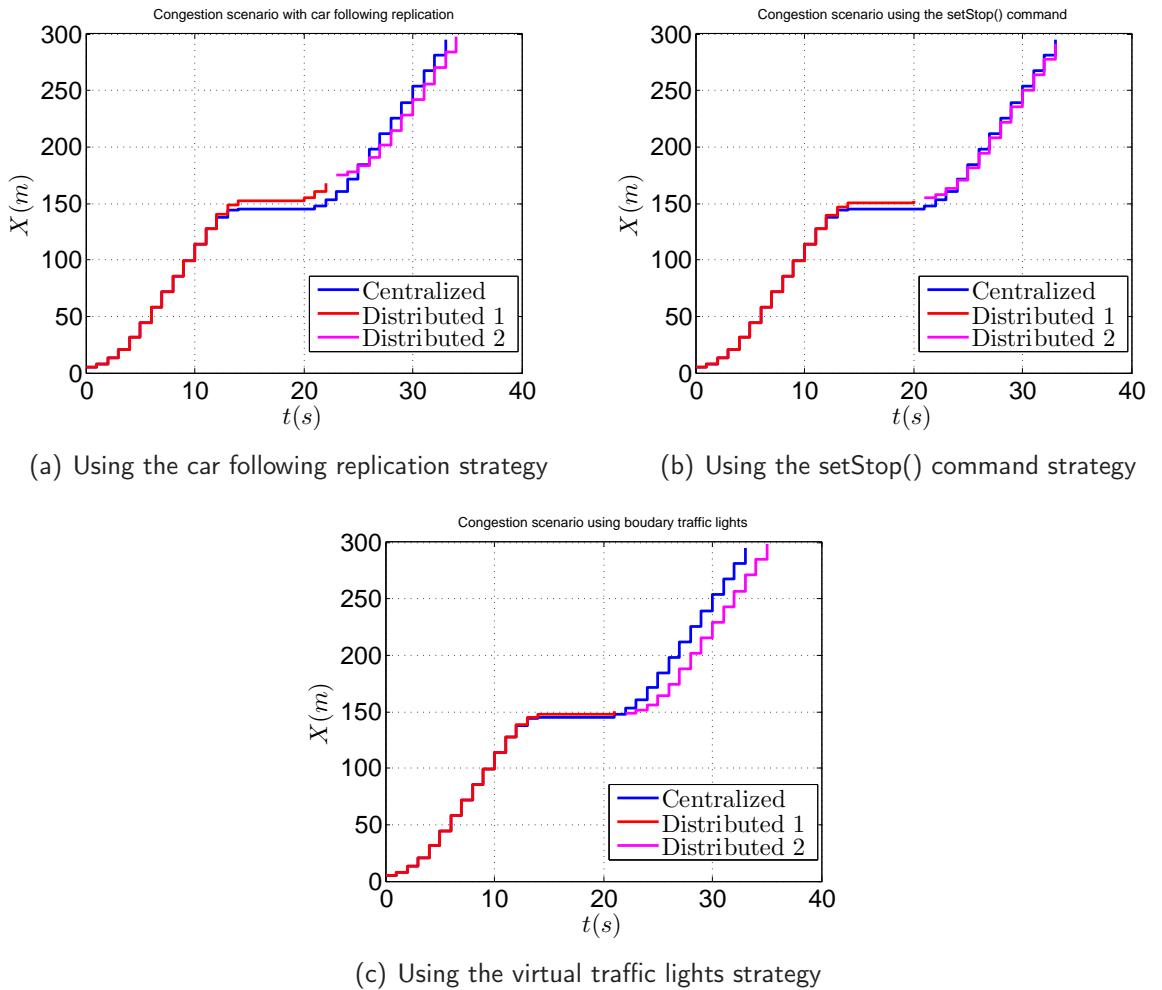
Figure 7.5: Obtained trajectories for the test vehicle

Table 7.1 summarizes the errors for the different border management strategies. Input and output times to edges were averaged over the entire number of edges comprising the route of the vehicle. The simulations were performed with a sample time of one second. From figure 7.5, it can be seen that the minimum error in the position where test vehicle stops is obtained with the virtual traffic lights strategy. However, it was found, that in some cases, since a yellow state for the virtual traffic light was not used, vehicles are forced to stop exceeding their maximum allowed deceleration. For

future improvements, this yellow state could be designed according to the queue length in the next partition.

Table 7.1: Mean errors in trajectory, input and output times to edges obtained for the test vehicle using three strategies for border edges management

Strategy	Mean trajectory error (m)	Mean input time error (s)	Mean output time error (s)
Car following replication	3.18	1.33	2.33
setStop() command	2.49	0.33	0.67
Virtual traffic lights	1.56	0.67	1

7.5 Conclusions

This article described strategies for network partitioning and border edges management aimed for achieving distributed simulations using SUMO. Starting from a given description of the different partitions comprising a simulation scenario, including the traffic demand definition, a detailed explanation of how to perform the partition on the network and the demand using different software tools found in the SUMO package was presented. Furthermore, three strategies for managing border edges in congestion scenarios with queue formation were introduced. Results show that it is possible to achieve simulations with no error in free flow scenarios. On the other hand, border edges in congestion scenarios are more difficult to simulate because the interaction with vehicles in the next partition is affected by the discrete-time nature of the simulation.

7.6 Acknowledgements

7.7 References

[1] Acosta, Andrés, Jorge E. Espinosa and Jairo Espinosa: *Developing Tools for Building Simulation Scenarios for SUMO Based on the SCRUM Methodology*. In *Proceedings of the 3rd SUMO User Conference SUMO2015*, pages 23–35, Deutsches Zentrum für Luft - und Raumfahrt e.V., May 2015. Deutsches Zentrum für Luft - und Raumfahrt e.V.

[2] Arora, Sanjeev, Satish Rao and Umesh Vazirani: *Expander flows, geometric embeddings and graph partitioning*. Journal of the ACM (JACM), 56(2):5, 2009.

[3] Bragard, Quentin, Anthony Ventresque and Liam Murphy: *dSUMO: towards a distributed SUMO*. 2013.

[4] Ding, Chris HQ, Xiaofeng He, Hongyuan Zha, Ming Gu and Horst D. Simon: *A min-max cut algorithm for graph partitioning and data clustering*. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 107–114. IEEE, 2001.

[5] FLAKE, GARY WILLIAM, ROBERT E. TARJAN and KOSTAS TSIOUTSIOULIKLIS: *Graph clustering and minimum cut trees*. Internet Mathematics, 1(4):385–408, 2004.

[6] POTHEN, ALEX: *Graph partitioning algorithms with applications to scientific computing*. In *Parallel Numerical Algorithms*, pages 323–368. Springer, 1997.

[7] POTUZAK, TOMAS: *Distributed-Parallel Road Traffic Simulator for Clusters of Multi-core Computers*. In *Proceedings of the 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications*, DS-RT '12, pages 195–201, Washington, DC, USA, 2012. IEEE Computer Society.

[8] SCHLOEGEL, KIRK, GEORGE KARYPIS and VIPIN KUMAR: *Graph partitioning for high performance scientific simulations*. Citeseer, 2000.

8 The design and development of an HLA-based open platform to model urban environments in SUMO

Christos Tsotskas, Didac Busquets, Robin North, Raymond Francis, Patrizia Franco, Fabio Galatioto, Carl Goves, Roland Guichard, Eifion Jenkins, Ecaterina McCormick, Abraham Narh, Graham Ravenscroft, Richard Tusting, Vittoria Parisi, Nicolas Le Glatin, Graham Fletcher;
Transport Systems Catapult, UK, Transport Systems Catapult, 3rd Floor The Pinnacle, 170 Midsummer Blvd, Milton Keynes, MK9 1BP, +44 (0)1908 359999 ext.
{firstname.lastname}@ts.catapult.org.uk

Abstract

Several software packages and methods exist that simulate aspects of an urban environment in a monolithic way, but frequently it is challenging to scale up and link to applications without changing the internals of participating components. We present the structural design of an in-house open platform, so as to address strategic and operational requirements of urban traffic modelling and illustrate its use with SUMO. This builds on High Level Architecture (HLA; IEEE 1516e) methods developed for military applications. The main goal is to provide the means for academia and industry to integrate related systems in a transparent and time-synchronous way in accordance with well-established industrial standards. By extending Simulation of Urban MObility (SUMO), it was possible to achieve higher levels of integration and interoperability of existing and new urban models. This will enable traffic modellers to simulate more complicated real-world applications and simulate interactions that would not be achievable otherwise, while relying more on open-source software packages. Ultimately, it is expected to assist decision makers to make more informed decisions and influence the future of urban traffic modelling in a continuous shared space.
Keywords: SUMO, HLA, transport modelling

8.1 Introduction

In most countries, especially those with well-developed or expanding economies, traffic congestion is widely recognised as a major problem given that it makes a significant contribution to global warming, environmental pollution and the depletion of fossil fuels (6; 20; 21). Traffic congestion causes substantial delays, especially in urban areas, which adversely affects journey times and the convenience of road travel in general. Congestion has a significant impact on local air quality with road transport being responsible for over 90% of domestic transport emissions, and the declaration of air quality management areas, in UK urban areas (27; 15). The estimated cost of congestion is £13-£20 billion per year to the UK economy (13; 15).

In spite of these adverse impacts there has, over last decades, been a rise in car ownership and vehicle mileage. Building new roads to increase urban road capacity on a large enough scale is nowadays unlikely as this often tends to be unpopular, economically unaffordable and environmentally unsustainable (21). Along with several initiatives to curb the rising growth in mileage and ownership

level, it is also envisaged that the automated systems will play a significant role in addressing traffic problems (4).

The up-to-date progress in research makes the potential uptake of automated vehicles imminent. However, modelling of automated vehicles on the road networks and shared spaces, including their interactions with humans and traditional drivers, is a challenging task. Consequently, the use of existing standalone tools may result extremely time consuming, not efficient and sometime not fitting for the purpose at all.

Moreover, the current era of pervasive computing and advancement of computing tools presents a unique opportunity to understand the interactions of the various entities (vehicles, pedestrians, cycles, infrastructure etc.). This is possible through seamless and interoperable means of modelling these behaviours in order to enhance the practice of traffic management (17). This work suggests answers to the following research questions:

- How to structure the core of a systems architecture in a sustainable way that could meet the current and future requirements of urban transport modelling?

- How to model the shared space between pedestrians, vehicles and additional types of actors?

- How to simulate entities and their interactions to other entities in an urban environment at scale?

- How to develop a system that can be used by both academia and industry so as to provide further insight to decision makers in real-world problems?

The scope of this work is to present an architecture that can enable the integration of software packages (originally not linked to each other), reuse available models in order to implement more complicated applications to tackle real-world problems, reduce barriers to analysis and to accelerate research and development. The key features behind the suggested architecture are flexibility, integrating both open-source and commercial software packages (or, even, a combination of those) to model complex real-world problems, simplicity, and the ability to reconfigure its structure in an easy way. Because these packages implement the same business logic their contribution to the architecture should be transparent, but minor discrepancies are expected. Several software packages and methods that simulate aspects of an urban environment operate in a monolithic way and coupling them together is not a trivial task. By plugging them into the suggested architecture, further integration would be feasible, which would enable more stakeholders to conduct further research and development for academic or industrial applications. Although open-source packages have been used to create a demonstrator based on this architecture, all of the components of the architecture can be exchanged with other tools of the same class. This gives the flexibility to the user to use the current compilation of tools, or to create a more tailored version, as required. In this paper, an integration of pedestrian and vehicle models will be presented to demonstrate the aforementioned principles.

The global aim of this research is to bring together a wide range of heterogeneous tools and methods both from academic and industrial fields. This heterogeneity covers a wide range from geographically distributed resources to multiple platforms. Furthermore it has the potential to enable interactivity (human or external system interaction). In the case where there is no human in the loop it is possible to accelerate simulation, as also mentioned in (26). Ideally, the flexibility of open-source software packages and the performance of commercial software package could be joined together. This would enable multiple stakeholders to run complicated simulations that model the interaction of entities in an urban environment at scale. The range of applications can vary from modelling a simple linear corridor up to modelling an entire city with traffic signals, public transport options and weather impact to name but a few.

8.2 High Level Architecture at a glance

The High Level Architecture (HLA) (1), primarily a communication protocol, was selected to be at the core of the architecture so as to achieve the objective of establishing transparent communication among the heterogeneous components and principles. HLA is a standard that enables distributed simulations and favours interoperability and reusability of components. The most important feature is that it enables synchronous communication between heterogeneous components. A number of rules can define business principles and pervade the whole architecture (16). By definition, distributed simulations are supported by declaring a set of run-time services and participating components communicate to each other by publishing and subscribing data of interest (1).

More specifically, HLA comprises a federation and one or more federates. A federation could be considered as the whole system, whereas federates can be seen as the independent modules of the system. A participating federate communicates with the system only through the Runtime Infrastructure (RTI), as shown in Fig. 8.1. This expands to more Federates, as depicted in Fig. 8.2. More precisely, each federate uses a special module, called an Ambassador, so as to communicate with the RTI. Similarly, the RTI has also an Ambassador to speak to the Federate's Ambassador. As such, the communication is achieved via the Ambassadors, as illustrated in Fig. 8.3.
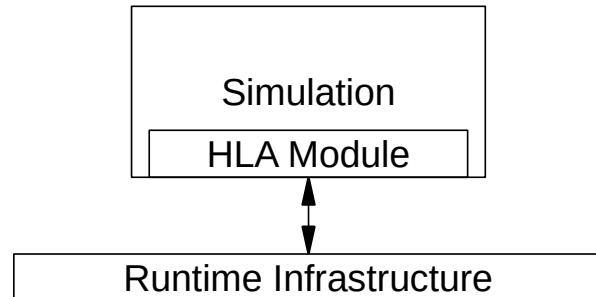

Figure 8.1: Communication in the federation
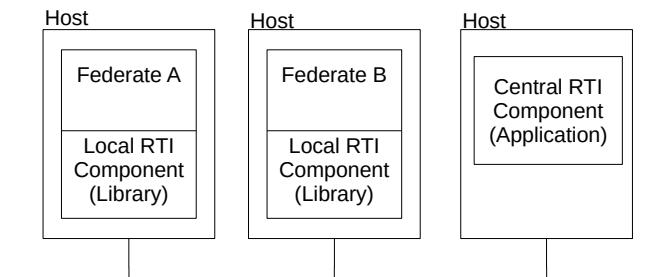

Figure 8.2: A network of federates
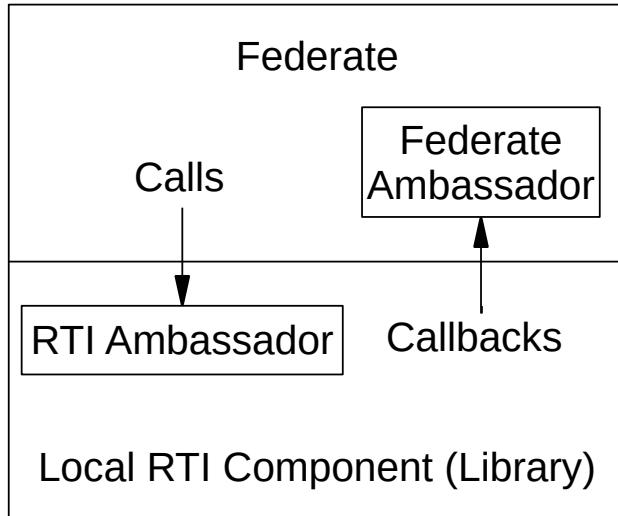
Figure 8.3: Communication through ambassadors

HLA provides a "Publish / Subscribe" data exchange mechanism between model components using a shared networked RTI. By moving the communications of a distributed simulation intelligence to a lower level, external to the component that actually carries out the simulation, the HLA federation is more versatile to changes, as the communications take place in the background. The simulation components may publish all the data that may be required in any application, incurring a low communication and computational overhead as a result. Only when another simulation component subscribes to it, the data are transferred which allows each component to be prolific in the data it publishes. This scheme facilitates data exchange while each individual simulation runs. Moreover, this keeps each component independent of each other, since they only have to be able to send and receive information to the RTI, regardless of what information other components will receive or send. A pairwise (or multi-party) communication can be established when at least one federate publishes information through the RTI and at least another subscribes to the same information (a federate can equally subscribe and publish). In a single-federate federation, a federate can publish (or subscribe) without any federate subscribing (or publishing), until more federates join at a later stage. In practice, each federate contains a small and private version of the whole world, where a set of mechanisms sense and react to entities in that area.

The structure of the data to be exchanged are described in the Object Model Template (OMT), which is described in great detail in (16). Among other files, the most important one is called the Federate Object Model (FOM), which describes the world in terms of HLA. FOM can be modular and can expand the definition of the world.

8.2.1 Related work based on HLA

HLA has been successfully used in a number of real-world applications. By using HLA in the background along with explosive detonation simulations and wind field computations, the generation of pollutants' concentration was calculated and combined with traffic flow simulation to simulate scenario of urban chemical disaster (26; 8). This could be used in cases of mission rehearsal, systems acquisition and analysis of alternative.

The execution of an assembly of simulations was accelerated by using HLA, as described in (24). For the purposes of research projects, off-the-shelf simulators were used to quickly prototype new applications in the fields of automotive and aeronautics. A cognitive architecture was suggested to speed up the execution of software packages by using open-source implementations of RTI and to

enable the researchers to swap different versions of RTI. This was used in two projects; first in an intelligent simulation for driver behaviour and, second, in simulating distributed agent systems of manned and unmanned aircraft (with commercial stakeholders).

In order to enhance military simulations in terms of accurately representing pedestrians at a city-scale level, a new tool was introduced in (18) to model vehicle and pedestrian activity. By using HLA, geographical information systems are linked to the system to accurately represent the terrain. Models related to traffic simulation and behaviour representation were also presented. This compilation was successfully used to carry out three different military training operations. However, architectural details in terms of systems modelling and data models are not revealed, and that framework depends on commercial implementations of HLA, which could potentially impede research applications.

By using SUMO (9) and an in-house computational framework under HLA, the quality of air was studied in (19). Electric buses were used to simulate and minimise the environmental footprint, which was validated against experimental data. In the same work (19), it is stressed that new flexible technological approaches are required by research and development stakeholders to evaluate and simulate more complex cases that combine automotive and transportation research.

So far, all the cited work attempts to include specific parts of an urban environment. However, none of them includes pedestrian activity and easy access to the developed tools. To the best of the authors' knowledge, it is the first time that the pedestrians, vehicles and their interactions are going to be simulated together in a shared space environment under a publicly available framework, as described in the remainder of the paper. This application could ultimately be used to simulate the impacts of transport proposals and thus assist decision makers in shaping Future Urban Transport Systems.

8.3 Platform architecture

At the top level of the data model, the architecture is very abstract and becomes more specific while moving down. The top level of the architecture is shown in Fig. 8.4. The "bridge" design pattern was selected to be applied because it separates the implementation details from the abstraction. At the top level lives a class that represents the software packages, which simulate traffic or movement, also called microsimulators, and a class that controls the microsimulator.

This architecture is very extensible, because it is based on the HLA. If the user needs to add a new vehicle microsimulator, just extending the respective class and providing the corresponding connector would be enough. The first class will be a simple wrapper around the software package and the connector would be a wrapper to the Application Program Interface (API) that controls that software package. By specifying HLA's FOM and microsimulators' configuration settings is sufficient to run the application. The merit of this structure would be more obvious as more principles would be integrated, such as environmental and financial models.

Following the design pattern of "single-responsibility principle", the described architecture aims to support microsimulators and communication protocols, while being in the background. Hence, additional effort was put in designing the architecture in order to couple HLA with other microsimulators in the most transparent way, so as to enable other developers and future extensions to use the system as much conveniently as possible. This architectural decision also favours backwards compatibility with several other packages, too. Integrating all the other modules (described below) took considerably less effort. All the user needs to do is to expand the aforementioned classes and instantiate the microsimulator wrapper, as required. For instance, in an open-source environment that uses SUMO to simulate road vehicles, the user just needs to use a compatible version of SUMO from the online repository and specify the location of the preferred executable in the environmental variables. Further specific implementation details for the use case in 8.4 are provided in the corresponding section.
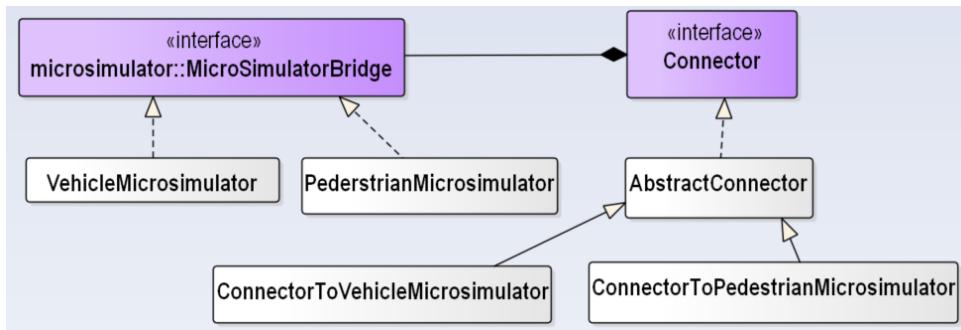
Figure 8.4: The "bridge" design pattern of the architecture

The communication among the participating components is achieved by using HLA. Here, HLA is particularly useful because it contributes to the concept of the shared space. More specifically, a simulated vehicle can sense (via the publish/subscribe mechanism of HLA) the existence of a pedestrian and react accordingly.

All the participating modules exchange files in extensive Mark-up language (XML). These type of structured files were chosen because it was reasonable to support other legacy systems, since most of the software packages are still using XML files. In addition, a lot of the modern frameworks, such as the Spring framework (22), depend on XML. The information exchanged via XML files depends on the level in the stack; for instance, in Listing 8.1, federation-specific information is prescribed, whereas in the communication among microsimulators the most commonly exchanged information includes the location, direction and speed of the moving entities.

The system is developed in Java. This was selected because of the inherent cross-platform compatibility. In addition, HLA's RTI was already based in Portico(23) and SUMO's connector is TraCl4J(3), both implemented in Java.

8.3.1 Structure of the Federation

The HLA federation was designed to be flexible and easy to extend. It is created as an assembly of federates, where each of them is constructed on-the-fly from an external list (an XML file) that describes the federation. Each federate in the federation is further described by their own FOM. An example configuration of the federation is provided in Listing 8.1, where details about the size of the world and the participating federates are presented. This structure is based on the Abstract Factory design pattern, where desired classes (here federates) are created based on a list of specifications. Here, this is based on the Spring framework (22), as an industry practice. This configuration can be generated either by another system or a human (ideally, by using an appropriate graphical user interface).

Listing 8.1: Configuration of the federation with a single federate

```
1  <subSystemConfiguration>
2      <name>PedestrianSimulation</name>
3      <porticoFederateAmbassador>
4          <name>Test HLA Ambassador</name>
5          <federationName>TestFederation</federationName>
6          <federateName>TestFederate</federateName>
7          <federateType>TestFederateType</federateType>
8          <fomFiles>resources/foms/TestSimplePedestrianFom.xml</fomFiles>
9          <masterController>true</masterController>
10         <publisherId>HLAAmbassador</publisherId>
11         <autoTick>true</autoTick>
12         <tickRateMillis>1000</tickRateMillis>
13         <expectedFederateCount>1</expectedFederateCount>
```

```
14    </porticoFederateAmbassador>
15    <simplePedestrian>
16        <name>PedestrianSimulator3</name>
17    <publisherId>PedestrianSimulator3</publisherId>
18        <tickLength>100</tickLength>
19        <tickUnit>MILLISECONDS</tickUnit>
20        <tileOriginX>200</tileOriginX>
21        <tileOriginY>200</tileOriginY>
22        <tileShape>
23            <point>
24                <x>0</x>
25                <y>0</y>
26            </point>
27            <point>
28                <x>0</x>
29                <y>1001</y>
30            </point>
31            <point>
32                <x>1001</x>
33                <y>1001</y>
34            </point>
35            <point>
36                <x>1001</x>
37                <y>0</y>
38            </point>
39            <border>2</border>
40        </tileShape>
41    </simplePedestrian>
42 </subSystemConfiguration>
```

8.3.2 Structure of Federates

 HLA is primarily used to abstract the data transfer mechanism among independent process spaces. So federates could be running on the same computer or separated by a network, but HLA hides this from a federate developer, meaning there is only one implementation for whatever communication is required. However, the "publish-subscribe" concept of HLA can also be used to enable loosely coupled and modularised federates, similar to the observer design pattern.

By breaking federates down into small components, code can be reused across many federates and the functionality can be configured by adding and removing component as required. The "publish-subscribe" mechanism is implemented as a broker to which the different components publish all their data of interest, and components will receive the data only if they have subscribed (also via the broker) to the particular data type. In the broker, data is transferred between components within the process space, and therefore does not use HLA. The HLA ambassador simply becomes a component that subscribes to the relevant parts of the federate's data model through the broker and publishes all the data it receives from any component within the federate to the federation, according to the HLA standard.

8.3.3 Modelling vehicles and pedestrians

As vehicles and pedestrians are going to be simulated in a shared-space environment, their data model shares many attributes, as shown in Fig. 8.5, and it is possible to include a shared origin and destination. More importantly, the position of each vehicle and pedestrian is communicated to the others via RTI. Two separate reference types were created, a vehicle and a pedestrian, to act as the intermediate step when transferring information from one microsimulator to the other, because in

each case, the entities are modelled differently. When a conversion is required, it is handled by the core of the architecture so as to generate entities in accordance with the data model of the target microsimulator.

The data models are not identical. On one hand, vehicles are modelled to operate in the road network. Because vehicles and road segments are expressed differently on each microsimulator, a reference coordinate system is used as an intermediate communication point, for communication purposes. On the other hand, pedestrians are also modelled as 2D entities that can move freely within a designated area.
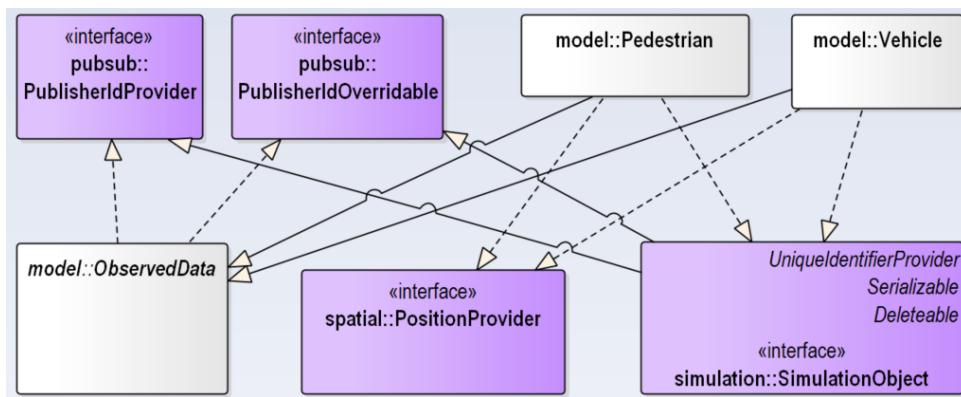
Figure 8.5: Vehicle and pedestrian class diagram

8.3.4 Modelling Locality

The space of interest is separated in shaped tiles that denote the area controlled by a microsimulator. In that area, all the entities of a certain type are simulated by a particular microsimulator. Additional care is required at the borders, where the adjacent tile expects a new entity to arrive. The specification of the world is defined in an XML format, as shown in the "tileShape" element in Listing 8.1.

The world could be simulated in two different ways. First, depending on the workload of a microsimulation, many tiles can cover the same area of the world, but each microsimulation is going to control different number (and type) of entities. Second, the world is going to breakdown into a number of small tiles that partly overlap, where all the entities of the same type will be controlled by a single microsimulator. Obviously, special policies are exercised at the borders, so as to hand-off entities from one microsimulator to the other. The performance of these two approaches is going to be investigated in the future (5).

8.4 Example: towards modelling shared spaces

Public space design has moved towards space sharing similar to "Woonerf" schemes in western countries over the last century (25). The objective has been to raise pedestrian priority and establish dynamics and interactions with pedestrian and cyclist within the street to modify drivers' behaviour (14). These redesigning streetscapes can contribute to an increase in the community texture and an improvement in social interactions (11). Modelling interactions with pedestrian and cyclist of standard vehicle using traffic simulators has been implemented, however several limitations are present when non standards vehicles (eg. autonomous vehicles) may be modelled. Existing tools are not appropriate for such scenarios: most vehicle simulators use lane-based approaches, thus restricting the movement of vehicles and pedestrians, and pedestrian simulators usually cannot model non lane-based vehicles. So far, existing methods (28; 10; 7) to uncover the laws which
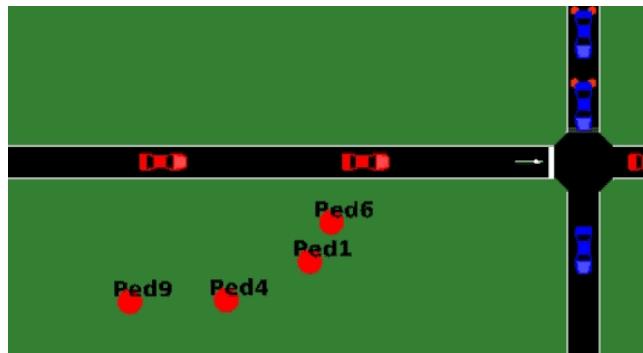
Figure 8.6: Free moving pedestrians within a SUMO network

govern vehicle and pedestrian traffic dynamics separately have been proposed and more recently (2) discusses behavioural patterns of pedestrians and cars in shared space areas using the Social Force Model (SFM) and argues that short range repulsive forces in the SFM leads to excessively frequent urgent detours to avoid collisions.

Although most traffic micro-simulators, including SUMO, are able to represent pedestrians, these are usually restricted to move in specific areas, sometimes also being limited to lane-based movement. These limitations prevent modelling shared spaces, where pedestrians can move freely around all the environment. As a step towards simulating this kind of environments, we have used our HLA architecture to combine SUMO with a pedestrian simulator that does not restrict pedestrian movement in any way. Thus, SUMO is aware of pedestrians not being simulated by itself, but by an external component. We have also modified SUMO to be able to react to these external pedestrians. In the next subsections we present how we represent free moving pedestrians within SUMO and how the vehicles react to them.

8.4.1 Representing free-moving pedestrians in SUMO

Although SUMO can represent pedestrians, these are restricted to move in specified areas, such as sidewalks, crossings and walking areas. In all of them, the movement of pedestrians is similar to those of vehicles in the sense that they use lanes, and so their movement is somehow restricted. Moreover, vehicles only react to pedestrians in crosswalks. These features are therefore not appropriate to model shared spaces where pedestrians and vehicles can walk freely around the environment, and both entities react to each other to avoid crashing.

To overcome these limitations, we have taken advantage of the representation of Points of Interest (POI) within SUMO. These are used to represent static non-interactive objects only for visualisation purposes, consisting of a 2D shape and some color. However, POIs can be dynamically moved using TraCI, and this feature allows us to represent a pedestrian as a (small) POI whose position can be updated at simulation time.

Thus, when the SUMO component (through its wrapper) receives information about pedestrians coming from other components via HLA, it creates a new POI (if it is the first time that pedestrian is received), or updates an existing one (if it had already been created). This results in the POIs moving according to the locations being given by the pedestrian simulator (see Figure 8.6).

The next step is then to make vehicles be aware of these pedestrians so that they react to them. Next section details how we have slightly modified the driving behaviour within SUMO to achieve this.

8.4.2 Pedestrian aware vehicles

In order to include pedestrian-aware behaviour to SUMO vehicles, we have used the Gipps car-following model (12). This model is one of the most basic car-following models, and it involves vehicles driving in a single lane road with the ability to estimate the speed of the preceding vehicle (if any). The model updates the speed of vehicle n by computing two speeds: one for the free driving situation, $u_n^a(t+\tau)$, computed as if there were no cars in front of the vehicle (Eq. 8.1), and another one for the car following situation, $u_n^b(t+\tau)$, taking into account the closest vehicle ahead (Eq. 8.2). The final speed adopted by the vehicle, $u_n(t+\tau)$, is the minimum of these two (Eq. 8.3):

$$u_n^a(t+\tau) = u_n(t) + 2.5 \cdot a_n \cdot \tau \cdot \left(1 - \frac{u_n(t)}{U_n}\right) \cdot \sqrt{0.025 + \frac{u_n(t)}{U_n}} \tag{8.1}$$

$$u_n^b(t+\tau) = b_n \cdot \tau + \sqrt{b_n^2 \cdot \tau^2 - b_n \cdot \left[2 \cdot (x_{n-1}(t) - s_{n-1} - x_n(t)) - u_n(t) \cdot \tau - \frac{u_{n-1}(t)^2}{\hat{b}}\right]} \tag{8.2}$$

$$u_n(t+\tau) = \min(u_n^a(t+\tau), u_n^b(t+\tau)) \tag{8.3}$$

where
a_n	is the maximum acceleration for vehicle n
b_n	is the maximum deceleration for vehicle n
$\hat{b}$	is the estimation of the maximum deceleration for vehicle $n-1$
s_{n-1}	is the size of vehicle $n-1$, including a safety distance
U_n	is the desired speed of vehicle n
$x_n(t)$	is the position of the front of vehicle n at time t
$u_n(t)$	is the speed of vehicle n at time t
τ	is the reaction time of vehicles

We have applied the Gipps model assuming that pedestrians are static (i.e. $u_{n-1}(t) = 0$). Although they might be actually moving, we assume that the slow speed of pedestrian makes no difference, while it simplifies computation. Thus, Eq. 8.2 can be rewritten as:

$$u_n^b(t+\tau) = b_n \cdot \tau + \sqrt{b_n^2 \cdot \tau^2 - b_n \cdot [2 \cdot (x_{n-1}(t) - s_{n-1} - x_n(t)) - u_n(t) \cdot \tau]} \tag{8.4}$$

We have included the computation of $u_n(t+\tau)$ when computing the speed at which each vehicle should drive at each time step. To do so, we first check which is the closest POI (i.e. pedestrian) to each vehicle (in the direction of travel, and within a given distance and field of view), and then apply the Gipps model to avoid running over that pedestrian (see Figure 8.7). This results in vehicles slowing down, and stopping if necessary, in the same way they would do if they had another vehicle or stop line in front of them (see Figure 8.8).

8.4.3 Further work

The presented example makes a step towards the simulation of shared spaces. Although vehicles still move using a lane-based behaviour, we have taken advantage of the flexibility offered by points of interest and used them to represent free moving pedestrians.

There is still much work to do to have vehicles moving freely in a shared space, probably with huge implications in the way micro-simulators are implemented. However, this work has addressed half of the problem, namely to allow pedestrians to move freely within a micro-simulator, and so we believe it is a step towards the final goal of fully simulating shared spaces.
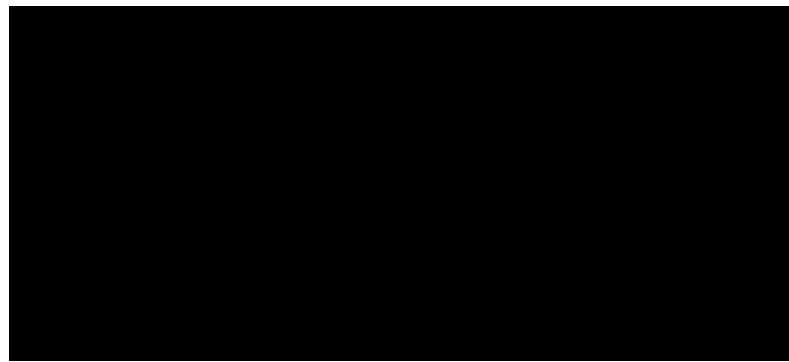
Figure 8.7: Computing Gipps model for the closest pedestrian (represented as the black dot)

Figure 8.8: Vehicles stopping to let pedestrians cross

8.5 Conclusions and Future Work

The architectural design and further implementation details of an open-source framework, developed by Transport Systems Catapult, that simulates pedestrians and vehicles at scale were presented. The business logic and data models were discussed along with architectural details, which are based on design patterns for a sustainable development that can scale up and out. The framework is based on open-source implementations of HLA, traffic simulators and pedestrian simulators. The flexible and open architecture of the framework would enable stakeholders from different fields to carry out both academic, research and industrial applications, and allow them to further extend it so as to include more principles.

Future extensions could add a number of principles to the framework. First, by decoupling the business operations from the actual processing, simulations could run even faster should more computational resources become available and make it increasingly fault-tolerant to hardware or software issues. Second, by integrating virtual reality features, it could be possible to carry out applications of immersed virtual reality, where humans could participate in the loop of simulations, for a more interactive experience and applications. Third, more HLA federates could be added, which will enable the federation to provide extra features of traffic control, selecting a mix of public transport options to carry out end-to-end journey planning and considering weather phenomena that are going to affect the business operations. Consequently, all this is expected to introduce a communication overload on the HLA RTI because of the number of messages exchanged by numerous entities, additional features of HLA (i.e. Data Distribution Management (DDM)) for selective communication and filtering will be developed to enable sustainable scaling up. To a greater extent, this framework could be linked to a larger framework with strategic modelling capability so as to enable more sophisticated cases of transport planning and modelling at a greater scale.

Acronyms

8.6 References

[1] *IEEE Standard for Modeling and Simulation High Level Architecture (HLA)– Federate Interface Specification*. IEEE Std 1516.1-2010 (Revision of IEEE Std 1516.1-2000), Aug 2010.

[2] ANVARI, BANI, W. DAAMEN, V. KNOOP, S. HOOGENDOORN and MICHAEL G.H. BELL: *Shared Space Simulation Based on Social Forces and Distance Potential Field*. In *International Conference on Pedestrian and Evacuation Dynamics*, 2012.

[3] APPEAL: *TraCI4J*. https://github.com/egueli/TraCI4J, 2014. [Online; accessed November-2014].

[4] BISHOP, R.: *Intelligent vehicle applications worldwide*. Intelligent Systems and their Applications, IEEE, 15(1):78–81, 2000.

[5] BUSQUETS, D., R. FRANCIS, C. TSOTSKAS, R. NORTH, F. GALATIOTO, P. FRANCO, R. GUICHARD, R. TUSTING, E. MCCORMICK, G. RAVENSCROFT and G. FLETCHER: *Distributed large-scale simulations of ITS application scenarios*. In *ITS European Congress*, Glasgow, 2016.

[6] CHENG, T. and E. MANLEY: *Understanding road congestion as an emergent property of traffic networks*. 2010.

[7] CHRAIBI, MOHCINE and ARMIN SEYFRIED: *Pedestrian dynamics with event-driven simulation*. In KLINGSCH, W. W. F., C. ROGSCH, A. SCHADSCHNEIDER and M. SCHRECKENBERG (editors): *Pedestrian and Evacuation Dynamics 2008*, pages 713–718, Berlin Heidelberg, 2010. Springer.

[8] COOLAHAN, J. E., M. T. KANE, J. F. SCHLOMAN, R. P. KOOMULLIL, A. M. SHIH, I. YASUSHI, E. I. KAISAR, K. K. WALSH and M.a M ABDULLAH: *Design of an urban chemical disaster simulation federation for preparedness and response*. In *2007 Fall Simulation Interoperability Workshop (SIW)*, 2007.

[9] D. KRAJZEWICZ, J. ERDMANN, M. BEHRISCH AND L. BIEKER: *Recent Development and Applications of SUMO - Simulation of Urban MObility*. International Journal On Advances in Systems and Measurements, 5(3&4):128–138, December 2012.

[10] FRANCA, R. D. S., M. D. G. B. MARIETTO, W. R. DE SANTANA and G. KOBAYASHI: *An agent-based simulation model for pedestrian unidirectional movement*. In *Applications of Digital Information and Web Technologies, 2009. ICADIWT '09. Second International Conference on the*, pages 375–379, Aug 2009.

[11] GILMAN, C. and R. GILMAN: *Shared-use streets: An application of "shared space" to an american small town*. In *3rd Urban Street Symposium*, pages 1–15, 2007.

[12] GIPPS, P.G.: *A behavioural car-following model for computer simulation*. Transportation Research Part B: Methodological, 15(2):105 – 111, 1981.

[13] GOODWIN, PHIL: *The economic costs of road traffic congestion*. 2004.

[14] HAMILTON-BAILLIE, B: *Towards shared space*. Urban Design International, 13(2):130–138, 2008.

[15] HITCHCOCK, G., B. CONLAN, D. KAY, C. BRANNIGAN and D. NEWMAN: *Air quality and road transport: impacts and solutions*. 2014.

[16] IEEE: *Standard for Modelling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (OMT) Specification*. Technical Report 1516.2, IEEE, 2010.

[17] KEPHART, J. O. and D. M. CHESS: *The vision of autonomic computing*. Computer, 36(1):41–50, 2003.

[18] LEVESQUE, J., F. CAZZOLATO, J. PERRON, J. HOGAN, T. GARNEAU and B. MOULIN: *CAMiCS: civilian activity modelling in constructive simulation*. In *Proceedings of the 2008 Spring simulation multiconference*, pages 739–744. Society for Computer Simulation International, 2008.

[19] MACEDO, J., Z. KOKKINOGENIS, G. SOARES, D. PERROTTA and R. J. F. ROSSETTI: *A HLA-based multi-resolution approach to simulating electric vehicles in simulink and SUMO*. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 2367–2372. IEEE, 2013.

[20] O'FLAHERTY, C.A. (editor): *Transport planning and traffic engineering*, chapter Traffic planning strategies, pages 132–153. Oxford: Butterworth Heinemann, 1997.

[21] PAPAGEORGIOU, MARKOS, CHRISTINA DIAKAKI, VAYA DINOPOULOU, APOSTOLOS KOTSIALOS and YIBING WANG: *Review of road traffic control strategies*. Proceedings of the IEEE, 91(12):2043–2067, 2003.

[22] PIVOTAL SOFTWARE, INC: *Spring Framework Reference Documentation*. http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/, 2015. [Online; accessed January-2016].

[23] POKORNY, T. and M. FRASER: *The portico Project*. http://www.porticoproject.org/comingsoon/, 2014. [Online; accessed January-2016].

[24] PUCH, S., M. FRÄNZLE, J.-P. OSTERLOH and C. LÄSCHE: *Rapid Virtual–Human–in–the–Loop Simulation with the High Level Architecture*. In *Proc. of Summer Computer Simulation Conference (SCSC 2012). The Society for Modeling & Simulation Int., Curran Associates, Inc., Genua*, volume 44, pages 44–50, 2012.

[25] SHEARER, D.N.: *Shared Spaces in New Zealand Urban Areas: A Thesis Submitted for the Degree of Master of Surveying, School of Surveying*. University of Otago, 2010.

[26] SIMULATION INTEROPERABILITY WORKSHOP: *Design of a bioterrorism crisis management simulation for preparedness and response*, San Diego, CA, 2009.

[27] TRANSPORT, DEPARTMENT FOR: *Transport Energy and Environment Statistics*. Technical Report, DfT Publications, 2011.

[28] ZHANG, H.M. and T. KIM: *A car-following theory for multiphase vehicular traffic flow*. Transportation Research Part B: Methodological, 39(5):385–399, 2005.

9 Towards the modelling of the approaching behaviour at controlled intersections

Michael Behrisch and Laura Bieker; German Aerospace Center
{Michael.Behrisch, Laura.Bieker}@DLR.de

9.1 Abstract

Sparse real world data at controlled intersections are analyzed and compared to simulated vehicles in SUMO (2; 1). As indicator the speed over time profile at a fixed distance in front of the traffic light is investigated. The results of the evaluation can be used to improve the approaching behaviour in the simulation.

Keywords: Vehicle Modeling, Traffic simulation, Real world data, Speed profile

9.2 Introduction

Traffic simulations can help to understand and improve complex traffic dynamics. Normally, real world traffic data are used to build a model for traffic simulation. The SUMO (Simulation of Urban MObility) package provides a large variety of traffic simulation tools and models. The vehicle driving dynamics can be divided into the following types of models:

- Car following model (5; 6): calculates the speed of a vehicle according to a leading vehicle

- Intersection model (4): describes the vehicle interaction at an intersection in regard to right of way rules, gap acceptance and avoiding junction blockages.

- Lane-changing model (3): determines the traffic behavior how and when drivers are changing the lane at multi-lane streets.

All of these models do interact with each other and togethr determine the acceleration and speed of the vehicle. Until now the approaching behavior at intersections was neglected in these models or rather subsumed under the assumption that it does not differ much from approaching a standing vehicle (or obstacle). This results in this event being covered mostly by the car following model when "following" a car with speed 0. It can be assumed that many drivers are looking ahead and adapt their speed when approaching an intersection especially at controlled intersections. Furthermore their speed and behavior will largely depend on the current and the expected traffic light phase. This paper evaluates real world traffic data to improve the traffic dynamic model. Improvement with this model can help to improve the quality of the traffic simulation and for further research questions (e.g. how does emergency vehicles are approaching an intersection in comparison to the normal approaching behavior).

9.3 Scenario description

The Institute of Transportation Systems in Berlin runs the large scale experimentation and detection facility Urban Traffic Laboratory (UtraLab) in Adlershof near the premises of the institute. It consists of a number of sensors (among others induction loops and optical sensors as well as detectors for wireless devices) on a public street for detailed measuring of traffic conditions. The street connects one of the major arteria in the southeast of Berlin to the highway leading to the airport in Schoenefeld. It has at least two lanes per direction and especially the eastern direction is regularly crowded in the afternoon peak.
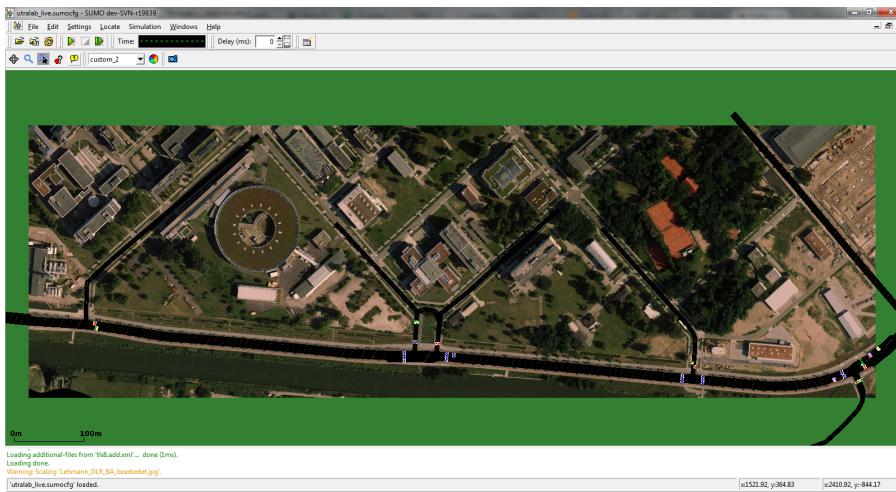
Figure 9.1: Berlin: UtraLab Scenario

The UtraLab also covers a single traffic light intersection where a minor road from the south enters the connecting street. The street is nevertheless important because one of the bus lines which connects the science and business park of Adlershof to the surrounding residential areas uses that street and has a prioritization on the traffic light.

Figure 9.2: Controlled intersection within the UtraLab scenario

All the incoming and outgoing lanes of the intersection are covered by induction loops which cannot only give aggregated data on vehicle counts and velocities but can also give individual events. The simulation is driven by induction loop events at the border of the scenario. Every induction loop signal will trigger the insertion of a vehicle. In order to evaluate the speeds near the traffic lights, the simulation contains simulated detectors at the same positions.

9.4 Evaluation of real world and simulated vehicle data

One of the main obstacles in modelling urban traffic is the determination of the correct traffic light phases. So in a first step it is necessary to adapt all traffic lights to at least fit the experimental data. As stated in the scenario description there are local speeds for all vehicles approaching the junction available. It is expected that those speeds correlate with the traffic light phase and show a repeating pattern for each cycle. In order to determine the length of this pattern (the cycle time) the local speeds were first resampled to equal distances of one second by using linear interpolation and then a simple autocorrelation was applied

In order to adapt the simulation to real world data a first evaluation was done only using the traffic light program as it was heuristically determined by SUMO's netconvert. Looking at the second detector on the eastbound lane the picture shown in red in Figure 9.3 emerged for one individual timeline and the blue line for the average speeds. The green line shows the effect of the guessed signal plan of SUMO.
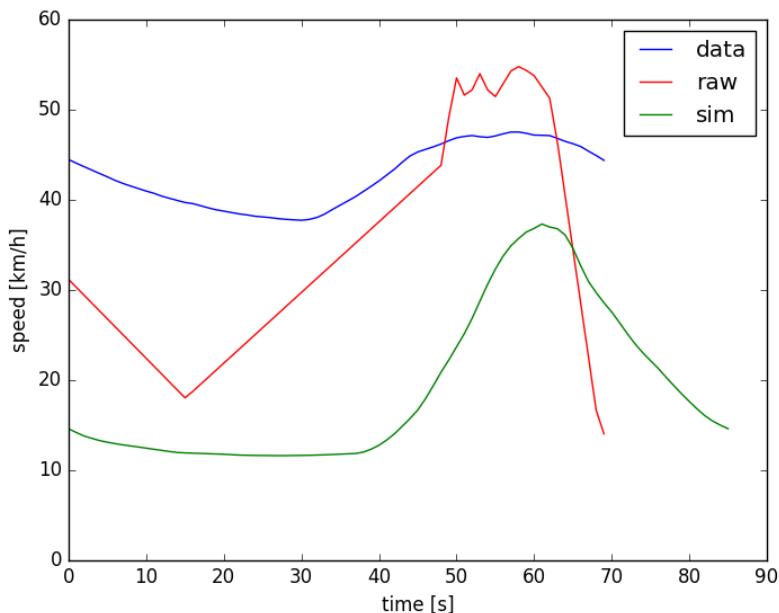
Figure 9.3: Real word speed profile and initial simulation values at example intersection

While these lines seem to be in fairly good agreement the first thing to note is that the gren line is much longer showing that the length of the initial SUMO signal plan deviates considerably from the length of the real plan. In order to determine the length of the real plan a standard auto correlation was applied to the measurements of one hour of vehicle speeds. The differences of the local maxima of this auto correlation were again averaged to increase statistical signifance while keeping the input data small to achieve higher reactivity. This resulted in the final signal plan length of 70 seconds compared to the initial 85 seconds.

Applying this signal plan led to the curves depicted in the left picture of Figure 9.4.

I is clear that the simulation data already gets much closer to the input data but there is still a considerable offset concerning the peak of the speed curve. To fix this, the difference between the maxima of the speed curves was calculated and applied as a signal plan offset to the simulation resulting in the situation of Figure 9.4.

While the overall shape of the two functions now matches very well there is still a considerable offset between the two functions. Furthermore the variation (the difference between the maximum and the minimum values) was reduced significantly whicon the one hand is in better agreement with the
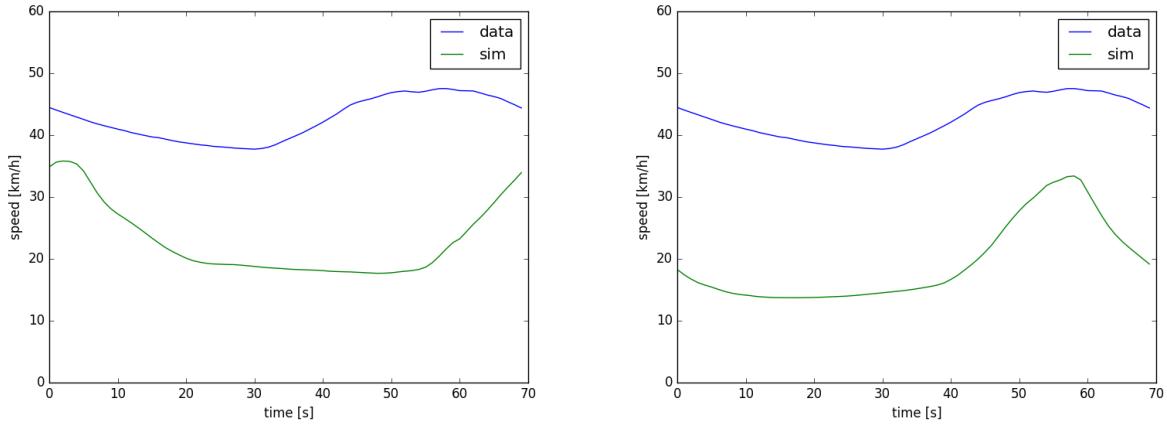
input data but on the other hand may be just an artifact of caluclating means of a lot of situations in both cases and not reflect the real variations at a single traffic light.

One positive effect of this adaption of the traffic light was a significant increas in successfully placed vehicles in the simulation. With the initial plan about 10% of the vehicles could not be inserted because they faced the wrong signal phase and approacheda t a high speed. With the fixed plan this number reduced to 5%.

A validation of the signal plan was done by rechecking the speeds on the opposite direction. Further results willbe published in the final paper.

9.5 Conclusion

The results of this small study can be used manifold. It is clear that it can improve existing traffic simulations to detect signal plans from measured speeds and fosters the implementation of adhoc scenarios. Furthermore it is shown that with relatively simple means (auto correlation and minimum / maximum search) and small datasets of about one hour it is possible to get meaningful results for an improved signal plan which can be a starting point for including traffic light plans without really having their data into a the fully automated scenario generation.

Furthermore and coming back to the original intention there is a big need for getting a more precise description of the approaching behavior to interjunctions. In a next step camera data will be used in addition to the induction loops to improve on the current results.

9.6 References

[1] *SUMO Wiki Webpage - Information for Simulation/Bicycles.*

[2] D. KRAJZEWICZ, J. ERDMANN, M. BEHRISCH AND L. BIEKER: *Recent Development and Applications of SUMO - Simulation of Urban MObility*. International Journal On Advances in Systems and Measurements, 5(3&4):128–138, December 2012.

[3] ERDMANN, JAKOB: *SUMO's Lane-changing model*. In BEHRISCH, MICHAEL and MELANIE WEBER (editors): *2nd SUMO User Conference*, volume 13 of *Lecture Notes in Control and Information Sciences*, pages 105–123. Springer Verlag, 2015.

[4] ERDMANN, JAKOB and DANIEL KRAJZEWICZ: *SUMO's Road Intersection Model*, November 2014.

[5] KRAUSS, S., P. WAGNER and C. GAWRON: *Metastable states in a microscopic model of traffic flow*. Phys. Rev. E, 55:5597–5602, May 1997.

[6] KRAUSS, STEFAN: *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*. PhD thesis, Universität zu Köln, 1998.

10 Integration of an External Bicycle Model in SUMO

Heather Twaddle, Georgios Grigoropoulos, Fritz Busch; Technische Universität München
{heather.twaddle, george.grigoropoulos}@tum.de

Abstract

Bicyclists are amongst the most flexible road users and can tactically choose their pathway across an intersection using available bicycle lanes, roadways and sidewalks, riding either with or against the direction of travel. This flexibility makes it difficult to accurately include the pathfinding behaviour of bicyclists in microscopic traffic simulation tools due to the limitations imposed by the internal design structure of the network elements. In this paper, a method for simulating the pathfinding behaviour of bicyclists crossing an intersection based on observed trajectory data is integrated with SUMO. This is done using a SUMO extension that enables exact road user placement in the simulation. The position coordinates [x,y] of externally simulated bicycles are translated within SUMO and the bicyclist is inserted at the given location on a lane of one of the edges or junctions in each simulation step. This approach for model integration in SUMO is evaluated and recommendations for future research are given.

10.1 Motivation

In urban areas with high volumes of bicycle traffic, the overall traffic efficiency and the capacity of the road network is greatly affected by the actions of bicyclists. This is particularly true at intersections, where streams of traffic held separate on road segments must interact with each other to carry out their desired manoeuvres. Partially compatible and incompatible traffic streams are controlled using either a traffic signal or right of way regulations to optimize intersection efficiency while ensuring road user safety. Unfortunately, bicyclists are disproportionally affected by persisting issues with traffic safety. In 2010, 17.5% of all injured and killed road users in Germany were bicyclists (11) despite an average modal share of bicycling of 10% in urban and rural areas (8). This disproportion is even more pronounced at urban intersections, where according to Gerstenberger (5), 39.1% of all collisions involve a bicyclist. Clearly, overall traffic efficiency and bicycle safety, particularly at urban intersections are important aspects to consider in the assessment of infrastructure planning and traffic management strategies.

Microscopic simulation tools are frequently used to predict the effects of infrastructure design, traffic control and driver assistance systems on traffic safety and efficiency. Considering the growing significance of bicycle traffic, it is essential to include realistic models of bicyclist behaviour in these microscopic simulation tools. Currently, the majority of available simulation tools do not feature a calibrated and validated model for bicycle traffic (7). According to the SUMO wiki web page (1), there is presently no specific model available in SUMO for simulating bicycle traffic and either a modified pedestrian or car model must be used. Additional issues arise in simulating bicycle traffic

realistically in SUMO, including difficulty simulating indirect left turns, simulating bidirectional bicycle lanes and shared space areas. Many of these issues are due to the construction of the network as a collection of one directional links. The network, route and flow computations in SUMO were extended for bicycle and pedestrian flow within the project COLOMBO (3). However, a method for modelling the pathfinding behaviour of bicyclists was not developed within the project as the car pathfinding method was deemed sufficient. In this paper a method for simulating the pathfinding behaviour of bicyclists as they cross an intersection is presented. Issues with indirect left turns, bidirectional bicycle lanes and shared space are addressed.

A secondary motivation of this paper is to investigate the potential of integrating and testing externally developed models in SUMO. Microscopic traffic simulation is improved through the development of new models for simulating and predicting road user behaviour as well as the adaptation and extension of existing models. Model developers generally focus their efforts on one very specific component of traffic simulation. However, it is necessary to embed the developed model within a simulated road environment, including the road network, signal control and simulated behaviour of other road users, in order to develop, verify and validate the new model. There are typically three options available for model development (13); create an independent simulation environment, modify the source code of an open simulation software, or use a commercially or publically available simulation software with a provided API. The first option offers utmost flexibility to the developer in terms of the structural framework within which the model operates. Disadvantages of this approach include considerable development time and lack of comparability or capability with other simulation software. The second option restrains the flexibility of the developer to the framework of the open source simulation but reduces the required development time. The third option requires the smallest investment in terms of development time. Consequently, the developer is at the mercy of the capabilities of the selected software and the documentation of the provided API. In this paper the potential of integrating an independently developed behaviour model operating within a unique framework (option 1) with the open source simulation software SUMO (option 2) will be investigated. A SUMO extension that enables that precise positioning of road users and the interface TraCI (Traffic Control Interface) are used to implement and test a method for simulating the flexible behaviour of bicycles at signalized intersections.

The approach presented in the following section, *Methodology*, offers a means for integrating observed road user trajectories within the traffic simulation software SUMO. In the first sub-section, *Data Analysis*, the methods used for collecting and analysing trajectory data at the study intersection are described and the analysis results are presented. The simulation approach for including bicycle behaviour based on these observed trajectories is presented in the second sub-section, *Simulation Approach*. The integration of the external method within SUMO is explained in the following section, *Integration with SUMO through TraCI*. The results are summarized in the subsequent section followed by a conclusion.

10.2 Methodology

10.2.1 Data Analysis

Video data was collected for four days at a signalized intersection in Munich, Germany with a relatively high volume of bicycle traffic of approximately 800 bicycles/hour. A two hour segment of video data was processed using the open source software *Traffic Intelligence* (10) to extract the trajectories of all observed road users. The resulting trajectory data is stored in an SQLite database, which contains the position and velocity for each tracked road user in each video frame. An automated method for classifying the road users as cars, pedestrians or bicycles based on their positions and dynamic attributes was developed (6) and used. The database was nevertheless

manually controlled to remove erroneous or superfluous trajectories, correct falsely classified road users and reconnect disjointed trajectories. The quality of each of the trajectories was qualitatively assessed to ensure that trajectories with jumps or discontinuities were not included in the analysis. Qualitative variables describing the situation and the actions of the bicyclists that were difficult or impossible to collect using automated processes were manually appended to the trajectory database. The timing of the signal phase changes for the observation days were provided by the City of Munich. An example of the trajectory data overlaid over the raw video is shown in Figure 10.1.
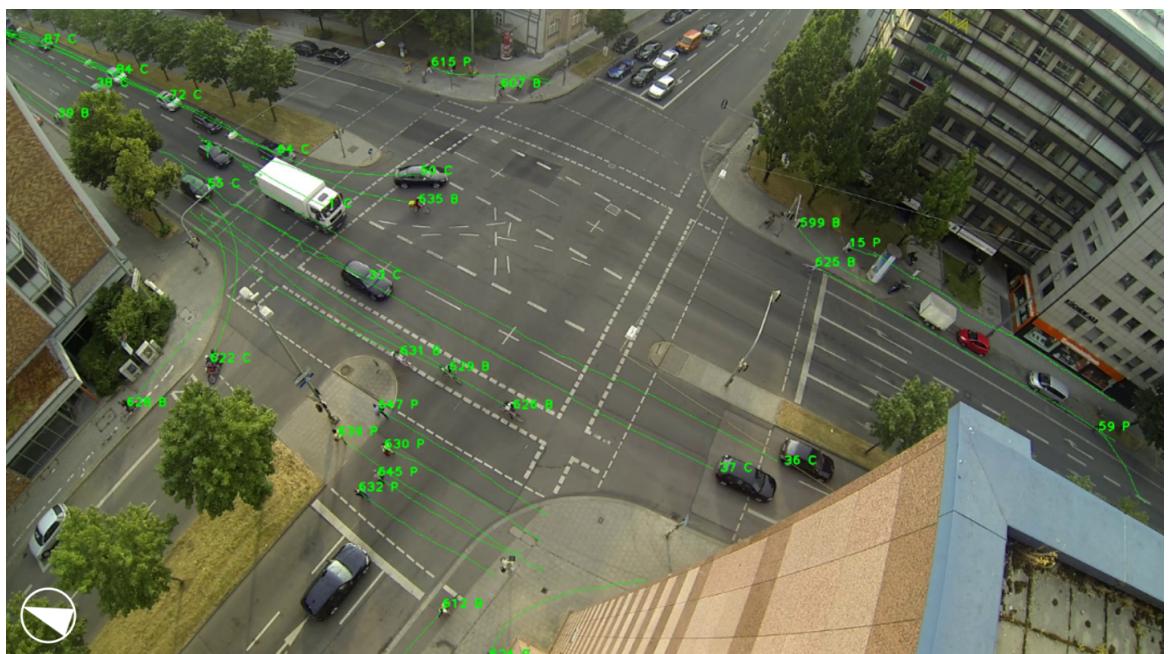

Figure 10.1: Trajectory data extracted using *Traffic Intelligence* (10)

The trajectories were analysed to gain a better understanding of the operational characteristics of bicyclists, which are defined by Michon (9) as subconscious behaviours that take place on a timescale of milliseconds, such acceleration and deceleration (12) as well as maintaining a safe distance to other road users and obstacles. In addition, the tactical decisions of the bicyclists, which are conscious decisions made on a time scale of seconds to minutes to allow road users to cope with the current traffic situation (9), were investigated.

Unlike motorized road users, bicyclists have a number of permissible and prohibited options regarding how, when and where to cross an intersection. The bicyclist can choose between riding on the roadway, the sidewalk or a bicycle lane, if available, either with or against the expected direction of travel. The observed trajectories were classified into 15 groups according to the type of manoeuvre implemented and the infrastructure used upon approaching the intersection. The type of manoeuvres available depended on the desired route across the intersection (left turn, right turn or straight). In this classification structure, right turning bicyclists and bicyclists riding straight across the intersection only had one available manoeuvre, as shown in Figure 10.2. Bicyclists turning left have three available manoeuvres. They can ride with the motorized traffic and complete the turn in one signal phase (direct left turn, manoeuvre 2 in Figure 10.2). Alternatively, they can ride with the pedestrian traffic and can complete their turn in two signal phases (indirect left turn, manoeuvre 3 in Figure 10.2). A third option is a pedestrian style turn but against the allowed direction of travel for bicyclists (indirect wrong way left turn, manoeuvre 4 in Figure 10.2).
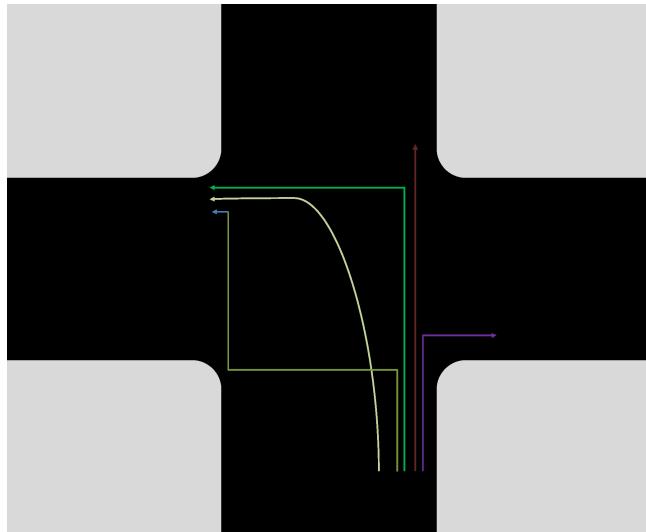
Figure 10.2: Classification of bicyclist manoeuvres

For each of the manoeuvres, the bicyclist can either arrive on a bicycle lane, sidewalk or on the roadway. This classification structure is a very simplified way of grouping the observed trajectories. More complex structures with additional options to include the possibility of bicyclists riding again the direction of travel could also be included in future work. The distribution of the observed bicyclists in the 15 groups is shown in Table 10.1.

Table 10.1: Distribution of observed bicycle manoeuvres

Route	Manoeuvre	Infrastructure	East	North	South	West	Total
Straight	-	Bicycle Lane	126 (97.7%)	499 (99.0%)	519 (92.2%)	121 (90.3%)	1265
		Sidewalk	3 (2.3%)	3 (0.6%)	3 (0.6%)	10 (7.5%)	19
		Roadway	0 (0.0%)	2 (0.4%)	1 (0.2%)	3 (2.2%)	6
	Total		**129**	**504**	**523**	**134**	**1290**
Right turn	-	Bicycle Lane	14 (87.5%)	51 (91.1%)	35 (85.4%)	2 (28.6%)	102
		Sidewalk	2 (12.5%)	5 (8.9%)	6 (14.6%)	4 (57.1%)	17
		Roadway	0 (0.0%)	0 (0.0%)	0 (0.0%)	1 (14.3%)	1
	Total		**16**	**56**	**41**	**7**	**120**
Left turn	Direct	Bicycle Lane	0 (0.0%)	1 (1.3%)	0 (0.0%)	1 (5.6%)	2
		Sidewalk	0 (0.0%)	0 (0.0%)	1 (3.3%)	0 (0.0%)	1
		Roadway	0 (0.0%)	3 (4.0%)	2 (6.7%)	0 (0.0%)	5
	Indirect	Bicycle Lane	16 (40.0%)	29 (38.7%)	14 (46.7%)	4 (22.2%)	63
		Sidewalk	1 (2.5%)	1 (1.3%)	0 (0.0%)	0 (0.0%)	2
		Roadway	0 (0.0%	0 (0.0%	0 (0.0%	0 (0.0%	0
	Indirect wrong way	Bicycle Lane	22 (55.0%)	35 (46.7%)	11 (36.7%)	11 (61.1%)	79
		Sidewalk	1 (2.5%)	0 (0.0%)	1 (3.3%)	2 (11.1%)	4
		Roadway	0 (0.0%)	6 (8.0%)	1 (3.3%)	0 (0.0%)	7
	Total		**40**	**75**	**30**	**18**	**163**
	Total		**185**	**635**	**594**	**159**	**1573**

The trajectories from two observed bicyclists were selected from each of the 15 groups. Bicyclists

were selected which were tracked for a large portion of their crossing manoeuvre (long trajectory) and carried out an average manoeuvre in relation to other bicyclists in the same group. This was determined by plotting all the observed trajectories in each group and qualitatively selecting the most representative trajectory. Two observed bicyclists were selected, one that stopped at the traffic signal and another that did not. For groups with very few observations, the available trajectory in the group was used and an addition trajectory was created for the missing case. Examples of two trajectories from bicyclists travelling straight on the south approach on the bicycle lane are shown in Figure 10.3. The trajectory on the right is from a bicyclist who stopped at the signal and the trajectory on the left is without a stopping manoeuvre.
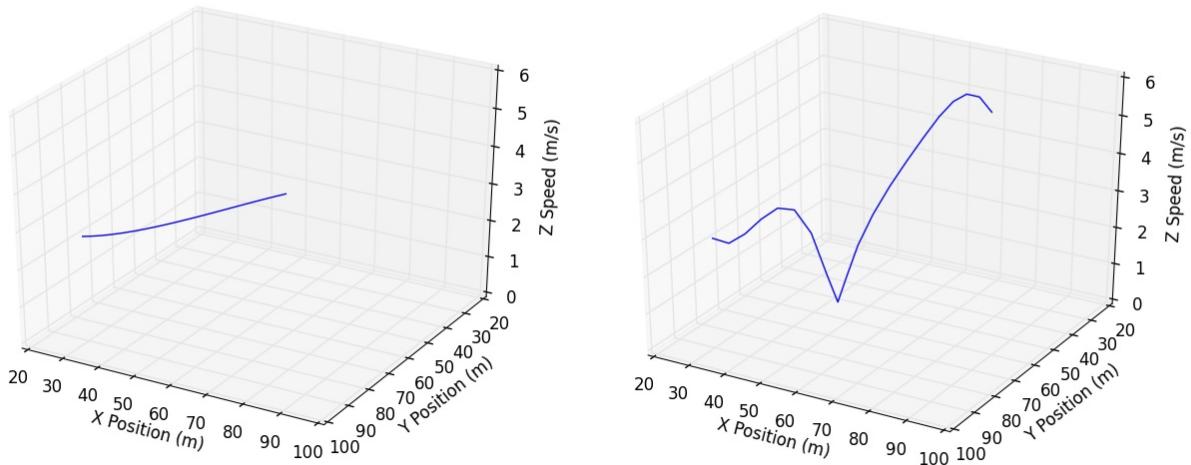
Figure 10.3: Examples of selected trajectories from bicyclists travelling straight on the south approach without stopping (left) and with a stop at a red signal (right)

10.2.2 Simulation Approach

The concept behind the simulation approach is to directly integrate the observed behaviour of bicyclists at an intersection with microscopic traffic situation SUMO. The representative trajectories selected for the 15 groups of tactical manoeuvres are used as 3D guidelines for the simulated bicyclists. In each time step, an action is selected for the simulated bicyclist that will result in the lowest cost, as defined in Equation 10.1. This approach is quite similar to the discrete choice model of pedestrian behaviour proposed by Antonini, Bierlaire and Weber (4).

$$Cost_\alpha = \beta_{dist}dist_\alpha + \beta_{\Delta v}\Delta v_\alpha + \beta_{iSG}iSG_\alpha \tag{10.1}$$

Where:

$Cost_\alpha$ = cost of carrying out a given action α
$dist_\alpha$ = distance between the position after action α and the guideline (m)
v_{obs} = velocity at a given point on the guideline (m/s)
v_α = velocity after carrying out a given action α (m/s)
$\Delta v_\alpha = \mid v_{obs} - v_\alpha \mid$
iSG_α = inverse space gap (m^{-1}) $(iSG_\alpha = 0$ if $SG \geq 5m$, $iSG_\alpha = \infty$ if $SG \leq 0m)$
$\beta_{dist}, \beta_{\Delta v}, \beta_{iTTC}$ = weighting parameters $(m^{-1}, s/m, m,$ respectively$)$

The cost of an action is a unitless measure that combines weighted terms to account for the physical distance from the 3D guideline in the x,y plane, the difference between the current speed and the

speed defined by the guideline (z plane) and the inverse distance (space gap) to other road users or objects. Each of these three terms is weighted using a β parameter that carries the inverse unit of the term, as done in discrete choice modelling approaches. For example, the unit of β_{dist} is m^{-1}. The values of the β parameters can be varied throughout a population of bicyclists to account for differences in personality and riding preferences. The distance from the 3D line is broken into two terms, physical distance and speed difference, in order to enable the application of different weighting values (β parameters.) In this trail implementation, β_{dist}, $\beta_{\Delta v}$ and β_{iTTC} were set as 0.3 m^{-1}, 1 s/m and 5 m, respectively. To reflect real behaviours, these values must be calibrated. An approach similar to that carried out by Antonini, Bierlaire and Weber (4) could be used to calibrate the model based on observed data. A value of zero is assigned to the inverse space gap if the distance is greater than 5 m based on the judgement of the authors. This limitation is not essential, however, as the influence of the term decays exponentially.

An action is defined here as the acceleration or deceleration in combination with the change in steering angle and is expressed using an [acceleration, angle change] pair. The range of acceleration and deceleration values is defined and divided into a set of discrete values. For example, an acceleration range of -1.0 m/s^2 to 1.0 m/s^2 can be defined and divided into the following set of five discrete values {-1.0, -0.5, 0.0, 0.5, 1.0 m/s^2}. The same approach is taken to discretize a given range of direction changes. In this example a minimum and maximum direction change of $\pm\pi/6$ is divided into set of five values {$-\pi/6$, $-\pi/12$, 0, $\pi/6$, $\pi/12$}. The values in both sets are combined to form a complete choice set of action pairs with 25 options in total. The cost of each action is calculated using Equation 10.1 and the action with the lowest cost is selected and executed. If more than one action is found to have the minimum cost, the action with the acceleration or deceleration closest to zero and smallest change in steering angle is selected. Two controls are applied in the creation of the choice sets to prevent unrealistic behaviour from occurring in the simulation. Firstly, no deceleration can be carried out that will result in a speed less than zero. Secondly, if a simulated bicycle has a speed of zero, a restraint is enacted to prevent a change in angle, which may result in the bicyclist spinning on the spot.

10.2.3 Integration with SUMO through TraCI

The study intersection was simulated in SUMO to test the developed method. The simulated traffic volumes for pedestrians, cars, trucks and bicycles and turning rates of all modes were based on traffic counts from the video data. The intersection is controlled using a fixed cycle control and was created in SUMO based on observed signal changes in the video. All road users are simulated using default models for the given vehicle types in SUMO. A plan of the study intersection provided by the City of Munich (left) and a picture of the intersection simulated in SUMO (right) are shown in Figure 10.4.

TraCI was developed to offer a generic interface for connecting a road traffic simulator with a network simulator (2). Using this interface in conjunction with SUMO, it is possible to extract and manipulate many attributes of the simulated road users and environment. Here, TraCI is used to extract the position of the other road users and the phase of the traffic signal and to manipulate the position and speed of the externally controlled bicyclists. The externally controlled bicyclists are steered using the default SUMO model until they reach the intersection. The bicyclist is detected upon approaching the intersection and is lead across the intersection using the proposed simulation approach. Once the manoeuvre is complete, the bicyclist is returned to SUMO control. In order to realize this, pick-up lines are defined approximately 10 m back from the stop lines of the intersections. Upon crossing the pick-up line, the bicyclist is assigned a manoeuvre (column 2 of Table 10.1) based on the route assigned by SUMO (column 1 of Table 10.1) and the probability of the observed manoeuvres in the video data, as given in Table 10.1. If the light is green, the guideline without a stop is assigned (left in Figure 10.3.) If the light is red upon crossing the pick-line, the bicyclist is
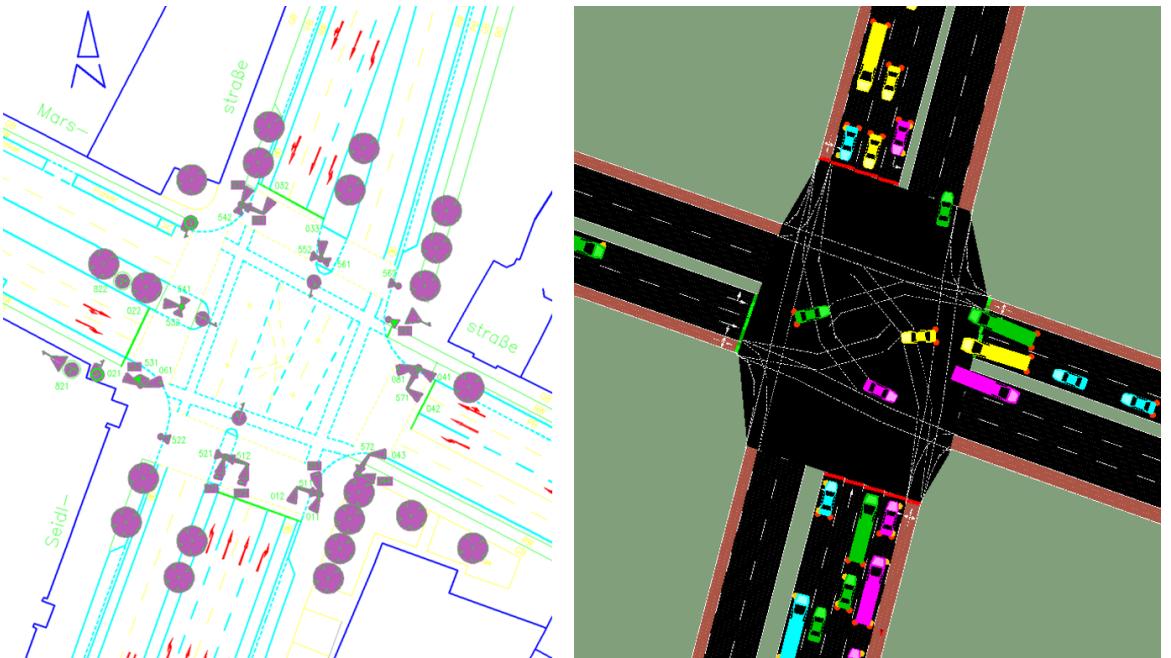
Figure 10.4: Study intersection Marsstraße/Seidlstraße plan (left) and SUMO simulation (right)

assigned the guideline with a stop (right in Figure 10.3.) If the light changes while the bicycle is between the pick-up line and the stop line, the guideline is switched to the respective phase. The selected guideline is extended to the point at which the bicyclist crosses the pick-up line. From this point on, the position and speed of the bicyclist are controlled through the *moveToVTD* and *setSpeed* SUMO functions in accordance to the external simulation approach. The bicyclists are steered using the external model until they reach the predefined drop-off line of their desired route. The bicyclists are introduced to the network using the standard SUMO approaches for creating bicycle traffic flows on the edges and assigning routes.

One issue in integrating an external model with a simulation tool is the alignment of the coordinate systems of the two environments. The best option is to develop both environments using the same coordinate system. However, in some cases, the externally developed model operates on a uniquely defined coordinate system and adjusting this system to fit the coordinate system of the simulation environment is difficult. In this case, the 3D trajectories extracted from the video data are given in a local coordinate system in which the point [0 px, 0 px] is at the top left corner of the video frame. The movement of the road users, which is originally measured in pixels per frame, is translated to a local coordinate system in meters using a perspective transformation of the coordinates and a measured meter per pixel value. The resulting position and speed data is based on a coordinate system defined by this initial perspective transformation. Unfortunately, it is relatively difficult to construct a road network in SUMO based on this coordinate system.

To address this issue, the same perspective transformation approach that was implemented to translate the video frame to the local coordinate system was used. Using the function *findHomography* of the open source library *OpenCV* (http://opencv.org), a perspective transformation matrix H is found between two plans defined by a number of corresponding points. The coordinates of points that can be identified in both the model (original plane) and simulation (target plane) environments, such as start and end points of stop lines at intersections, are input into the function. The resulting perspective transformation matrix H is then used to translate the coordinates of the guidelines to those of the simulation environment.

10.3 Results

The initial implementation of the external method for simulating bicycle traffic based on observed trajectory data in the simulation tool SUMO delivered promising results. The placement accuracy of the externally controlled bicyclists in SUMO was found to be highly dependent on the location of the bicyclist. Small placement errors ranging between 0.1 cm and 0.3 cm were found on road edges, regardless of the lateral resolution. Within the junction, error values were found to be similarly small. However, much larger placement errors were observed when an externally controlled bicyclist crossed from a road edge into a junction. During these transition phases, much larger placement errors of up to 6.7 m were observed. This can be seen in the jumps in the trajectories in Figure 10.5 (right.) Without the extension for exact vehicle placement, the average distance between the desired position of the externally controlled bicyclist and the position realized in SUMO was found to be 3.70 m. The extension was found to reduce this error to 0.40 m, a reduction of 89.2% of the placement error. In the figure below, the resulting modelled trajectories using the cost function in Eq.10.1 (left) and the resulting trajectories in SUMO (right) are shown. A qualitative assessment of the proposed simulation approach was carried out and the method was deemed to produce realistic trajectories with variation due to the position and actions of other road users. The trajectories realized in the simulation SUMO reflect the difficulties expressed above; the positioning of externally controlled road users near the edges of the junctions tends to be problematic. The placement error was not found to be correlated with the type of manoeuvre.
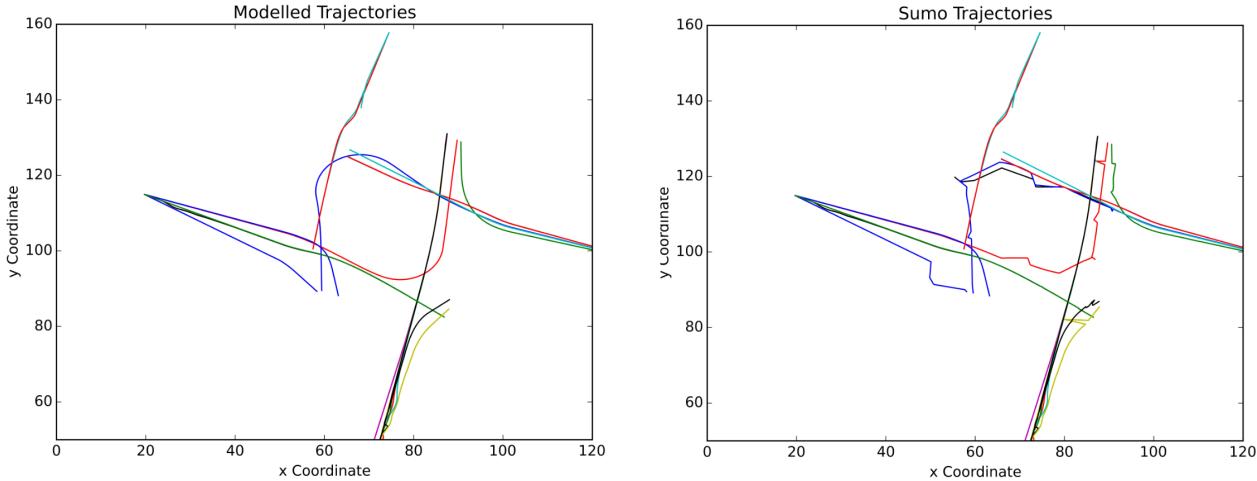
Figure 10.5: Modelled trajectories (left) and the realized SUMO trajectories (right) of the externally controlled bicyclists

The simulated road users in SUMO were found to be able to detect and react to the externally controlled bicyclists in the same way that they react to other SUMO controlled road users. Even in cases where the externally controlled bicyclists moved against or perpendicular to the direction of travel on a given edge, the SUMO road users stopped and waited for the road user to pass. The reaction of the externally controlled bicyclists to the SUMO simulated road users was found to be satisfactory. The current approach uses the position of the simulated road user as provided by TraCI, which is the front middle point of the vehicle. The shape of the simulated road user is not taken into account. As a result, the externally controlled bicyclists react very well to short and narrow road users (other bicyclists), but not as well to larger and wider vehicles (trucks.)

10.4 Conclusion

The paper describes an approach used for integrating a simulation method based on observed trajectory data with the microscopic simulation tool SUMO. The approach enables the simulation of realistic trajectories of bicyclists crossing signalized intersections, both with regard to the speed and the position. Although the bicyclists are not assigned explicitly to edges in the SUMO simulation, the TraCI function *moveToVTD* automatically translates the x,y coordinates of the externally controlled bicycle to the nearest point along an existing lane and edge. The other road users in the simulation can therefore react to the presence of the externally controlled bicyclists, even if they are moving in the wrong direction across or along a given edge.

The implementation of the model within the SUMO simulation environment was deemed to be successful enough to use for further development and validation of the external approach for simulating bicycle traffic. The simulation approach itself has not yet been validated or compared to other simulation models for bicycle traffic. Although this approach offers a good starting point for integrating observed trajectory data into a microscopic traffic simulation, a number of potential improvements have been identified for future work. Currently, the choice between manoeuvres given a route across the intersection is random according to the observed probabilities of that manoeuvre being chosen. It is hypothesized, however, that the choice between manoeuvres is dependent on both the personal characteristics of the bicyclist and the current traffic situation in which the bicyclist finds himself or herself. The realism of the simulation could be improved by introducing more advanced choice models that take into account the geometric attributes of the intersection, the presence and actions of other road users and the state of the traffic signal. The signal timing and the time of arrival are likely highly correlated with the type of manoeuvre carried out, particularly for left turning bicyclists. The approach could be further improved by introducing an unsupervised learning approach for clustering manoeuvres based solely on the observed trajectory data and not on a predefined classification structure. In this paper, a representative trajectory was selected for each manoeuvre. The generality of the approach would be greatly improved if a method for combining the observed trajectory data into an average 3D trajectory was developed.

10.5 References

[1] *SUMO Wiki Webpage - Information for Simulation/Bicycles.*

[2] A. Wegener, M. Piorkowski, R. Maxim H. Hellbrück S. Fischer and J. P. Hubaux: *TraCI: An Interface for Coupling Road Traffic and Network Simulators.* In *Proceedings of the 11th Communications and Networking Simulation Symposium*, pages 155 − 163, 2008.

[3] D. Krajzewicz, J. Erdmann, J. Hï¿½rri and T. Spyropoulos: *Including Pedestrian and Bicycle Traffic in the Traffic Simulation SUMO.* In *10th ITS European Congress*, page 10, 2013.

[4] G. Antonini, M. Bierlaire and M. Weber: *Discrete choice models of pedestrian walking behavior.* Transportation Research Part B: Methodological, pages 667 − 687, 2006.

[5] Gerstenberger, M.: *Accidents at Intersections: Basic Analysis of Causation and Approaches for Improvements.* Elektronik, 2015.

[6] H. Twaddle, T. Schendzielorz, O. Fakler S. Amini T. U. München H. Boesefeldt and T. Gmbh: *Use of automated video analysis for the evaluation of bicycle movement and interaction.* In *Proc. SPIE 9026, Video Surveill. Transp. Imaging Appl. 2014*, 2014.

[7] H. Twaddle, T. Schendzielorz and O. Fakler: *Bicycles in Urban Areas: Review of Existing Methods for Modeling Behavior.* Transp. Res. Rec. J. Transp. Res. Board, pages 140 – 146, 2014.

[8] infas Institute für angewandte Sozialwissenschaft and DLR: *Mobilität in Deutschland 2008*, 2010.

[9] Michon, J. A.: *Critical view of driver behavior models: what do we know, what should we do?* In *Human behaviour and traffic safety*, pages 485 – 520, 2014.

[10] S. Jackson, L. Miranda-Moreno, P. St-Aubin and N. Saunier: *A Flexible, Mobile Video Camera System and Open Source Video Analysis Software for Road Safety and Behavioural Analysis.* Transp. Res. Rec., pages 90 – 98, Vol. 2365,no. January 2013.

[11] Statistisches Bundesamt: *Unfallentwicklung auf Deutschen Straßen*, 2011. Beuth Verlag GmbH, Berlin.

[12] Twaddle, H. and G. Grigoropoulos: *Modeling the speed, acceleration and deceleration of bicyclists for microscopic traffic simulation.* In *To be published in Transportation Research Record (TRR), Journal of the Transportation Research Board*, 2016.

[13] W. Schakel, B. Van Arem, H. Van Lint and G. Tamminga: *A Modular Approach for Exchangeable Driving Task Models in a Microscopic Simulation Framework.* In *Intelligent Transportation Systems-(ITSC) 16th International IEEE Conference*, 2013.

11 Simulation framework for testing ADAS in Chinese traffic situations

Marc Semrau[1], Jakob Erdmann[2], Bernhard Friedrich[3] and René Waldmann[1];
[1]Volkswagen AG, Berliner Ring 2, 38440 Wolfsburg, Germany
[2]German Aerospace Center, Rutherfordstr. 2, 12489 Berlin, Germany
[3]Institut für Verkehr und Stadtbauwesen, Technische Universität Braunschweig, Hermann-Blenk-Straße 42, 38108 Braunschweig, Germany
{Marc.Semrau}@volkswagen.de, Jakob.Erdmann@dlr.de ,

Abstract

This paper describes a new simulation framework assisting the development of ADAS optimized for Chinese traffic. The framework couples the traffic simulators SUMO and VTD to combine different scales and fidelities. SUMO focuses on traditional European and American traffic, making it necessary to implement, calibrate and validate new traffic models. To this end, two driver studies conducted in China are described first. It is stated which characteristics in the driving behavior can be found and how they are reproduced in the simulation. Subsequently, the development and the implementation of the simulation framework are presented and test results are discussed. The focus on Chinese traffic demands a sophisticated combination of classic microscopic traffic simulations and more detailed considerations of the ego vehicle surroundings which are both integrated in the presented framework.

Keywords: Heterogeneous traffic, Driver Model, Testing ADAS

11.1 Introduction

Recently a new field of application for traffic simulation and driver modelling is arising, namely testing and developing advanced driver assistant systems (ADAS). When developing new ADAS or improving existing ones, a lot of real driving tests are carried out, since they deliver the highest degree of precision. With rising complexity this becomes not only more and more expensive but is also extremely time consuming. Consequently, time restrictions often render comprehensive real world tests cumbersome, which is why simulations are indispensable.

So far traffic simulations have mainly been used to evaluate traffic measures and new technologies concerning traffic flow. For this purpose even the most detailed traffic simulations, the microscopic simulations, have a reduced number of parameters, which minimize the effort of calibration and still fulfill the requirements of the stated use cases: traffic measures and new technologies concerning traffic flow (10). The emulation of vehicle dynamics and the interactions between vehicles in the direct surroundings is necessary for testing and developing ADAS, since those systems are not able to work without this information. This kind of simulation is named sub microscopic simulation (9). Up to now sub-microscopic simulations are mostly used for scenario based testing. This kind of simulation is quite common in the development process of driver assistant systems (11). Schiller et al. states, that the current emphasis is primarily on the simulation of individual vehicles at a very

detailed level (3). Additionally sub microscopic simulators like Virtual Test Drive (VTD) are able to reproduce the same scenarios several times in the exact same way (12). This approach has one major disadvantage: It is limited to few situations and there is no real interaction of the surrounding vehicles with the ADAS controlled vehicle. This may be acceptable for safety validation of ADAS but for improvement and development a closed loop simulation with consideration of real traffic is needed.

To the best of our knowledge there is no work focusing on the coupling of a traffic simulation developed and calibrated for Chinese traffic with a sub-microscopic simulation that aims for testing ADAS. This issue becomes even more relevant when considering other traffic situations than typical domestic market scenarios. When developing ADAS optimized for local conditions in different countries the effort of real driving tests rises significantly, which makes the manual definition of relevant scenarios nearly impossible, at least for foreigners. To prove this point and give a comprehensive example, this paper describes two driver studies, which were carried out in China. Afterwards, characteristics of the examined driver behavior are described. These characteristics represent the basic knowledge for the design of the simulation framework. Subsequently the implementation of the framework is described and extensions of the used simulation tools are realized. Calibration and test of the framework will complete section 11.3. Section 11.4 concludes the paper, by discussing limitations of the developed framework and giving an outlook on future work.

11.2 Chinese traffic

It is necessary to find and analyze characteristics in Chinese traffic to evaluate the usability of existing simulation tools and models. Thus, situations and scenarios that differ from the European market which are relevant for (future) ADAS are investigated. Then, necessary model improvements and extensions are derived.

11.2.1 Studies

To figure out which situations differ from the European market a combination of a survey and driving test in real traffic was carried out. Since little background knowledge on this kind of studies was available before, we chose to conduct two consecutive studies. Before these two studies are described, results from the ADAS study that was carried out by Audi in 2013 but analyzed by the author in 2014 (13) are introduced. The focus group study explained in subsection 11.2.1 took place in 2014 and was executed in cooperation with Volkswagen AG and Volkswagen Group Research China. Its purpose was to point out critical situations typical for Chinese traffic which require assistance. The last study described in subsection 11.2.1, was planned and executed in 2015 in cooperation with Audi AG and Audi Group China and is used to analyze and describe the situations, found in the previous studies.

ADAS Study

The ADAS study is a large-scaled study with the aim to investigate existing driver assistant systems (DAS) on the Chinese Market, respectively in Chinese traffic. The focus is the evaluation of availability und usability of DAS (including the recording of sensor data) as well as the subjective opinions of the participants. Therefore, it is possible to derive situations in Chinese traffic which cannot be handled by state-of-the-art DAS but assistance was demanded by the test persons. To get representative results the study was carried out with 40 Chinese drivers that where selected due to the typical Chinese driver distribution, concerning age, driving experience and gender. The surveys in this study took part in several phases of the test. The drivers were interviewed before, during and after the driving tests. Additionally, sensor data and data of the vehicle were recorded

to make the analysis of the occurring situations possible. The study showed that all tested DAS exhibited save behaviors. The biggest limitation was the availability during normal traffic conditions in Chinese traffic. For example the adaptive cruise control is working well in free flow and medium dense traffic but the availability can be enhanced in dense traffic and congested situations. This effect is known from German traffic, too. However, these situations are more common in China and the availability of the system is therefore relatively poor. Moreover, the complexity in congested situations is much higher in China. Details on this point will be shown in the following sections. In summary it can be stated that the availability of existing driver assistance is low in congested traffic situations. Since these situations are quite common in China and drivers would like to have assistance in these situations they will be described in detail in the following sections. (13)

Focus Group Study

Since the ADAS study explicitly asked the participants about existing DAS, a subsidiary one, without this (possible) interference, was planned. For this study 24 participants have been selected. Similar to the selection in the ADAS study the typical Chinese driver distribution was the guideline for the selection. The participants were asked nine different questions about their driving experience, habits while driving and dangerous situations they experienced. For the purpose of this paper two questions are relevant:

1. What was the most dangerous situation you ever met?

2. Are there any boring situations on our trip?

The first question aims to find situations which are not usual for German drivers and considered as critical by Chinese drivers. The second one is useful to find out in which situations drivers may be inattentive and therefore need assistance. After the questionnaires have been completed a camera was installed on the front passenger seats of the vehicles of 8 participants to observe the drivers action in the car as well as the view through the front window and the mirrors. The collected videos were analyzed to extract situations addressed by the questions. Following this, the videos were shown to focus groups of six participants to collect insights out of the discussions, about the shown situations, within the focus groups. In summary, lane changes where named as the most dangerous situations. The insights gained out of the focus group discussion are that these dangerous lane changes happen everywhere. Heavy traffic was named as the main reason, sometimes in combination with trucks getting stuck in the fast lane. Interestingly, traffic jams are named as the most boring situations. Thus, we concluded that heavy traffic situations and traffic jams are not only a reason for dangerous driving behavior but also situations with need for assistance. Moreover, this outcome coincides with the knowledge gained in subsection 11.2.1 For this reason these traffic situations will be investigated in detail in subsection 11.2.2.

Traffic Situation Study

After china specific situations as well as the demand for assistance have been identified it is necessary to objectively analyze them. To this end, a study was planned in cooperation with Audi AG and Audi Group China. During the study that was carried out in November 2015, 24 participants drove in Shanghai during rush hour. Again the participants were selected with respect to the real Chinese driver distribution. They were neither told what the study is about nor to behave in a certain way. The purpose was to collect data in the focused traffic situations without any prior charge of the participants. This data will be used to complement the data collected in the ADAS study. The resulting data basis will be used to characterize lane changing in dense Chinese traffic.

11.2.2 Lane change characteristics in dense Chinese traffic

Real sensor data from the ADAS study and the traffic situation study is used to gain knowledge about the characteristics of lane changing in Chinese traffic. The data basis contains around 150 hour of driving tests in real Chinese traffic. In general the data analysis shows that the lane changing behavior strongly depends on the traffic conditions and on the level of driving experience. The difference between novice and expert driver increases with rising complexity of the traffic situations. As a result it is necessary to distinguish between driver types and traffic conditions. Thus, the simulated models must be capable to handle both.

Traffic conditions in China are not always determined by the road type. Therefore the differentiation between highway, country roads and city roads is not used here. Moreover there are a lot more road types occurring in Chinese traffic that do not fit into these standard categories. Due to these circumstances it is distinguished between free flow traffic, dense traffic, congested traffic and traffic jam. Motivation for lane changes in general can be divided into (2):

1. Tactical: lane changes to gain speed

2. Strategic: lane changes to follow the route

3. Cooperative: lane changes for cooperative behavior

4. Obligatory: to follow the keep right rule

The analysis regarding the motivations for lane changes in different traffic conditions is summarized in Table 11.1. The total amount of lane changes under the different traffic conditions is given in brackets and the rows give the amount of the specified lane change type under the give traffic conditions.

Table 11.1: Motivation for lane changes dependent on the traffic situation

Traffic condition / lane change type	Free flow traffic (79.5%)	Dense traffic (17%)	Congested traffic (2.5%)	Traffic jam (1%)
Tactical left	38%	55%	33%	0%
Tactical right	37%	30%	66%	0%
Strategic	23%	10%	0%	100%
Cooperative	1%	5%	0%	0%
Obligatory	1%	0%	0%	0%

The table also proves the strong dependency on traffic conditions. The most obvious fact is that there are nearly no lane changes observed in traffic jam. The only type of lane change that occurs in traffic jam is the strategic lane change, whereas tactical lane changes are much more common in dense and free flow traffic. This has two major reasons. In traffic jams the possible speed gain is usually lower than in dense traffic situations and the possibilities for lane changes are also relatively rare. Thus, the motivation for changing the lane is not big enough to perform tactical lane changes. This is different for lane changes in congested traffic. Here, the observed drivers only performed tactical lane changes. This behavior is also known from German traffic, where drivers want to get on the (apparently) faster lane in congested traffic (1). Consequently, this has to be treated by the simulation model. Moreover, it is obvious that there is no difference between tactical

overtaking on the left and the right side in free flow traffic. Therefore, the lane changing models have to consider this characteristic and evaluate both neighbor lanes in view of tactical advantages. Surprisingly speeding is not common in the observed situations; vehicles mostly stick to the speed limit or exceed it just little. This has to be considered, when the tactical advantages of lane changes are evaluated. In dense and congested traffic the amount of lane changes to the left and the right is not equal. Anyway lane changes to the left and the right are performed frequently. In dense traffic more tactical lane changes to the left are performed. One possible reason is an average speed gain on the left lane. In contrast this seems to prove wrong for congested traffic.

Strategic lane changes should be similar under most circumstances but in the observed situations it often happens that drivers obviously miss to observe their route for some time and therefore have to change lanes quickly to follow the original route. Due to this characteristic a lot of critical lane changes occurred. It depends on the traffic conditions how critical such lane changes are. Thus, this has to be considered in the simulation of strategic lane changes. Driving between lanes is another special characteristic. In a lot of situations there are more lanes in reality than painted as road marks.
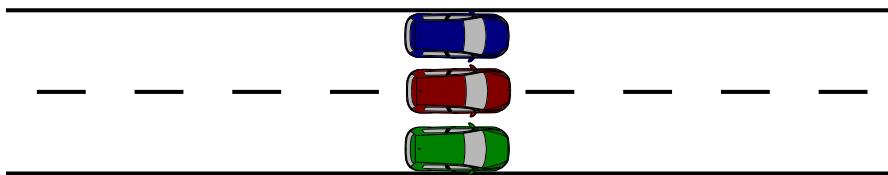

Figure 11.1: driving between lanes

For a better understanding Figure 11.1 shows such situation. It can be seen that the cars do not stick to the marked lanes. As long as they fit next to each other they just create their own lanes (the reason for this behavior is not visualized in the picture). This manner determines an extension in the lane changing logic, which is described in subsection 11.3.2.

There are only few cooperative lane changes in the observed data, mostly the cooperative changes are forced by other drivers that are very impatient. In these cases the cooperation is limited to the level to just prevent a collision. This behavior is one major reason for driving between lanes.

The last class of lane changes is the obligatory one which means clearing the left lane to obey the keep right rule. In the observed data this lane change motivation was quite rare. Usually this kind of lane change is caused by fast vehicles approaching from behind. Vehicles that still do not clear the left lane may also be one reason for overtaking on the right side.

11.3 Framework

To the best of our knowledge, no existing framework is able to fulfill all the acquired needs of the use case treated here. To meet the presented requirements for testing ADAS in Chinese traffic a co-simulation has been designed and implemented. This co-simulation has to deal with traffic flow simulation and driver assistant system simulations. Therefore, the proposed design combines a classic traffic simulation with a scenario based simulation tool. Moreover, a new instance is introduced for both controlling the simulation and implementing the ADAS.

11.3.1 Design

For microscopic traffic simulation SUMO (Simulation of Urban MObility) was chosen for its power and flexibility. SUMO already includes a set of different driver models and makes it relatively easy

to include additional models. This compares favorably to commercial alternatives such as VISSIM which only includes a single (albeit configurable) model and does not allow modifications of its models.

The sub-microscopic simulation software VTD is widely used for testing ADAS in the automotive industry and will therefore be integrated, too. Schiller et al. did some fundamental work in this area by introducing a multi-resolution simulation (11). In contrast, in the framework presented here VTD is not coupled directly as an equally ranked simulator. To construct a multi-resolution simulation a central control component for the simulation framework is implemented in C++ and executed in ADTF (Automotive Data and Time triggered Framework). This so called simulation master is responsible for the time synchronization, data exchange and the integration of additional components. The described approach has several advantages as the paper focuses on realistic traffic simulation executed by SUMO. As there is no handover of the simulation control between the used tools it is easier to ensure the compatibility of vehicle and driver control. This is in fact an issue, since the drivers in SUMO always follow routes, whereas VTD is not necessarily based on routes. Due to the central simulation master this can be tackled accordingly and it is still possible to modify the behavior of drivers.

VTD is responsible for the visualization and the vehicle dynamics is used to convert the control quantities of coupled ADAS or driver models (like steering angle and acceleration) into new state variables (like position and speed). Moreover, the data of the surroundings of the ego vehicle are transmitted from VTD to the tested ADAS.

The central component has two more benefits. Both, SUMO and VTD do not require any special implementation and thus the usability with all (including future) versions of SUMO and VTD is made sure. Moreover, the implementation of the simulation master described in subsection 11.3.2 ensures the usability with OMNeT++, too. Hence, it is easy to either extend the framework or to change the focus from testing ADAS in traffic to testing ADAS with respect to V2X technologies. Aside from the design of the presented framework it is necessary to guarantee the compatibility of the maps. The most important issue is to ensure the consistency of the coordinates in both maps. The OpenDrive standard is employed in VTD, which is widely used when it comes to testing of ADAS. In contrast, SUMO requires an individual xml based map format. The existing SUMO map converter netconvert allows back-and-forth transformations between SUMO maps and OpenDrive maps. For this project it was improved and extended to ensure a high level of geometrical accuracy when transforming between these formats. It is important to generate maps base on these transformations because independent generation would make it quite hard to ensure consistency of the coordinates. Finally it is indispensable to ensure that the requirements defined by the characteristics of Chinese traffic are met. Therefor an enhancement of the traffic simulation software itself is necessary. The steps to implement the developed design are described in subsection 11.3.2.

11.3.2 Implementation

Simulation Master

The simulation master is the central instance of the presented framework. It controls the simulation framework and it realizes the coupling of the tested ADAS. To meet the requirements of most ADAS a step size of 40 ms has to be realized. This is by far more than classic traffic simulations like SUMO are able to realize. Consequently, a simulation step size of 200 ms is chosen for SUMO. This value seems reasonable since it corresponds to the average reaction time of humans (14). To meet the requirement of tested ADAS the simulated measurement data is taken out of VTD. Nevertheless, VTD is not taking control of the vehicles. The basic assumption is that the driver behavior is not changing within 200 ms because this would be below the reaction time to even notice a change in conditions. The schematic sequence of the simulation is illustrated in Figure 11.2. One can easily see that there is data exchange between VTD and the ADAS every 40 ms whereas the data exchange

with SUMO is triggered every 200 ms. Moreover, the data exchange is visualized. The first frame is
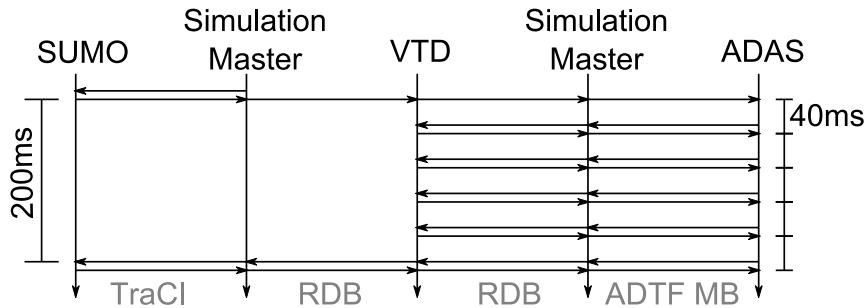

Figure 11.2: schematic sequence of the simulation

requested by the Simulation Master over the Traffic Control Interface (TraCI). The same interface is used by SUMO to send the requested data. The Simulation Master sends this data via the Runtime Data Bus (RDB) to VTD and via the ADTF Message Bus (ADTF MB) to the tested ADAS. Before the next frame of SUMO is requested after 200 ms the environment data is provided by VTD. It is send to the Simulation Master via RDB and forwarded via the ADTF MB to the ADAS.

Driver Models

Before the extensions of driver models are described a brief discussion of the available car following and lane change models is given.

The car following behavior in Chinese traffic is relevant for the lane changing decision and process too, since a vehicle may only change lanes if it can follow the leading traffic in its target lane "safely". The car following model is responsible for defining this safety criterion. Likewise, the following traffic in the target lane must be able to follow the ego vehicle safely. The safety criteria in this case might differ considerably between "Western" and Chinese drivers. The general principle of lane changing is shown in Figure 11.3.

The dependency between longitudinal and lateral movement is also the reason why researchers like
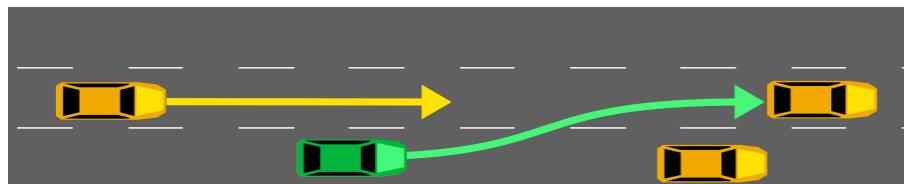

Figure 11.3: lane changing process

Gipps and Kesting et al. developed integrated lane change models (4; 5).

Gipps was the first who developed an integrated lane change model in 1986. His model is based on the principle of Cellular Automaton, but in contrast to other Cellular Automaton models, like the Nagel Schreckenberg model (8), the model of Gipps is continuous in spatial resolution (4). Therefore, it is better suited to a space-continuous microscopic simulation such as SUMO. The integrated lane change model presented by Kesting et al. is named MOBIL (Minimizing Overall Braking Induced by Lane Change) and is based on the IDM (intelligent driver model) which is described in (6).

Due to the fact that Cellular Automaton Models are limited to their spatial resolution they are not suitable for the adaption to Chinese traffic situations. All the described models, which are continuous in spatial resolution, share similar opportunities for adaption. Typically acceleration and deceleration capabilities of the vehicles are important factors. Additionally, the gap acceptance of drivers has

significant influence on the drivers behavior. Besides these factors the reaction time of the drivers is considered when calculating the drivers actions. None of the mentioned models includes a driver perception. Even though there are models that include a basic perception this will not be considered here, since the focus is on dense traffic and the perception limits concerning the distance to other vehicles has minor influence.

SUMO uses the car-following model of Stefan Krauß (7) by default. This model is quite similar to the Gipps model in limiting the driver to those velocities that still allow safe stopping if the leader vehicle comes to a full stop. For this paper a modified version of the Krauß model and an integrated lane change model is used. The reason is the good capability of this model in all the mentioned traffic conditions along with its simplicity (7).

Sub-lane model in SUMO

To meet the requirements of Chinese traffic it is necessary to adapt the fixed concept of lanes in the simulation tool. SUMO has focused primarily on European and American traffic where vehicles have a high rate of lane-following compliance and furthermore, the fraction of "narrow" two-wheeled road users is low. Thus, SUMO vehicles occupy exactly one lane in the lateral direction. To allow for
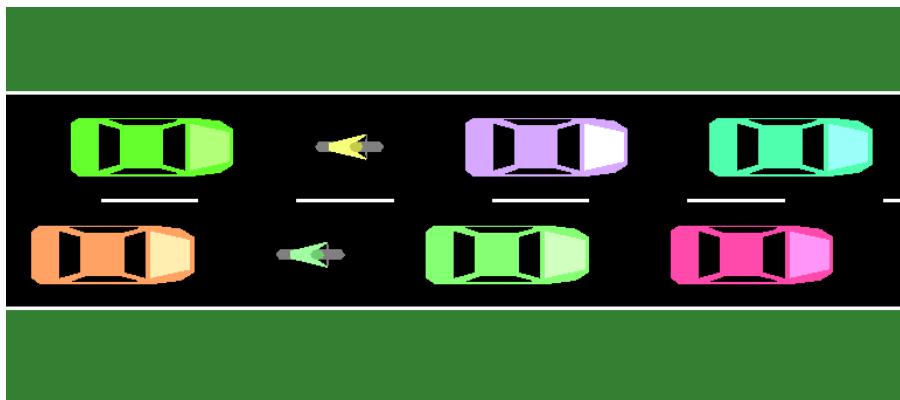

Figure 11.4: standard lane model in SUMO

Chinese traffic dynamics, it is deemed necessary to break up this fixed relationship and allow multiple vehicles on the same lane as well as vehicles which occupy more than one lane. This has benefits for modelling European traffic as well, since it allows a better representation of car-bicycle interactions. In the remainder of this section we describe the extensions to SUMO that were developed to meet these requirements.

Traditional lanes are divided laterally into a number of so-called *sublanes*. Each vehicle occupies a number of sublanes according to its lateral position and width. To maintain the concept of collision free traffic two vehicles may not occupy the same sublane when driving side by side. This necessitates further model changes for longitudinal movement (car-following) as well as lateral movement (lane-changing). Previously each vehicle had at most one immediate leader vehicle. With the introduction of sublanes a vehicle may have multiple immediate leaders (i.e. multiple motorcycles driving side by side on the same lane). Consequently, the car-following model is applied to all leader vehicles and uses the minimum safe speed to ensure safe driving.

It is expected that the existing car-following models can be calibrated to cover the behavior of Chinese drivers. If this assumption proves false, a new model may be necessary here as well.

A novel lane-changing model is required to make use of the sublanes. This model extends the lane-changing model described in (2).
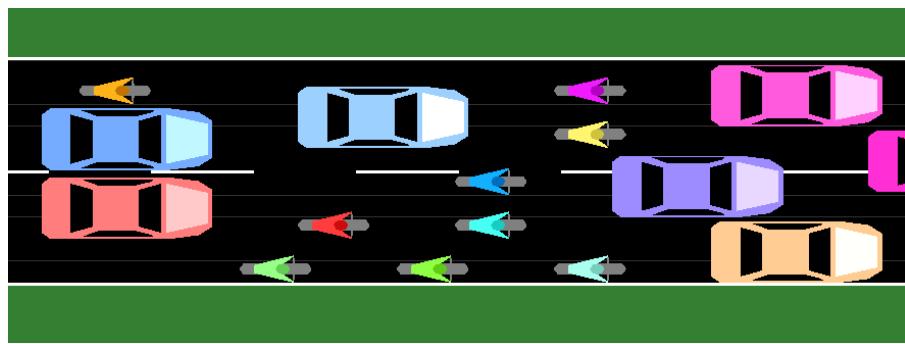
Figure 11.5: sub-lane model in SUMO

The most important changes are:

- The number of possible maneuvers that have to be considered is increased from the number of neighboring lanes (at most 2) to the number of neighboring sublanes (10-100 depending on the configured sublane resolution)

- The larger number of choices requires novel trade-offs to be made: (i.e. should a vehicle attempt to find a gap in the right sublane to increase its speed or should it move immediately to the left sublane where speed is also higher but not as high as to the right)

- When traversing multiple sublanes in a single maneuver, each of the intermediate sublanes must be checked to avoid collisions

- Lane changing motivations are no longer mutually exclusive: A strategic change to the right lane does not preclude sublane-changes within that target lane to optimize for travel speed

In addition to the existing lane-changing motivations, we introduce a new motivation for achieving a desired lateral position within a lane in the absence of more urgent motivations. The behavior is user configurable and includes such behaviors as

- Stay in the center or to a particular side of a lane

- Compact alignment to neighboring vehicles to maximize capacity

Another new aspect that needs to be modelled is lateral distance keeping. This scenario requires safety and speed considerations comparable to challenges of the traditional adaptive cruise control.

11.3.3 Calibration

This section describes the calibration of the driver models running in SUMO with real world data. To this end, the information gathered from the real world measurement data analyzed in subsection 11.2.2 is used. As the focus is on lane changing processes, the lane change motivation is considered first. As stated in section 11.2.2 there are four types of lane changes. Tactical lane changes are carried out to gain speed. The first thing to calibrate is the proportion of left and right overtaking for all speeds. Moreover, it is necessary to adopt the threshold values for the possible speed gain over a certain time. In Chinese traffic these values depend on the traffic situation. For example, there are only few tactical lane changes in traffic jam but a lot in dense traffic situations. The speed gain value already depends on the velocity. In contrast to the mentioned relation the value is rising when the absolute speed is low. This behavior is implemented to model the observed tactical lane changes in congested traffic. Therefore, this relation is reasonable. Nevertheless the number of lane changes in congested traffic is low. The reason is the dense traffic itself: Since the gaps are

relatively small it is complicated for the driver to change the lane. In this context the impatience of a driver is an important factor. The impatience is not constant but changing due to various conditions. Drivers waiting for merging into a lane several times will be more impatient than drivers who were able to merge without problems. Rising impatience causes a more aggressive driver behavior. This concept of impatience is also responsible for lane changes that force strong reactions from surrounding traffic, since drivers can reach a level of impatience that justifies this behavior. One more influence that mainly affects free flow traffic is the possibility of speeding. For this reason there is a factor, multiplied with the speed limit, which varies from driver to driver and the distribution of these factors has to be calibrated as well.

For strategic lane changes the foresight distance is the determining factor, it defines which distance ahead a driver knows his route. This factor is normally constant but for a more realistic simulation the attention of the driver is taken into account, whereby the foresight distance varies from driver to driver. To realize this, a varying factor must be introduced and calibrated. A short foresight distance can result in very egoistic and potentially dangerous lane changes because the lane change must be executed quickly to follow the route. In this case the urgency is not rising continuously but jumping up at the time the driver notices the situation. The very special characteristic of cooperative lane changes in Chinese traffic makes it quite complicated to model them. The way this forced cooperation is recognized here is strongly linked to the sub lane model: The lateral orientation of vehicles within a lane is normally defined by a factor that either causes the vehicle to stay in the center or a particular side of a marked lane or to drive as compact as possible to maximize capacity. This factor is now used to model the forced cooperation. To do so the factor is set to maximize capacity when other vehicles try to force a lane change. This causes vehicles to drive parallel instead of causing strong breaking reactions to avoid a collision. After ten seconds, the factor is set back to its original value and the vehicles start merging into the original lanes again. For the calibration of obligatory lane changes the probability for obligatory lane changes is reduced and the time the right lane has to be empty before drivers consider an obligatory lane change is raised.

Clearing the way for fast follower vehicles is not currently represented in SUMO. It can be considered as a form of cooperative lane change that can be triggered based on speed difference between ego and follower vehicle in relation to the inconvenience of performing the lane change. This new behavior will have to be calibrated as well. Besides the lane change motivation the viability of the lane change has to be considered. Like most lane change models an integrated car following model is used here. The model allows the calibration of acceleration and deceleration limits and the adoption of the desired gap.

In SUMO these parameters are attributes of each vehicle. To meet the characteristic driver distribution in Chinese traffic which comprises novice drivers, normal drivers and experts in a representative distribution of parameters will be calibrated.

11.4 Conclusion and future work

The previous sections have shown that it is possible to combine a simulation of Chinese traffic with the possibility of testing ADAS. For this purpose, the calibrated simulation framework is used for prototypic testing of the autopilot functions in Chinese highway traffic. The design and implementation of the framework for a closed loop traffic-ADAS simulation is described in subsection 11.3.2. This framework enables us to simulate realistic interactions between surrounding vehicles and the ego vehicle, without predefining a specific scenario. The simulated drivers are calibrated on the basis of real world measurement data. Future work will focus on model validation as well as evaluation of the framework. Additionally the framework will be extended by a more detailed driver model with consideration of driver perception in future.

11.5 References

[1] BUSCH, FRITZ: *Spurbelastung und Häufigkeit von Spurwechseln auf einer dreispurigen BAB-Richtungsfahrbahn*. Strassenverkehrstechnik, (6), 1984.

[2] ERDMANN, JAKOB: *SUMO's Lane-Changing Model*. In *Modeling Mobility with Open Data*, pages 105–123. Springer, 2015.

[3] GIETELINK, OLAF, JEROEN PLOEG, BART DE SCHUTTER and MICHEL VERHAEGEN: *Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations*. Vehicle System Dynamics, 44(7):569–590, 2006.

[4] GIPPS, PETER G.: *A model for the structure of lane-changing decisions*. Transportation Research Part B: Methodological, 20(5):403–414, 1986.

[5] KESTING, ARNE: *Microscopic modeling of human and automated driving: Towards traffic-adaptive cruise control*. 2007.

[6] KESTING, ARNE, MARTIN TREIBER and DIRK HELBING: *Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity*. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 368(1928):4585–4605, 2010.

[7] KRAUSS, STEFAN: *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics*. PhD thesis, Universitat zu Koln, 01.01.1998.

[8] NAGEL, KAI and MICHAEL SCHRECKENBERG: *A cellular automaton model for freeway traffic*. Journal de physique I, 2(12):2221–2229, 1992.

[9] REIMANN, MICHAEL: *Simulationsmodelle im Verkehr*, 2007.

[10] SCHAKEL, WOUTER, VICTOR KNOOP and BART VAN AREM: *Integrated lane change model with relaxation and synchronization*. Transportation Research Record: Journal of the Transportation Research Board, (2316):47–57, 2012.

[11] SCHILLER, MANUEL, MARIUS DUPUIS, DANIEL KRAJZEWICZ, ANDREAS KERN and ALOIS KNOLL: *Multi-Resolution Traffic Simulation for Large Scale High Fidelity Evaluation of VANET Applications*. SUMO 2015 - Intermodel Simulation for Intermodel Transport, (28):37–50, 2015.

[12] SCHULDT, F., F. SAUST, B. LICHTE, M. MAURER and S. SCHOLZ: *Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen*. AAET2013-Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel, Braunschweig, 2013.

[13] SEMRAU, MARC: *Entwicklung eines Verfahrens zur Identifikation von ausgewählten Verkehrssituationen aus dem chinesischen Großstadtverkehr*. PhD thesis, Technische Universität Braunschweig, Braunschweig, 11.11.2014.

[14] ZOELLNER, HENDRIK and WOLFGANG HUGEMANN (editors): *Zur Problematik der Bremsreaktionszeit im Straßenverkehr*, 1998.

12 Improving Traffic Lights Management Sytems Using Information Available by VANET

Department of Computer Science and Software Engineering
Faculty of Sciences and Engineering
Pavillon Adrien-Pouliot, 1065, avenue de la Médecine
Université Laval Québec (Québec), Canada, G1V 0A6

francois.vaudrin.1@ulaval.ca   *Laurence.Capus@ift.ulaval.ca*

Abstract

Traffic has significant impacts on waiting time and CO2 emission. Construction or road widening is not always a viable alternative to reduce road congestion. An economical solution is to enhance the current traffic lights system. However, traffic lights management is not a simple problem because combinatorial complexity so that problem resolution become out of reach. We are working on a solution that would reduce computational complexity to dynamically control a network. We tested our solution on SUMO simulator, which allows us to show that we can get a significant improvement in the total waiting time. Although the amount of used information is reduced, one must be able to get the data update at every second because vehicles move quickly. For this, we look to use data from the emerging technology Vehicular's ad hoc networks (VANET). We believe it is possible to monitor real-time traffic network with VANET and wireless technology connected by the Internet. This approach is promising for the cities because it does not require the costly capture equipment installation. The purpose of this paper is to show how the solution can be implemented with VANET and how the SUMO simulator could be useful to experiment this issue.

Keywords: Traffic, control, simulation real-time, VANET, SUMO.

12.1 Introduction

Traffic has significant impacts on waiting time and CO2 emission. Construction or road widening is not always a viable alternative to reduce road congestion. This work is expensive and sometimes difficult to implement due to lack of space. In addition, it is often a temporary solution that simply displaced the problem elsewhere. An economical solution is to enhance the current traffic lights system (TLS). In contrary poor programming of TLS increases waiting times, the risk of accidents, aggressiveness, fuel consumption and pollution.

TLS management is not a simple problem because combinatorial complexity increases with the addition of junctions and vehicles so that problem resolution become out of reach. Most of the studies propose solutions obtained by simulation software with the aid of a set of data obtained on the field. The TLS is then programed using a simulation model. Although the proposed solutions be useful,

they are not suited to an environment which is constantly changing. Some solution been proposed are the installation of loop detectors that help to control traffic flow by extending the duration of traffic lights according to the arrival of vehicles. Such systems have limitations, because they are no longer effective in situations of congestion. Detection loops are sensors allowing counting numbers of vehicles that are approaching a junction. They are installed under the pavement and they have an electronic mechanism able to detect vehicles when passing over the loop (2) (3). These installations require costly investments for governments. To have an efficient dynamic and automatic TLS, traffic managers must find a valuable solution at every moment. They also have to make sure to get valuable real-time data to be able to intervene at the right moment.

We are working on a solution that would reduce computational complexity to dynamically control a network of several intersections. To achieve this, we have reduced the quantity of information to process. We believe it is advisable to use approximate data because the model is operating in an unpredictable changing environment. Although the amount of information is reduced, one must be able to get the data update at every second because the vehicles move quickly. For this, we look to use data from the emerging technology Vehicular's ad hoc networks (VANET). The purpose of this paper is to show the feasibility of our solution using VANET.

This paper is written as follows: Section 2 addresses the problems related to TLS management, section 3 on the challenge of collecting real-time data, section 4 of the proposed solution with VANET. We conclude with the main benefits of this proposal.

12.2 Dynamically managing TLS: a dual problem

Traffic issues have never been more important. A recent Canadian study found that annual costs of road congestion are more than 1.4 billion dollars in Montreal[1] and 7 billion in Toronto metropolitan area[2]. This is one of the most important problems in several cities around the world.

TLS are the principal means to control vehicles flow but there are a huge number of unpredictable events such as aggressive behavior, imprudent and distracted driving, human errors, weather conditions and so on. That makes this task infinitely complex to manage.

During a microscopic simulation, movements and interactions of vehicles are simulated by the model used by the simulator. A widely used simulation model is the car-following-model. It describes how a vehicle is following the next vehicle. This model defines the minimum distance between two vehicles (headway), how the vehicles react in acceleration and deceleration from the vehicle ahead. Aggressive behavior may also be considered in the model. In addition, the number of lanes and the maximum speed are parameters affecting the simulation. It is the combination of the results of all vehicles (waiting time, average speed, travel time, etc.) that permit to judge the effectiveness of the system (3). The goal is to find the best traffic lights sequences to reduce inconvenience to users. In the field of optimization it is about to minimize the objective cost function.

---

[1] Transport en commun - Au coeur du développement économique de Montréal,http://www.ccmm.qc.ca/documents/etudes/2010_2011/10_11_26_ccmm_etude-transport_fr.pdf,http://www.metrolinx.com/en/regionalplanning/costsofcongestion/ISP/_08-015_Cost_of_Congestion_report_1128081.pdf,Montreal, Consulted november 30, 2015.
[2] Canada's Ecofiscal Commission, HDR inc. (2008). Cost of Road Congestion in the Greater Toronto and Hamilton Area: Impact and Cost Benefit Analysis of the Metrolinx Draft, Chambre de commerce du Montréal métropolitain. (2010), Consulted november 30, 2015.

Simulation is usually done using commercial simulators by circulation experts in public administration. They use for this work historical data that have been measured on site during a sufficiency period of time. It is based on the results of the simulation that TLS is programed for several years. This task provides a snapshot of the reality at some point but this is far short of what is happening at every moment. It is reasonable to think that it is impossible to minimize an objective function at every moment. Despite remarkable improvements of communication means and the huge amount of data available, it does not exist in our knowledge effective systems that are able to manage large networks in real time. We must therefore look at other approaches to manage a TLS dynamically.

For smaller networks, a strategy to synchronize traffic lights is to induce vehicles to move in groups in order to reduce frequent stops and departures. This approach is recommended when distance between junctions is short (15). This approach is inapplicable in chaotic conditions or when the volume exceeds the network capacity. Two methods are typically used for optimization of TLS: graphic and through simulation (11). Traditionally, the graphic method was used, but it is now limited to simple cases.

But even by using the most efficient algorithms, it is possible that no solution can be found within a reasonable period of time because the problem may be too complex. This is a fundamental issue in computer science because some problems are still impossible to resolve. Moreover, the programming of TLS dynamically is a complicated task that needs time, effort, resources and technical skills (12). In addition, one must be able to provide an acceptable solution in a very short period of time. If the response is not fast enough, it may no longer be relevant because the situation has changed in the meantime (13). A method for controlling real-time TLS is to collect data from sensors. The next step is to process data by optimization algorithms and to modify the sequences of traffic signals accordingly. Even if this method is theoretically possible, it involves performing complex optimization calculations instantly. Also, it quickly reached its limits and the problem is becoming out of reach for large networks. Also, even if it was possible to find a quick solution at the right moment, it could be difficult to implement on the spot for safety reasons[3].
Finally, any real-time solution for monitoring a network requires data availability. However, governments are reluctant to adopt a new approach if it implies important investments to install equipment (optical fiber networks, cameras, movement detecting devices, etc.). Moreover, one must not neglect recurrent costs for operations and maintenance of this equipment.

So there are two key issues to consider controlling TLS dynamically. On one side, one must find an efficient solution instantly for a very complex optimization problem. On the other hand, it needs the required data in real-time in order to process the algorithm and implement the changes. Although the first issue is crucial, we believe that the study of the second has a major effect on the final result.

12.3 Experiment with SUMO simulator

Studying data to collect in real time amounts to asking what is the minimum information needed to be able to dynamically control TLS. We believe that the first step is to try to simplify the problem. Our works were inspired by the work of traffic officers in order to provide a common sense resolution model. We seek to imitate as much as possible the process of decision making of a traffic officer. This model allows us to maintain a balance between the major uncertainties related to the environment and data precision. In other words, a huge amount of data does not necessarily provide a better solution in an unpredictable environment.

---

[3]Institute of Transportation Engineers. Traffic Signal Timing Manuel, Washington, 2009.

We therefore looked for the minimum amount of information to process. We decided to use the occupancy rate of each road segment. This is a measure consistent with the variability of the environment. Our algorithm allocates more fairly waiting time on each street than a static system would do. Even if this system has some similarities with detection loops we believe that it offers more potential in order to recognize recurring traffic patterns, which could be studies and reused to manage future problems.

We have conducted a series of tests with the SUMO[4] simulator on a model with three intersections (see Figure 1). SUMO is a free and open traffic simulation suite which is available since 2001 and supported by the German Aerospace Center DLR[5] Institute of Transportation Systems. SUMO allows modeling of intermodal traffic systems including road vehicles, public transport and pedestrians. These works are the following of those explained at the SUMO conference 2015 (16).

The network is located in Québec City in Canada along one of the main roads linking to downtown. This network is made using free software OpenStreetMap (OSM[6]) and the Java application JOSM[7]. The arrival of the vehicles is random according to Poisson distribution.
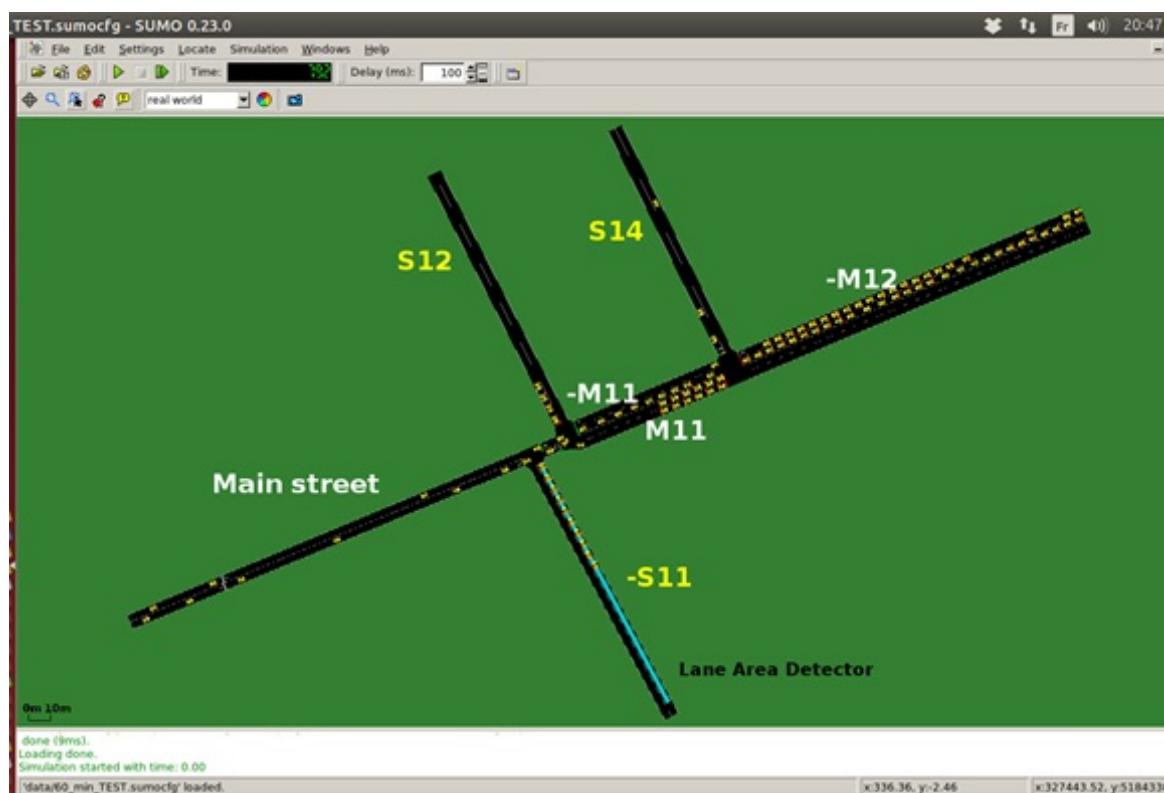

Figure 12.1: SUMO network

We aim to reduce the number of stop and go to create a green wave and promote grouping vehicular movement. The strategy is to modify or extend duration of green traffic lights on main road segments. We try to react like a traffic officer would do in real life by prioritizing vehicles traveling on main road. To test the algorithm, we conducted two simulations (static and dynamic). We have

---

[4]http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/, accessed March 25, 2015.

[5] http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-1221/1665_read-3070/, accessed April 10, 2016.

[6]OpenStreetMap (OSM) is a collaborative project to create a free editable map of the world, https://en.wikipedia.org/wiki/OpenStreetMap, accessed March 25, 2015.

[7]JOSM is an extensible editor for OpenStreetMap written in Java 7, https://josm.openstreetmap.de/, accessed March 25, 2015.

previously upgraded the static system to obtain best results. The measure of effectiveness is based on the total waiting time and maximum halting time. SUMO provides this information in an output file at the end of each simulation. For the adaptive system, we used the Tracy module also includes in SUMO that allows to intervene dynamically during the simulation.

Preliminary results get a significant improvement in reducing waiting time (see Figure 2). We also checked whether some vehicles are waiting too long a time. In both simulations maximum waiting time are comparable (not shown on Figure 2). However, in the static system, the maximum waiting time is on the main road while in dynamic simulation, it is on a secondary road. This is relevant because the system favors the road that has the highest amount of traffic.
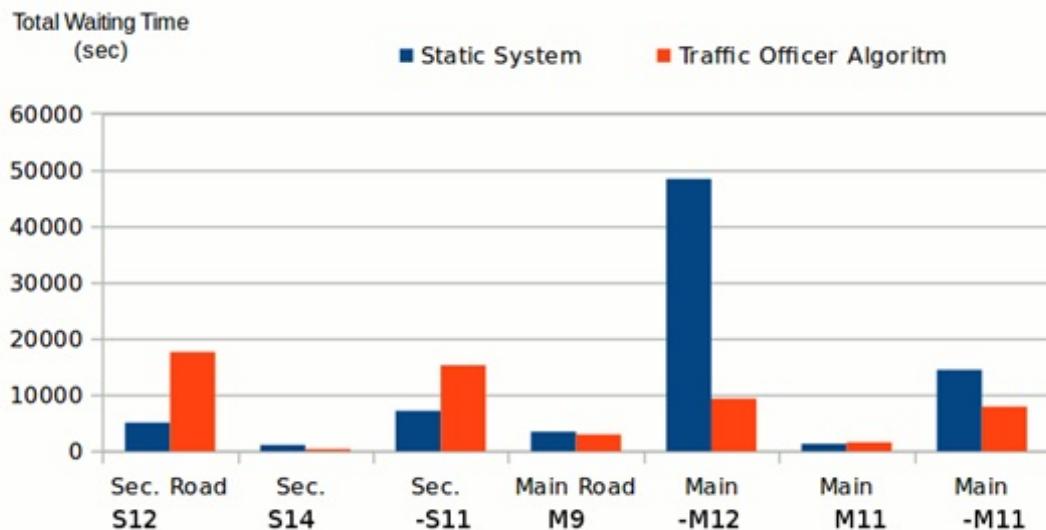

Figure 12.2: Total waiting time

The first tests were about the number of segments in pre-congestion and we plan to add criteria to improve performance. For example, we think that it is possible to add weighting factors on specific edges to improve the decision process. Although theoretically, our system can improve traffic flow, our goal is to establish technical feasibility for managing TLS network dynamically in real life. It is therefore to examine how to obtain traffic data in real time.

12.4 Collecting real time data

Some data are available relate to road occupation through Internet or vehicle navigation system (VNS) as Google Traffic or TomTom. But the frequency of updates is too long (2-5 minutes) (14). A further issue is related to the accuracy of the data provided by these systems. To our knowledge, these information systems provide a global average level of traffic but do not give the occupancy rate on each segment. In addition, there is a risk that public administrations become dependent on the information provided by third parties. And even though the information was graciously made available to cities that would require that each city take an agreement with these suppliers and it has to consider legal issues that would result.

However, there is an emerging technology which is subject to a lot of research in the world. It is the Vehicular's ad hoc Networks (VANET) defined by the IEEE 802.11p standard. In VANET every vehicle acts as a node of a temporary network. Each vehicle can receive information from other

vehicles every split second (0.1 sec) regarding their status in a range of about 300 meters. It also can send information to other vehicles. The information that circulates in the network are about speed, acceleration, distance between each vehicle, the braking system and so on. This information can also be transmitted to communication stations installed along the road - Road Side Unit (RSU) (6) and relayed to a Base Station and to a traffic control center (see Figure 3). This is called a communication mode from vehicle to vehicle (V2V) and Vehicle to Infrastructure (V2I).

VANET operates on a dedicated frequency spectrum (75 MHz bandwidth and 5.9 GHz allocated by governmental authorities). Furthermore VANET can operate in variable weather conditions (rain, fog, snow) and confidentiality of information is maintained.
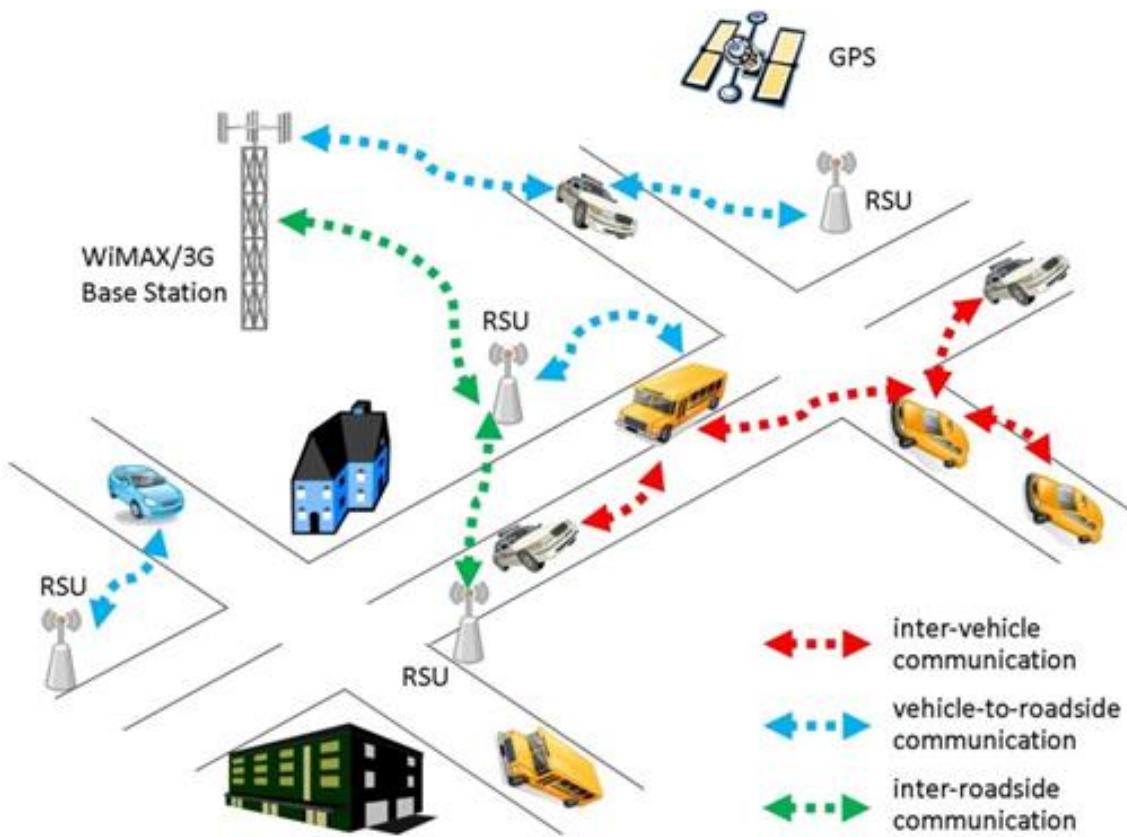

Figure 12.3: VANET from (8)

The main reason why this system was introduced is to improve safety (5). It also aims to improve circulation for example by providing information about alternate paths. It can also be used for commercial applications or to transmit messages to users (8). Car manufacturers install increasingly these devices in new vehicles and it is reasonable to assume that in a few years most vehicles will be linked by VANET. However, there is a delay in the RSU installation. These works require investments from the governmental authorities concerned.

Moreover, much research touch safety and development of algorithms to facilitate travel. However, there is little research about using VANET to manage circulation systems (14) (9). It is what we are proposing to do. We aim to use this technology in order to collect data in real time and to be able to control a TLS dynamically.

The data that interests us relate to the density of traffic on each road segment. In our model, we use the distance between each vehicle (headway) to calculate the occupancy rate (Figure 4).
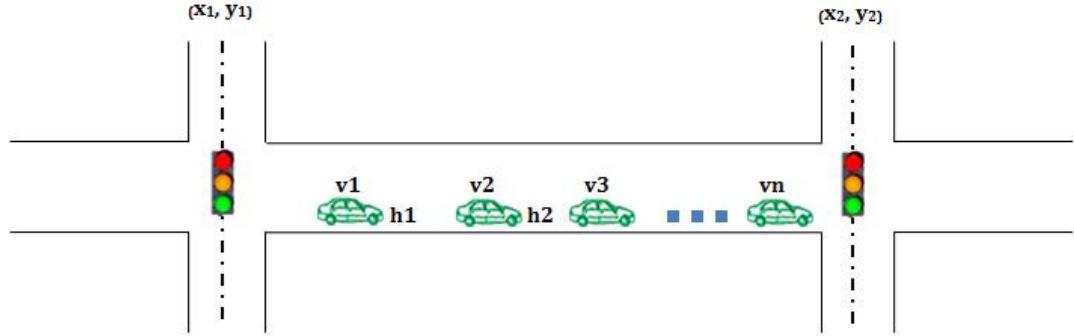
Figure 12.4: Headways

The localization of each junction is known using Open Street Map (OSM) or another source. The length of the segment can be calculated by the Euclidean distance. The summation of headway is used to calculate the available space on the road and by adding the distance between the first junction and the first vehicle, and the distance between the last car and the last junction.

This is to make the same calculations for each segment and to use the results as in our model. For example, when the number of problematic segments exceeds a critical threshold, the system instructs to modify the sequences of traffic lights. The data transmission takes place instantaneously with VANET and it is relayed to the circulation control center by an Internet network dedicated for better performance.

$L = Edge\ length\ (between\ two\ junctions)$
$(x_1, y_1) = coordinates\ of\ first\ junction$
$(x_2, y_2) = coordinates\ of\ second\ junction$

$$L = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \tag{12.1}$$

$h = headway(distance\ between\ each\ vehicle)$
$n = vehicles\ in\ Edge$
$S_h = free\ space\ headways\ Edge$

$$S_h = \sum_{i=1}^{n} h_i \tag{12.2}$$

$V_{1x}, V_{1y} = localisation\ of\ first\ vehicle$
$S_{1-} = free\ space\ between\ the\ first\ junction\ and\ the\ first\ vehicle$

$$S_{1-} = \sqrt{(V_{1y} - y_1)^2 + (V_{1x} - x_1)^2} \tag{12.3}$$

$V_{nx}, V_{ny} = localisation\ of\ last\ vehicle$
$S_{n+} = free\ space\ between\ the\ second\ junction\ and\ the\ last\ vehicle$

$$S_{n+} = \sqrt{(y_2 - V_{ny})^2 + (x_2 - V_{nx})^2} \tag{12.4}$$

$$S_{free} = total\ free\ space\ in\ Edge$$

$$S_{free} = S_h + S_{1-} + S_{n+} \tag{12.5}$$

$$Occupation_{rate} = \frac{L - S_{free}}{L} \tag{12.6}$$

12.5 Implementation and benefits using SUMO

As mentioned earlier, we cannot use data obtained from satellite navigational systems due in particular to time delay problems. VANET, however, is fast enough to provide such data instantly. The data can then be relayed to a traffic control center via the RSU and the base station. Knowing this information it will be possible to intervene immediately to reduce traffic congestion. System architecture and data flows are shown in Figure 5.
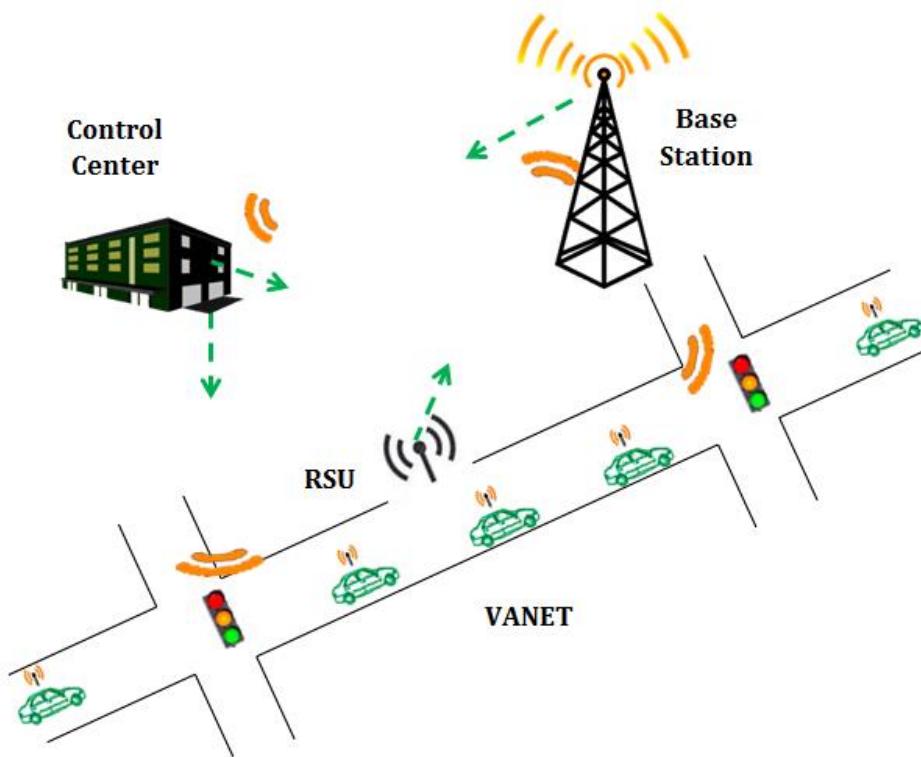
Figure 12.5: TLS with VANET in Real-Time

The system installed in the control center received data and analyze them. Decision making can be done automatically by implementing the model that we used for our simulation. This step can be done very quickly, because the calculations are minimized. It is also possible to develop other algorithms. Thereafter, the instruction of change is transmitted to the TLS. Transmitting instructions to traffic lights involves linking the control center to the TLS. The connection is done via the Internet and an electronic device installed on the post traffic.

There is also an emerging technology that aims interconnecting the infrastructure to the Internet - the Internet of Things (IoT) - and to make cities more efficient (Smart Cities) (10). A pilot

project was conducted in Italy to validate the feasibility of this technology (17). They sought to check the temperature, humidity, sunshine and the amount of benzene in the air of the city of Padova. Sensors devices have been installed on streetlights which relayed information to the control management system of the city. It has been demonstrated that reliable information could be transmitted in real time through the Internet. Similar equipment could be installed on traffic post.

The main advantages of using VANET are that data moves quickly and that it is free. Appliances and energy to power the network comes from each vehicle. There is no expensive equipment to be installed to collect data except RSU terminals and other small devices.

The drawbacks are related to the loss of signals for various reasons, including the presence of buildings (4). There is also a risk of malicious data attacks (9). In addition, there are few RSU installed in Canada (and elsewhere), so there is a need for investment from governments (1).

Furthermore, this concept has some similarities with the COLOMBO project (7). This project also proposes using data from VANET but it is in a distributed system based on swarm intelligence and self-organizing adaptive TLS. The instructions are transmitted directly to each intersection without using a centralized control center. In our approach, we also propose to collect data in real time with VANET but we transmit them to a control center via the RSU stations and telecommunication towers. After that, this information is processed in real time through various algorithms. The goal is to collect data in real time and we believe that the judicious use of VANET can solve this key aspect.

With the benefits that VANET could bring to the well-being of road users, it is likely that the framework has the potential to continue to grow in the coming years. Thus, a complete traffic lights management system using VANET could emerge as for VNS.

The benefits of using SUMO are many. SUMO simulator permits to test various approaches and make change until satisfactory results. It also helps to understand and regulate traffic flow, to identify specific problems and reuse solutions that worked well. Moreover, SUMO is a free open-source software, it is easy to use and the learning curve is reasonable. Additionally, most open-source projects rely on the goodwill of those participating and SUMO provides continuous support by experts working for the German Aerospace Center DLR. These experts are experienced researchers in the area of transport and they are also IT experts. Furthermore, SUMO permits dynamic simulations with Tracy module and to simulate VANET's communications with ns-3. SUMO also uses OpenStreetMap that allows to develop a network easily and to transfer it to the simulator via the Netconvert module included in SUMO. It uses Python programming and provides an easily modifiable template to link with the dynamic module Tracy. In contrast, commercial softwares are expensive even for academic versions and sometimes you have to make your own program in C++ to intervene dynamically during a simulation. SUMO allows a large number of researchers worldwide to do research and we think that would be impossible to achieve if they had to rely on commercial software.

Conclusion

The purpose of this article was to show that VANET technology can be used to control traffic lights systems in real time. We have also seen that it is now possible to control remotely equipment installed on infrastructure in cities with the assistance of wireless devices and through the Internet. These technological advances give hope that it would be feasible within a few years to monitor a network without having to install costly sensors devices. To our knowledge, this is a new approach which seems promising for public administrations.

Finally managing a large network is a problem that seems to us too complicated to be resolved by conventional methods. As part of our work, we suggest using a more simplified approach and to process a minimum of data. Future works are to demonstrate the proposed approach by simulating multiple junctions including data transmission by VANET.

Acknowledgements

12.6 References

[1] BALDESSARI, ROBERTO, ANDREAS FESTAG and JULIEN ABEILLÉ: *NEMO meets VANET: A Deployability Analysis of Network Mobility in Vehicular Communication*. In *Proceedings of 7th International Conference on ITS Telecommunications*, pages 375–380, Sophia Antipolis, France, 2007. June.

[2] BELLEMANS, TOM, BART DE SCHUTTER and BART DE MOOR: *Models for traffic control*. Delft University of Technology, Delft, The Netherlands, 2002.

[3] BRADLEY, W and P E YARGER: *Fully Actuated vs. Semi-Actuated Traffic Signal Systems*. Technical Report, Yarger Engineering Inc., Indiana, 1993.

[4] CODECA, LARA, RAPHAEL FRANK and THOMAS ENGEL: *Luxembourg SUMO Traffic (LUST) Scenario: 24 Hours of Mobility for Vehicular Networking Research*. In *2015 IEEE Vehicular Networking Conference - VNC*, Kyoto, Japan, December 2015.

[5] DEPARTMENT OF TRANSPORT US (DOT): *Dedicated Short-Range Communications (DSRC)*. www.its.dot.gov, accessed February the 29, 2016.

[6] JALIL, PIRAN MOHAMMAD, G. RAMA MURTHY and G. PRAVEEN BABU: *Vehicular ad Hoc and Sensor Networks; Principles and Challenges*. International Journal of Ad Hoc, Sensor and Ubiquitous Computing (IJASUC), 2, June 2011.

[7] KRAJZEWICZ, DANIEL and JAKOB ERDMANN: *Road intersection model in SUMO*. In *1st SUMO User Conference-SUMO 2013*, volume 21, pages 212–220. DLR, 2013.

[8] KU, IAN, YOU LU, MARIO GERIA, FRANCESCO ONGANO, RAFAEL L GOMES and EDUARDO CERQUEIRA: *Towards Software-Defined VANET: Architecture and Services*. In *Ad Hoc Networking Workshop*, 13th Annual Mediterranean IEEE. IEEE Xplore Digital Library, June 2014.

[9] KUMAR, BHOI SOURAV and PABITRA MOHAN KHILAR: *Vehicular communication: a survey*. Published in IET Networks, August 2013.

[10] MISCHA, DOHIER, CARLO RATTI, JURIJ PARASZCZAK and GORDON FALCONER: *Smart Cities*. IEEE Communications Magazine, pages 70–71, June 2013.

[11] ROESS, ROGER P, WILLIAM R MCSHANE and ELENA S PRASS: *Traffic Engineering*. Prentice Hall, University of Idaho, 2nd edition, 1998.

[12] SADEK, ADEL W, GARY SPRING and BRIAN L SMITH: *Toward More Effective Transportation Applications of Computationnal Intelligence Paradigms*. In *TRB Annual Meeting*, page 4, 2003.

[13] SHAHADAT, HOSSAIN and LINA KATTAN: *Responsive Signal Control for Rush Hour Traffic in an Arterial*. In *WCTR*, page 3, Lisbonne, Portugal, 2010.

[14] SHEN, WANG, LERO RINCE, SOUFIENE DJAHEL and JENNIFER MCMANIS: *An Adaptive and VANET s-based Next Road Re-routing System for Unexpected Urban Traffic Congestion Avoidance*. In *IEEE Vehicular Networking Conference (VNC)*, Kyoto, Japan, December 2015.

[15] TETTAMANTI, TAMAS and ISTVAN VARGA: *Distributed traffic control system based on model prodictive control*. Periodica Polytechnica, 2009.

[16] VAUDRIN, F. and L. CAPUS: *Experiment of a common sense based-approah to improve coordination of Traffic Signal Timing System with SUMO*. In *Proceedings of the SUMO 2015: Intermodal Simulation for Intermodal Transport*, volume 18, pages 127–135. DLR, Berlin-Adlershof, German Aerospace Center, Institude of Transportation Systems, May 2015.

[17] ZANELLA, ANDREA, NICOLA BUI, ANGELO CASTELLANI, LORENZO VANGELISTA and MICHELE ZORZI: *Internet of Things for Smart Cities*. IEEE Internet of Things Journal, 1, February 2014.

13 JuPedSim: an open framework for simulating and analyzing the dynamics of pedestrians

Mohcine Chraibi and Jun Zhang; Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH
{m.chraibi, ju.zhang}@fz-juelich.de

Abstract

There is a wide range of simulation software used to investigate pedestrian dynamics. It is paramount for researchers to know exactly what the models are performing as this provides a better interpretation of the results. In this paper we introduce *JuPedSim*, the Jülich Pedestrian Simulator. JuPedSim is an extensible framework for simulating and analyzing pedestrians' motion at a microscopic level. It consists at the moment of four modules which are loosely coupled and can be used independently. The first module simulates the movement of pedestrians given a geometry and an initial configuration. The second module visualizes the geometry and the trajectories. High resolution videos can also be recorded directly from the module interface. The third module analyses the results from the simulation or any other source, for instance experiments, and generates different type of plots such as densities, velocities and flows. The last module is a geometry editor to prepare the needed environment and input files for the simulation and the analysis.

Keywords: microscopic models, simulation, analysis, software, open-source, pedestrians.

13.1 Introduction

The growing complexity of modern buildings and pedestrians' outdoor facilities make the use of simulation software inescapable. Simulations are used not only in the design phase of new structures but also later in the preparation or monitoring of large-scale events. To obtain reliable results, it is paramount to understand and reproduce the underlying phenomena which rule the dynamics of pedestrians. There exist numerous software tools, mostly commercial (or with proprietary licenses), for simulating pedestrians (1; 5; 9; 16; 12). They usually implement a single model and are only limitedly suitable for academic purposes, where the aim is generally a rapid prototyping framework for implementing and testing new concepts or models. There exist open source tools as well (11). However, they are mainly designed for a specific application and do not offer easy access for model testing or extension with new features.

In addition the state of the art in the modeling of pedestrians is characterized by a high dynamic and the zoo of models is still growing. However, it lacks a common basis for model comparison and benchmarking. From the scientific and academic point of view, it is often crucial to understand how a model has been implemented since the mathematical description and the computer implementation may differ. This also raises issues about the validation of the models, especially against empirical data, which is often neglected in many software.

Another motivation behind a versatile framework for models is the fact that in many universities and thesis projects, an important amount of time is spent on setting up a proper environment for the development of new ideas. Although the main objective is, for instance the development of a new pedestrian model, the work on utilities like an editor for the geometry or for the data entry, a tool for visualizing the results is an enormous entry barrier. Another side issue is the definition of file formats. Only after those steps have been successfully taken, the real work starts. Most of the time, the development of well-designed software is not in the focus of research related work, which results in code which is neither reusable, nor maintainable, nor scalable. The primary goal of JuPedsim is to provide students and researchers with a framework that will let them focus on their main task, i.e. development, calibration, and validation of new models. JuPedSim is currently focusing on evacuation, but will be extended to cover other areas as well.

When it comes to model validation and calibration using empirical data, a key aspect is the quality of the underlying data and the measurement methods. Recently several laboratory experiments as well as field studies have been performed to collect empirical data for model calibration by different research groups. However, there is still no common view about the type of experiments and their relevance for addressing specific questions. Also the ultimate step of extracting trajectories (positions over time) from video footages is cumbersome and error prone. Many empirical data have only been analyzed roughly and macroscopically and detailed information were not extracted due to a lack of technical background. The camera settings must been known exactly as minor errors already lead to huge discrepancies in the trajectories (3). In addition different measurement methods are often used to extract the same quantities and relationships, for instance on the fundamental diagram, and they have a huge impact on the results (19). Most importantly, for model calibration the empirical data and simulation results should be based on the same measurement method. By providing a unified framework for analysing data, it is ensured that all comparisons are at least based on the same measurement method and implementation. There only exist a few tools which integrate more than one pedestrian model.

The second section of this article gives an overview over the architecture of the framework. The third section focuses on the validation steps for the framework. The article is concluded in the fourth section with discussion and planned features.

13.2 Framework

The architecture of the framework is presented in Figure 14.1. It consists at the moment of three loosely coupled modules: a simulation module, a visualisation module and a reporting module. JuPedSim implements state of the art models and analysis methods. Two models at the operational level and three models at the tactical level are actually implemented in the framework. Further models can be incorporated by third parties without much efforts. The reporting tool integrates four measurement methods. Possible analysis include densities, velocities, flows and profiles of variables in given geometry. In contrast to other simulation packages, an emphasis is set on the validation of the implemented models. All inputs and output files are XML based. JuPedSim is platform independent, released under the LGPL license and written in C++.

13.2.1 JPScore

The module JPScore simulates the movement of pedestrians given a geometry and an initial configuration. The initial configuration includes the desired destinations, speeds, route choices and other demographic parameters about pedestrians. The simulation modules implemented follow the strategic/tactical/operation levels paradigm as described in (8). Three models at the tactical level (route choice, short term decisions) are already implemented in the framework: a shortest path strategy using the Dijkstra algorithm, a quickest path based on visibility and jam avoidance (10)
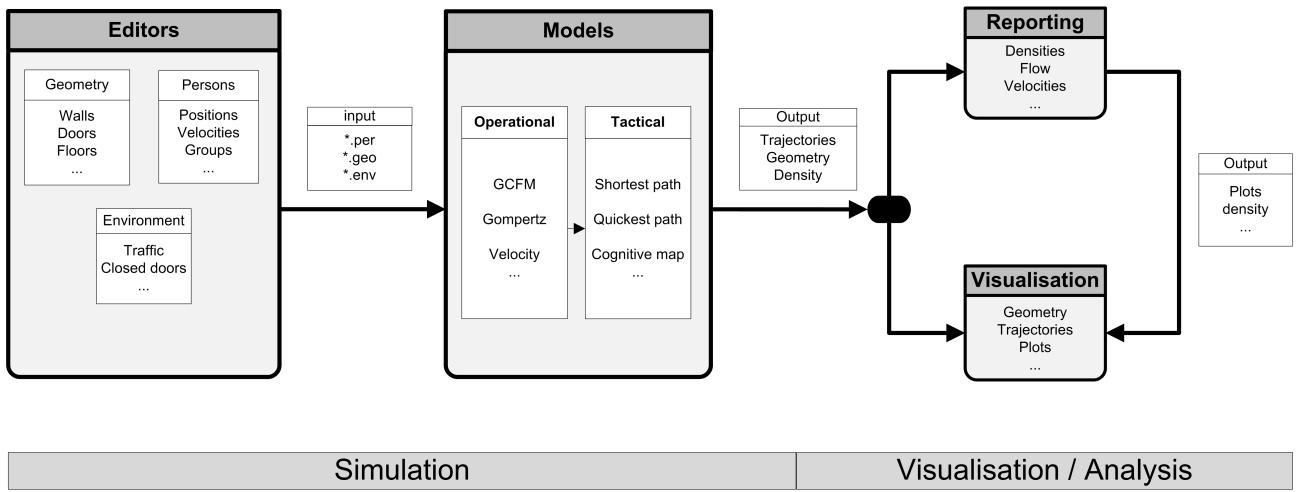
Figure 13.1: Architecture of JuPedSim built around independent modules. The simulation models, reporting and visualiation are fully functional. The editor and converter for the geometry are in active development.

and a cognitive map, giving agents the possibility to explore the environment and discovers doors for instance (7). Other models are in the process of being integrated. In addition some behavioral features are implemented such as the possibility to share information about closed doors with other agents and the ability to explore an unknown environment looking for an exit. On the operational level (locomotion system, collision avoidance) JuPedSim comes with different force-based and velocity-based models:

- Generalized Centrifugal Force Model (GCFM) (4).

- The collision-free velocity-based model (2).

- The Wall-avoidance model (6).

13.2.2 JPSreport

The reporting module $JPSreport$ analyses the trajectories from simulations or any other sources for instance experiments and generates different results. The module integrates four measurement methods. Possible analysis include densities, velocities, flows and profiles of pedestrians in a given geometry. Figure 13.2 shows the architecture of $JPSreport$.

Its input is composed of three parts: trajectory, geometry and a configuration file. The trajectory file from simulation or experiment contains the person id, time and the corresponding coordinates while in the configuration file the parameters for each method are given, for example the specification of the measurement areas. With these information the corresponding quantities are calculated and saved in different files. Figure 13.3 shows a sample work flow for analyzing trajectories: performing experiments, extracting the trajectories, analysing the desired quantity, in this case density and velocity. The densities and velocities can be calculated entirely for a selected area or for a unique pedestrian in the selected area. The time series of density, velocity as well as the Voronoi cell of each pedestrian can be visualized directly during the calculation enabling a live check of the results. From the profiles, the spatiotemporal distribution of density and velocity can be identified easily which in turn can be used to detect transition regions in a geometry and provide an optimal design for facilities. More information about the measurement algorithms are given in (17).
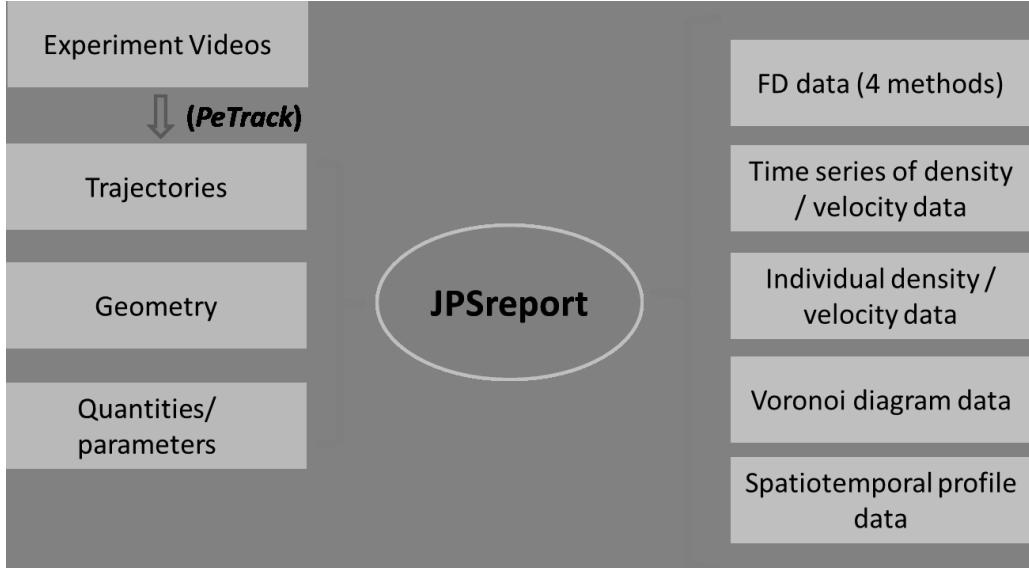
Figure 13.2: Architecture of the reporting module $JPSreport$.
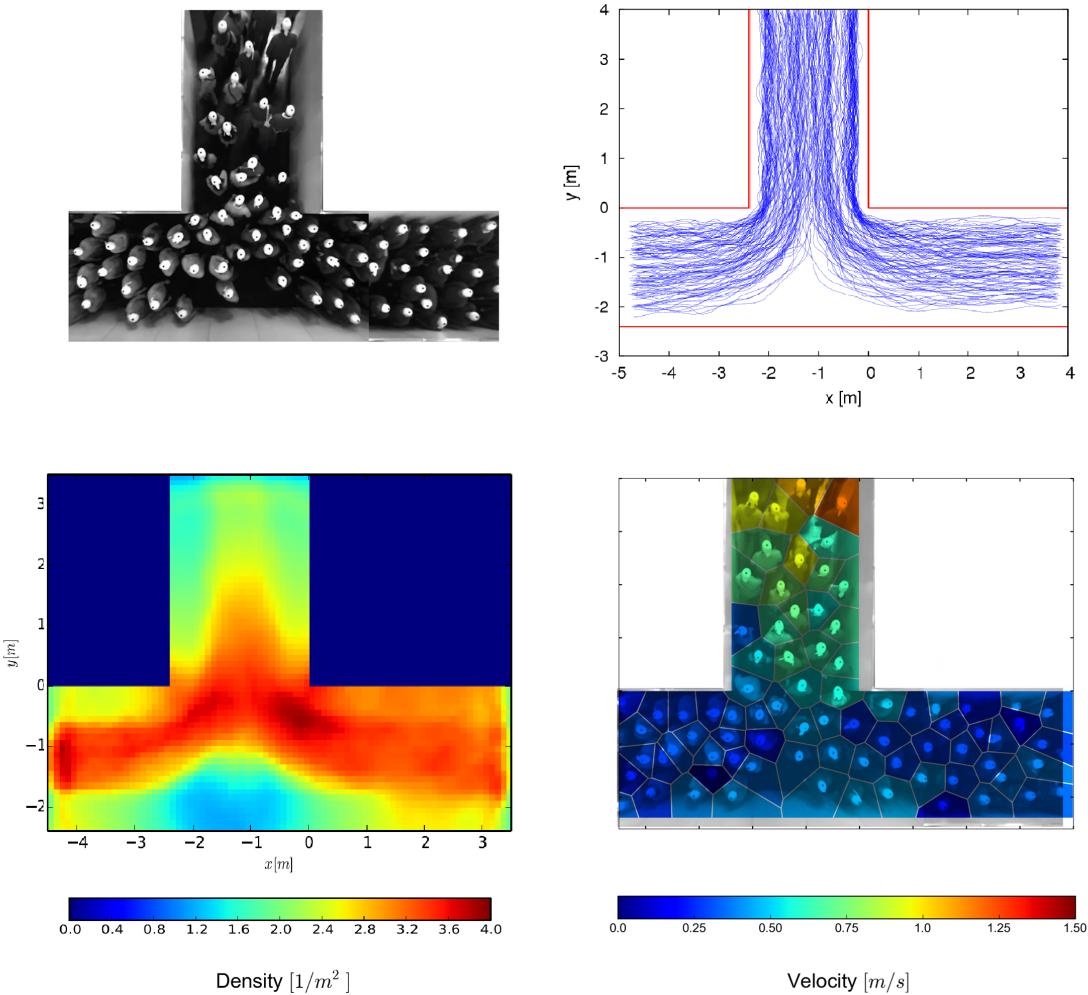
Figure 13.3: Workflow for analyzing trajectories using $JPSreport$. Top left: experiment involving two pedestrian streams coming from left and right and merging in a T-Junction. Top right: automatically extracted trajectories using PeTrack (3). Bottom left: density profiles computed with JPSreport using the Voronoi method (18). Bottom right: instantaneous velocities of the pedestrians using the Voronoi method.

13.2.3 JPSvis

The third module $JPSvis$ reads a file containing the simulation results (coordinates, velocities, orientations, . . .) together with geometry information and allows the user to interact with this information in form of an animation, for instance focusing on an area of interest or masking views. $JPSvis$ can also be used in an online mode, where simulation results are directly streamed to the application. This is a practical feature especially during the process of developing/testing new models or when simulating large scenarios, where results need to be visualised online. High resolution videos can be recorded directly from the module interface as well. 13.4(b) shows a simulation of a three-floor building. In the 3D perspective pedestrians are visualised with cylinders (in the model they are presented as ellipses). The colors of pedestrians are correlated with their instantaneous speed. The 2D perspective of a larger scenario is shown in 13.4(a). It features the evacuation of a stadium with an initial configuration of many thousand pedestrians distributed in the different sections. The red spots denote the areas with a low velocities. They are mainly located in the grand stand and are due to physical constraints imposed by the walls and seats. The situation is different in the promenade, where there is more free space. Small jams are observed directly at the exits.
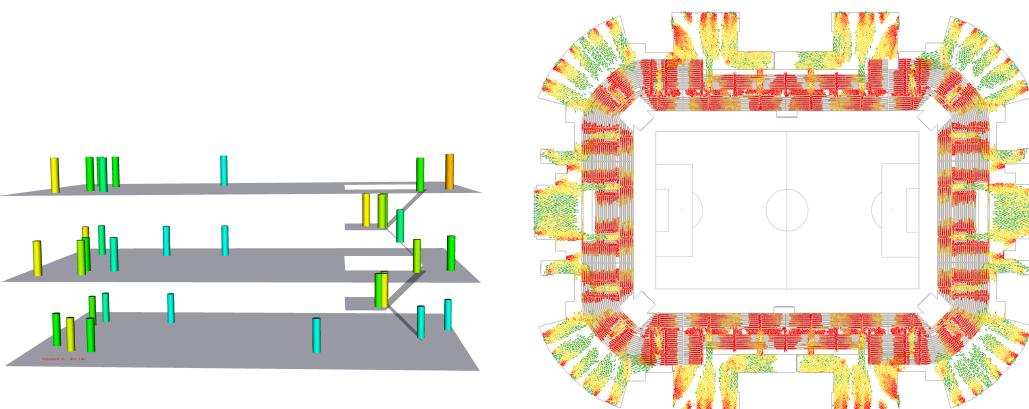
(a) Screenshot of the simulation of a building evacuation with three floors connected with two staircases.

(b) Screenshot of the simulation of a soccer stadium. Red marked pedestrians have a lower velocity compared to green ones, who move at their desired speed (ca. 1.34m/s)

13.3 Verification and validation

Validating and testing the reliability of models are an integral part of the development cycle of any software. Within JuPedSim new standards and benchmark scenarios for evaluation of pedestrian simulations are developed. The models implemented are validated using a benchmark suit, which is a compound of verification and validation tests. The verification tests are mostly based on the RiMEA test cases (14). RiMEA is a German based independent organization which aims at providing directives for microscopic crowd evacuation analysis. Its members include universities, research institutions and industries. Other verification tests e.g. (15) and the IMO validation tests (13) are strongly based on the RiMEA tests.

The aforementioned verification standards consider common sense tests that ensure reasonable simulation e.g. agents do not go though walls, or agents cannot find their ways in a building to outside. However, the realism of the simulations in comparison with empirical findings is not guaranteed. For JuPedSim we base the validation process on 7 test scenarios that are issued from published empirical data gathered from experiments under laboratory conditions:

- Test 1 – 1D unidirectional flow in corridor with periodical boundary

- Test 2 – 2D unidirectional flow in corridor with periodical boundary

- Test 3 – Unidirectional flow in corridor with open boundary

- Test 4 – Unidirectional flow around a Corner

- Test 5 – Flow through bottleneck

- Test 6 – Merging flow in T-junction

- Test 7 – Bidirectional flow in corridor

The empirical data and the results are published in (17) and they are available online on our central repository[1]. Further verification tests necessary to benchmark certain specific elements of the software for example the parallel architecture are implemented. In order to ensure the quality of the framework and guarantee its functionality, JuPedSim is tested on a daily basis against the afore mentioned tests.

13.4 Conclusion

This article presented JuPedSim, a framework for rapid prototyping new pedestrians models. The framework also includes state of the art measurement methods. All information including documentation, source code and experimental data are available at `www.jupedsim.org`. JuPedSim has been used for instance in a real time evacuation assistant for arenas (10). Planned features for the framework include a graphical user interface for editing the geometry, which will also include import capabilities for various CAD formats and the connection of the pedestrian simulation with a fire simulator. In addition a benchmark suit for verification and validation of pedestrian dynamics software is under development and will be released in the future.

13.5 References

[1] *Simulex User Guide, Virtual Environment 5.9.* www.ies4d.com.

[2] ANTOINE TORDEUX, MOHCINE CHRAIBI and ARMIN SEYFRIED: *Collision-free speed model for pedestrian dynamics*. Delft, Neederlands, 2015. TGF'15.

[3] BOLTES, MAIK: *Automatische Erfassung präziser Trajektorien in Personenströmen hoher Dichte*. Dr., Universität zu Köln, Jülich, 2015. Universität zu Köln, Diss., 2014.

[4] CHRAIBI, MOHCINE, ARMIN SEYFRIED and ANDREAS SCHADSCHNEIDER: *Generalized centrifugal force model for pedestrian dynamics*. Physical Review E, 82:046111, 2010.

[5] GALEA, E. R., S. GWYNNE, P. J. LAWRENCE, L. FILIPPIDIS, D. BLACKSPIELDS and D. COONEY: *buildingEXODUS V 4.0 - User Guide and Technical Manual*, 2004. www.fseg.gre.ac.uk.

[6] GRAF, ARNE: *Automated Routing in Pedestrian Dynamics*. Master's thesis, Fachhochschule Aachen, 2015.

[7] HAENSEL, DAVID: *A knowledge-based routing framework for pedestrian dynamics simulation*. Master's thesis, Technische Universitaet Dresden, November 2014.

---

[1]www.asim.uni-wuppertal.de/datenbank

[8] Hoogendoorn, S. P., P.H.L. Bovy and W. Daamen: *Microscopic pedestrian wayfinding and dynamics modelling*. In Schreckenberg, M. and S.D. Sharma (editors): *Pedestrian and Evacuation Dynamics*, pages 123–155, 2002.

[9] I.S.T. Integrierte Sicherheits-Technik GmbH: *Benutzerhandbuch Aseri, Version 4.6*.

[10] Kemloh Wagoum, Armel Ulrich, Mohcine Chraibi, Jonas Mehlich, Armin Seyfried and Andreas Schadschneider: *Efficient and validated simulation of crowds for an evacuation assistant*. Computer Animation and Virtual Worlds, 23(1):3–15, 2012.

[11] Korhonen, Timo and Simo Hostikka: *Fire Dynamics Simulator with Evacuation: FDS+Evac*. Technical Report, VTT Technical Research Centre of Finland, 2009.

[12] Kretz, T., Stefan Hengst and Peter Vortisch: *Pedestrian Flow at Bottlenecks - Validation and Calibration of Vissim's Social Force Model of Pedestrian Traffic and its Empirical Foundations*. In Sarvi, M (editor): *International Symposium of Transport Simulation 2008 (ISTS08)*. Monash University, Melbourne, Australia, 2008.

[13] Organization, International Maritime: *Guidelines for Evacuation Analysis for New and Existing Passenger Ships*, oct 2007.

[14] *RIMEA-Richtlinie für Mikroskopische Entfluchtungs-Analysen*, 2007. www.rimea.de.

[15] Ronchi, Enrico, Erica D. KuKuligowski, Paul A. Reneck, Richard D. Peacock and Daniel Nilsson: *The process of verification and validation of building fire evacuation models*. Technical Report, National Institute of Standards and Technology, November 2013.

[16] TraffGo HT GmbH: *Handbuch PedGo 2, PedGo Editor 2*, 2005. www.evacuation-simulation.com.

[17] Zhang, J., W. Klingsch, A. Schadschneider and A. Seyfried: *Transitions in pedestrian fundamental diagrams of straight corridors and T-junctions*. Journal of Statistical Mechanics: Theory and Experiment, 6:06004, june 2011.

[18] Zhang, Jun, Wolfram Klingsch, Tobias Rupprecht, Andreas Schadschneider and Armin Seyfried: *Empirical study of turning and merging of pedestrian streams in T-junction*. In *Fourth International Symposium on Agent-Based Modeling and Simulation (ABModSim-4)*, Vienna, Austria, 2012.

[19] Zhang, Jun, Wolfram Klingsch and Armin Seyfried: *High precision analysis of unidirectional pedestrian flow within the Hermes Project*. Guangzhou, China, December 2010.

14 PARCOURS: A SUMO-Integrated 3D Driving Simulator for Behavioral Studies

Johannes Eising[1], Felix Fischer[1], Jan Flaig[1], Olaf Florian[1], Christoph Fritze[1], Karmel Indych[1], Markus Montenegro[1], Sebastian Müller[1], Markus Schneider[1], Nicole Vieregg[1], Andreas Wegge[1], Holger Döbler[†1], Florian Koller[2], Olaf Menzel[1], Stefan Dietzel[‡1], Björn Scheuermann[1],

[1]Department of Computer Science, Humboldt-Universität zu Berlin,
[2]Department of Psychology, Humboldt-Universität zu Berlin,
{ [†]holger.doebler, [‡]stefan.dietzel}@informatik.hu-berlin.de (corresponding authors)

Abstract

In behavioral studies, driving simulators are established means to investigate navigation systems and route choice behavior. Most driving simulators, however, lack mechanisms to simulate realistic traffic demand in large scenarios, thereby limiting scientific validity. With PARCOURS, we integrate a 3D virtual reality driving simulation with SUMO's realistic traffic demand models. Our integration is performed in a running SUMO simulation: participant-controlled vehicles affect and are affected by SUMO vehicles. PARCOURS translates movement to the associated SUMO cars, which move through LuST, Luxembourg traffic scenario. Our driving simulation renderer is based on the Unity engine, which provides high realism and scalability; vehicles are controlled using gaming wheels, and Oculus Rift headsets can be worn to enhance driver immersion. PARCOURS, thereby, enables more representative measurements of behavioral and other psychological artifacts. We expect such experiments to contribute to more realistic models for route choice strategies, better driving predictions, and improved microscopic traffic simulations.

Keywords: driving simulation, traffic simulation, behavioral studies, SUMO, LuST scenario, virtual reality

14.1 Introduction

Advanced intelligent urban personal transportation systems that aim at reducing travel costs, e. g, time, fuel or emissions, by coordinating vehicles form a major research trend that receives increasing attention from academia and industry alike. For example, Youn, Gastner, and Jeong (29) showed that uncoordinated drivers waste a significant amount of their travel times within real-world road networks. Lujak, Giordani, and Ossowski (18) proposed an optimization model with a specific route assignment of paths to vehicles. With their model, it is possible to optimize mean travel times, but for some individual journeys the result could be unfair. A route guidance system that is designed to achieve system-optimal travel times and that simultaneously guarantees fairness based on route assignment was already introduced by Jahn, Möhring, Schulz, and Stier-Moses (15). In most cases, the focus has exclusively been upon the algorithmic perspective, i. e., upon computer

science. However, it is imperative to understand how humans adapt and change their travel behavior using such route guidance systems on particular journeys.

Behavioral studies are an integral part of transportation research. Such studies include research from psychology and human geography. For instance, psychological examinations focus upon "human behavior and well-being in conjunction with the regularized movement of people, goods, and services regarding work, personal life, and community activity systems" (25). The examinations are conducted to assess how humans' motional, cognitive, and personality factors, as well as environmental constraints, affect their attitudes, performance, decisions, and experiences, which can be used to infer requirements for optimal system development.

Driving simulators are established means to examine driver performance and behavior. Driving simulator studies constitute an important proportion of published papers in relevant journals. Whereas real-world studies lack appropriate control elements, simulator studies allow the investigation of experimental manipulation, treatments, impairments, and what-if scenarios related to new technologies (7). The latter are mostly devoted to specific topics referring to individual behavior of single drivers in safety-critical situations. To our best knowledge, there is no simulator aiming especially at social aspects of route choice, e.g. fairness or social dilemmas.

Basically, a driving simulator is "an interactive system in which a respondent 'drives' an animated vehicle in a computer-generated 'graphic environment'" (6). A number of driving simulator implementations with different levels of participant immersion have been proposed (e.g., (17; 16; 23; 14; 27)). It is mandatory to carefully decide based on study goals which kind of implementation is appropriate as the decision affects study costs and influences the validity of the obtained results.

Regarding route choice experiments, the goal is, in general, to achieve behavioral correspondence by reproducing the behavioral environment so that participants' responses are similar to those expected under real-world conditions (6).

Existing work often uses low-fidelity approaches – the so-called stated preference or response approaches. Their main benefit is repeatability. Although these approaches may lead to valid results, they are also prone to errors (6; 26; 19). In order to improve route assignment according to specific traffic demand and travel routines within a real existing geographic area, we would in contrast like to observe route choice of fully immersed participants. This requires realistic simulations of large, repeatable, real-world-like scenarios in a driving simulator.

Adler, Recker, and McNally (4) stated that the ability to model participant choices in simulator studies is based on the manner in which the simulator can effectively translate the real-world situation, and it is based on the manner in which active physical elements of the real world are represented. In that sense, besides the simple depiction of choice situations, the real-world situation is required to consist of actually existing physical elements, serving as naturalistic cues. Besides familiar buildings and the road network, this contains other road users as well. These other road users have to be represented in a scientifically rigorous manner and should choose routes affected by their own daily routines.

Most driving simulators lack mechanisms to simulate such realistic traffic demand in large real-world scenarios. Even high-fidelity simulators only include microscopic traffic simulations that simulate so-called *realistic traffic conditions* within depleted environments and hypothetical driving scenarios. That is, they provide realistic tactical interaction of other vehicles with the observed driver. But they do neither recreate empirically assessed strategic routing of, for instance, commuters in a specific city region nor do they let several users interact strategically under real-life conditions (7; 27; 9).

A new simulation is needed. It must integrate immersion, interaction, rigorous traffic simulation, and an environment based on geodata. Under these preconditions, it would be most valuable resorting to the newest developments in the field of virtual reality (VR) and head mounted displays, too. Such displays back up the participants' identification with the scenarios, thereby enhancing realism (12; 5).

We therefore contribute PARCOURS – a software for Psychological Assessment of Route ChOices

in Urban Road networkS –, which combines the realism of traffic simulators with the participant immersion of driving simulators. PARCOURS is a 3D VR driving simulator. It encompasses real-world situations, including an actually existing road net, rigorous traffic demand, and car following models. Detailed handling of large-scale traffic phenomena requires a microscopic traffic simulation that can be executed in real time. To this end, we use SUMO and the established LuST scenario (10; 8). Multiple participants are able to drive around simultaneously together with simulated other vehicles in a virtual 3D environment that is automatically created from the SUMO scenario data. Data regarding conditions and behavior of all road users, participants, and agents, as well as of the road infrastructure state, e. g., traffic lights, are being logged and supplied for offline analysis.

The rest of this paper is organized as follows. We discuss the related work in Section 14.2. In Section 14.3, we describe the architecture of our approach in detail, and we state details about the hardware and software that we used to implement our prototype in Section 14.3.4. In Section 14.4 we explain our plans to extend the presented framework, in order to make it usable for real behavioral experiments.

14.2 Related work

Nakasone et al. (22) introduced OpenEnergySim, an immersive 3D multi-user driving simulator that integrates traffic simulation. It synchronizes a client application based on OpenSim,[1] an open source multi-platform 3D application server, and a server module controlling an X-Roads traffic simulator. The aim is to investigate advanced intelligent transport system strategies. Data of computer-controlled and user-controlled cars can be stored. Multiple users can simultaneously move virtual cars, their avatars, across urban roads using a gaming wheel interface. The simulation consists of static objects, the environment, and objects generated at runtime, e. g., vehicles. While driving, users can be provided with dynamically manipulated information on traffic lights, variable message signs (VMS), break lights, and CO_2 indicators. Environmental conditions, like weather, can be changed. Regarding the server–client communication, the authors struggled with performance issues, such as visualization smoothness due to asynchronism, different update rates, and update merging needs. Regarding vehicle management, physics, and interface issues like the steering ratio had been challenging. Inferring from the authors' description of the virtual world, the used environment shows no correspondence to an actually existing cityscape. Although OpenSim supports head-mounted displays, such as the Oculus Rift,[2] the authors did not decide to implement VR support. A similar application designed for small-scale phenomena was described by Miska and Kuwahara (20).

Gajananan et al. (11) presented a driving simulator that enables immersive multiuser driving experiments in 3D virtual environments. Their framework is also supposed to allow "to create experiences for multiple human subjects, with predictability, reproducibility and controllability in real-time" and to capture more natural responses to traffic situations. They integrated the cityscape of an urban area in Tokyo with Unity[3] and CityEngine[4]. The self-developed Distributed Virtual Environments supports 100 participants driving around simultaneously. For traffic simulation, the FrOnt framework is implemented. To smoothen the integration of simulators, the authors developed the OpenTraffic Middleware. The framework was tested with a scripted traffic accident scenario. A full-fledged large-scale traffic simulation is not provided and head-mounted displays are not supported.

The driving simulation introduced by Grasso, McDearmon and Kobayashi (13) includes a VR model of the city Phoenix, Arizona. It extends to a two square miles downtown area with exact lane conditions and controllable traffic signals. The modeling and texturing of the area is based on

---

[1] http://opensimulator.org/
[2] http://www.oculus.com/
[3] http://unity3d.com/
[4] http://www.esri.com/software/cityengine

photographs taken by a helicopter with the help of Nverse Photo[5]. The road network and traffic controls were created with UC-win/Road environment[6] which also supports extracting driver data. The authors' aim was to investigate driver behavior and vehicle emissions. They created a synthetic scenario with a traffic volume meant to be similar to a mid-afternoon weekend flow in this area. As future projects they identified integrating further behavior tracking systems and an interactive traffic control system. The observed drivers only interacted with computer steered cars.

A multiuser driving simulator was implemented by Yasar, Berbers, and Preuveneers (28) to evaluate a voice-based command assistance system. They deployed it on four different participants, each sitting in front of their own triple screen simulator. An actual intersection of Linköping was built to form the simulated environment using the stRoadDesign scripting language[7]. The traffic at the intersection consisted of twelve cars in total, eight of them acting autonomously. The participants had to drive around dealing with four different tasks and met here and there on the road net. They were provided with a gaming wheel, a gear shift, and pedals as input devices. Among others, speed, acceleration, and lane position of the drivers were logged. Participants in experiments run by Mühlbacher, Zimmer, Fischer, and Krüger (21) also shared the same virtual environment when performing in several cockpits. The aim was to study interactions between them while they were driving in a specific formation along a country road course with a length of 25 kilometers. Each car could be steered by a gaming wheel with force feedback and accelerated by pedals. The scenes were displayed on triple screens with a 150 degrees field of view. As a major drawback the authors stated that only a small number of drivers could be observed at the same time. Inferring from the authors' description the scenario was artificial.

Another multi-user approach was followed by Noth (24) to investigate more human-like driver-driver interaction. The virtual world and the vehicles were build with different modules of TrafficSimulation[8], a software with the focus on software/ hardware in the loop testing. Plug-ins were used to implement traffic lights and vehicles' CAN bus signals. Some add-ons were developed by the author facilitating multi-user functionality and remote client communication. Major technical problems had to be solved, namely, synchronicity, network failures, and design of a task-related road network. Participants were seated in a driving cabin equipped with a force feedback steering wheel and pedals. An artificial virtual road was rendered to one screen that was mounted in front of them. The number of participants driving simultaneously was limited to ten. The system was tested in a lane change task, where interaction was a crucial component. The results showed a different driver behavior compared to a single-user scenario.

OpenDS[9] is a driving simulator that is aimed mainly at human-machine interaction and comes with support for VR headset as well as gaming wheels (3). OpenDS ships with a number of predefined scenarios called "driving tasks" and allows for easy manual creation of additional 3D scenes along with fully scripted driving tasks. To some extent integration with a traffic simulation, namely SUMO, is possible, e.g., for the simulation of a junction's traffic light system, but we found no reference indicating that automatic coupling with pre-defined SUMO scenarios is possible. OpenDS employs the jMonkeyEngine game engine[10] for 3D scene rendering and sub-microscopic modelling of driving physics, which could prohibit simulation of city-sized scenarios in real time on standard workstation hardware. OpenDS appears to be restricted to single-user operation.

VIRES VTD[11] (Virtual Test Drive) is a driving simulator that is aimed at assessment of Advanced Driver Assistance Systems. It includes tools for scenario generation, sub-microscopic traffic simulation, sound simulation, and an API for third-party plugins. VTD is capable of interconnected

---

[5]http://precisionlightworks.com/nversePhoto.html
[6]http://www.ensoftinc.com/main/products/winroad/m_winroad.html
[7]http://www.stsoftware.nl/StRoadDesign.html
[8]http://www.nisys.de/products/trs/
[9]http://opends.de/
[10]https://github.com/jMonkeyEngine/jmonkeyengine
[11]http://www.vires.com/products.html

operation of multiple simulators in one scenario. We found no indication, that VR headsets are supported as display devices. Schiller et al. (**?**) have shown that VTD can be coupled with SUMO in order to reduce computation time, so that real-time simulation with VTD becomes feasible for larger scenarios.

The specified simulators do not satisfy the need of a full-fledged state-of-the-art tool to assess strategic route choice behavior. Each one comes along with constraints that put the augmentation of validity compared to more low-fidelity approaches into question. PARCOURS in turn integrates microscopic traffic simulation, latest participant immersion techniques, and interaction of multiple participants within an actual, existing road network. Participants can experience traffic related cues in the virtual world as they experienced them in real-life.

14.3 PARCOURS

PARCOURS is designed as a client-server architecture. The server host runs a SUMO process in real time, that is, with a real-time factor of one that is artificially enforced. The SUMO process serves as the central traffic simulation engine. Most vehicles' driving behavior is controlled by SUMO directly, whereas some vehicles are manipulated by participants. The server continuously communicates the state of all simulated vehicles that are close to a client-controlled vehicle to the corresponding client. Multiple clients may connect to the server and may simultaneously interact with the server's simulation. On the client hosts, a 3D video game engine[12] renders a scene that represents the SUMO scenario in real time. Each client is associated with exactly one vehicle in the simulation, and the scene is rendered in the first person perspective of the vehicle's driver. The client user controls her ego-vehicle with a steering wheel and pedals, while other car objects in the client's 3D scene follow the trajectories of the corresponding vehicles as retrieved from the server. The state of the client's associated vehicle is periodically communicated back to the server, which accordingly adjusts the state of the client-controlled vehicle within the SUMO simulation. Thereby, the client user experiences an interactive simulation scenario. The user is able to steer her vehicle, and she is affected by the traffic around her but also influences the surrounding traffic. An overview of the system architecture is given in Fig. 14.1.

In the following, we describe the communication, server, and client components of our architecture in more detail.

14.3.1 Communication

The communication is implemented as a star topology. That is, each client communicates directly with the server via an IP network, and there is no direct client-to-client communication. To increase performance, we divide our communication protocols into two modules: a static and a dynamic part. When a client spawns, it initially contacts an HTTP web server to fetch static information that does not change during the simulation run. The client downloads the SUMO network that is being simulated using the static protocol module.

Time-critical dynamic information, such as the continuously changing vehicle states, are not transferred via HTTP. Instead, they are communicated via a custom protocol on top of a TCP connection. We term this protocol the dynamic data connection (DDC). The DDC is implemented as a bidirectional stream of messages. The server periodically provides each connected client with the state of the SUMO-controlled vehicles, as well as other client vehicles. State information includes the vehicles' current position, heading, trajectory, as well as current traffic light status. In order to improve scalability, we implemented a filter that removes information about vehicles which currently do not influence the client. Specifically, all vehicles that are outside the client's rendered visible range are

---

[12]We used Unity 5 in our prototype implementation.
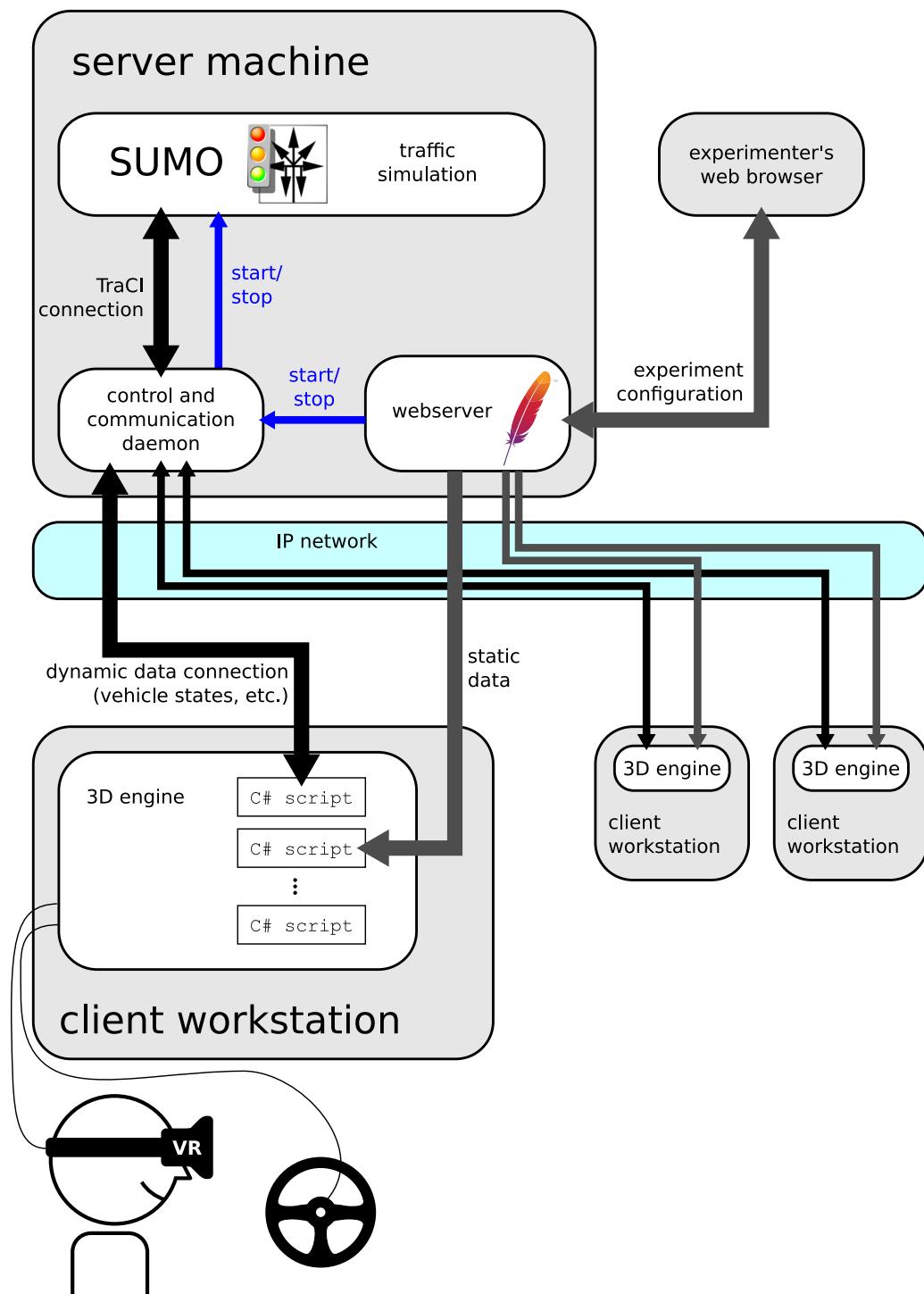
Figure 14.1: Schematic overview of system architecture.

not transmitted by the server. Clients periodically transmit to the server their current user-modified state, that is, their current position and heading. The client state is used by the server to feed the user's manipulations back into the SUMO simulation, and it is used to update the vehicle list filter for the next communication interval.

14.3.2 Server architecture

The server software is composed of three modules: a web server, SUMO 0.24, and a control and communication daemon (CCD).

The web server serves two purposes: first, it provides an interface that allows to manage the experiments. That is, it allows to choose a driving scenario and to start the CCD with the appropriate configuration. When an experiment run has finished, the web interface allows to retrieve the logged data that the experimenter is interested in. Second, the web server provides bulk static data to the client, such as the SUMO network, the pre-rendered 3D scene, and so forth.

The CCD, besides starting and terminating the SUMO process, triggers simulation steps at a constant tick rate, the reciprocal step length, thereby maintaining a real-time factor of one. We use SUMO's traffic control interface (TraCI) for all manipulations of the traffic simulator state. In the following, we describe the server's two main tasks: integrating client-controlled vehicles with the simulation and maintaining a constant SUMO timing.

Using a TraCI Context Subscription, the CCD extracts the state of all vehicles in the vicinity of each client-controlled vehicle. Then, it sends the appropriate set of vehicle states to each client. The CCD also sends the the state of all traffic lights to each client once per time step, enabling the clients to display that information appropriately.

Between two simulation steps, the CCD receives vehicle state updates from each client. Client updates may comprise no updates at all, one update, or many updates. In case of network congestion or packet loss, clients may not be able to provide updates for each simulation step. We then use SUMO's traffic simulation to predict client movement in between updates. Otherwise, clients will usually provide multiple updates per simulation step, because the client simulates vehicle movements at its frame rate, which is usually much higher than the server's tick rate. Therefore, the server keeps a list of the most recently received vehicle state records for each client. When a new record is received from a client, it overwrites the corresponding entry in the client vehicle state list. Immediately prior to triggering a simulation step, the CCD updates the state of the client vehicles in the simulation using TraCI's "move to VTD" command (0xb4).

Communication with the clients is done in a separate thread for each client, such that the main thread, which interacts with SUMO trough TraCI, is not affected by network communication lags. The desired tick rate is set to 10 Hz in our experiments as a compromise between network traffic requirements and smooth updates of other vehicles at the client. Immediately before the first simulation step, the wall clock time is recorded as a reference value, in order to define the nominal schedule time of all subsequent steps. For each simulation step, after vehicle states have been extracted, the main thread sleeps for a duration corresponding to the difference between the current wall clock time and the nominal time of the upcoming step. This results in the desired behavior of a fixed traffic simulation tick rate, independent from the computation and communication time experienced in a specific situation.

14.3.3 Client architecture

The client application renders a 3D scene representing a SUMO scenario in real time. The ground, roads, and buildings make up a static 3D scenery through which the user steers her car. Specifically, the following components are included:

- The ground of the world is a horizontal plane.

- Roads and road markings are drawn at the same level as the ground.

- Traffic lights are currently depicted as spheres that hover above the end of the corresponding lane; they are dynamically colored to match the active traffic light phase (see Fig. 14.2(a)).

- Buildings are rendered as right prisms, resulting from an extrusion of the polygons supplied in the `.poly.xml` file that comes with the LuST scenario. Buildings are uniformly extruded to a height of 6 meters.

| (a) junction with traffic light system | (b) heavy traffic on the left lane |
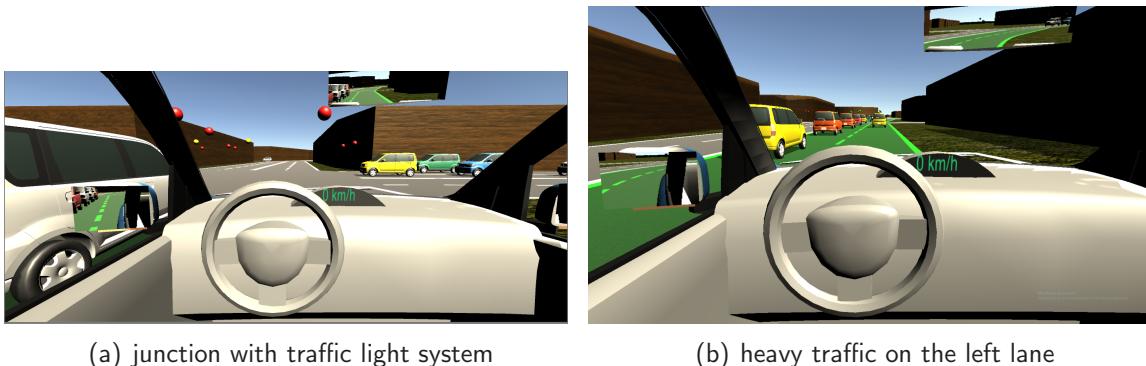
Figure 14.2: Virtual driving environment from the user's perspective.

Creating 3D objects out of the SUMO scenario data is a computationally intensive task. Therefore, the static part of the scene is generated once, stored on the server, and deployed to the clients at startup.

Cars are represented by a 3D model of a Mitsubishi eK (1). The car model we use includes a model of the car's interior. To increase rendering performance, the interior is rendered only for the client's own vehicle and is removed for all other vehicles. To improve user immersion, the client's vehicle is further enhanced with interior and exterior rear-view mirrors, as well a digital speedometer[13] (see Fig. 14.2(b)).

The client-steered car's behavior is simulated by the physics module of the used video game engine. It can be controlled by the client user through the steering wheel and pedals. The state of this vehicle is sent to the server via the DDC at the client's frame rate.

The positions of the assets corresponding to other vehicles are updated whenever vehicle state updates are received from the server via the DDC, thereby effectively letting other vehicles "drive" through the scenery.

Bridges and tunnels

In order to map the road network description given in SUMO's `.net.xml` format to a 3D scenery, we had to overcome several challenges. When one road crosses below a bridge carrying another road, the two of them intersect in the horizontal plane without logically intersecting in a junction. When rendering all roads in a horizontal plane, this creates several issues. First, it degrades the user's driving experience, since it is hard to distinguish true junctions from false intersections from the driver's perspective. Second, if other cars drive on both falsely intersecting roads, this renders the illusion that the other cars drive through each other. Third, in our implementation, we make use of the game engine's colliders to prohibit cars to pass through each other, as well as to forbid the user to leave the road. These road border colliders would, therefore, make it impossible for any vehicle to pass such an intersection.

---

[13]The design of these enhanced user interface elements is simple and pragmatic and is not intended to match the design of the real Mitsubishi eK's interior.

To resolve these issues, it is necessary to use non-trivial elevation for at least one of the intersecting roads. While the SUMO `.net.xml` file format is capable of storing elevation coordinates for junctions, this information is not included in the LuST scenario, mainly because the latter is based on OpenStreetMap (OSM) data, which in turn does not rigorously provide this data. Our solution is to render roads and junctions at ground level by default and to specify non-vanishing elevation values for selected junctions by hand in an additional file that is read when the static world scene is build from the SUMO scenario data. Junctions are then rendered at their specified elevation and road segments connecting junctions of different elevation are interpolated accordingly. For the sake of simplicity, we allow only non-negative elevation values, thereby making bridges possible but renouncing tunnels.

14.3.4 Implementation aspects

We assessed scalability and performance aspects of the system architecture and algorithms in the PARCOURS prototype implementation. The server builds upon vanilla SUMO 0.24 as the microscopic traffic simulator. The experimenter's web interface is written in PHP, and the CCD in Python 2.7 using the Python TraCI module that is included with SUMO.

The client is written in C# using the Unity 5 video game engine[14], which provides a real-time 3D rendering engine that can readily provide stereoscopic imagery for the Oculus Rift DK2 (2). We make excessive use of the game engine's physics engine, especially that of the car object module. The physics engine allows for vehicle physics simulations that incorporate a mapping of steering wheel and pedal input to front wheel angle and torque, respectively. As traffic scenario we use LuST 1.1 (8).

As the client machine in our prototype experiments we use a workstation powered by an Intel® Core™ i7-5820K hexacore processor, 32GB RAM, and an NVIDIA® GeForce® GTX 970 OC graphics processing unit. An Oculus Rift DK2 with head motion detection is used as VR display and a Logitech® Driving Force™ GT gaming wheel with pedals as input device. Figure 14.3 shows a mobile demonstration unit consisting of two complete client units mounted on a utility cart.

Figure 14.3: Mobile demonstration unit with two complete client units.

Initial experiments with simultaneously driving persons indicate promising results and provided positive feedback on user immersion.

---

[14] http://unity3d.com/unity/personal-edition

14.4 Conclusion and outlook

We have presented a framework to conduct experiments on strategical driving behavior using an immersive multi-user driving simulator hooked into a real-time microscopic large scale traffic simulation. With our implementation we demonstrated the feasibility of steering individual cars in a SUMO process, while displaying the surrounding traffic in real-time with a lightweight client software implemented using a video game engine. We will extend the presented implementation to further enhance realism and suitability for behavioral studies.

We plan to implement interpolation of vehicle positions in order to smoothen the perceived vehicle motion. We expect that some changes to the SUMO lane change model will be necessary for a reliable smooth display of continuous lane changes. Especially lane changes that take place over a very short driven distance, like forth-and-back changing within a few meters, are not easy to depict realistically in a virtual 3D environment.

The SUMO network data used for constructing the 3D scene does not contain any description of the non-functional environment apart from polygon shapes of the buildings. However, the original OSM data, which the scenario is based on, contains, to some extent, more detailed information, such as building height, front colors, and roof type, positions of trees, different ground types, and so forth. After realigning the SUMO network data to the original OSM data, we could build upon of this information in order to render a more realistic model of the environment.

The experimenter's user interface will be extended to allow for more sophisticated experiment definition and layout, specifically for the purpose of assessment of route choice behavior. The experimenter should be able to dictate the route or the set of routes that are suggested to a test person to follow in order to fulfill her driving task. To realize this, a route planning module will be included that supports automatic generation of alternative paths as well as manual creation by the experimenter. The virtual in-vehicle interface of the client will be extended by a mini-map, displayed as virtual navigation device, that can be programmed by the experimenter to display different sets of proposed alternative routes to the test person. In order to assess the effects of driving instruction representation, we plan to further extend the interface with a programmable virtual head-up display. A basic example hereof is already implemented in our prototype: the proposed route to follow is displayed by highlighting the corresponding road in green color.

In addition, the data recording capabilities will be extended to recording of head motion and viewing direction, as well as support for standard instruments used in empirical psychology, such as skin conductance, heart rate and other affective measures.

Besides the study of current navigation decisions, PARCOURS could be used in the future to study a number of other effects. For instance, we can study advances in autonomous driving, where users do not have to drive at all but may want to interact with a route choice module determining the next destination. The experience of large scale traffic will be enhanced when not competing with other tasks like driving and may be even more important regarding strategic route choice.

Acknowledgment

14.5 References

[1] *Car Mitsubishi Wagon N050810 - 3D model*. Accessed: 2016-02-21.

[2] *Oculus Rift Development Kit 2*. `https://www.oculus.com/en-us/dk2/`. Accessed: 2016-02-22.

[3] *Opends—Features*. https://opends.de/software/features. Accessed: 2016-02-29.

[4] ADLER, JEFFREY L., WILFRED W. RECKER and MICHAEL G. MCNALLY: *A conflict model and interactive simulator (FASTCARS) for predicting enroute driver behavior in response to real-time traffic condition information*. Transportation, 20(2):83–106.

[5] BLASCOVICH, JIM, JACK LOOMIS, ANDREW C BEALL, KIMBERLY R SWINTH, CRYSTAL L HOYT and JEREMY N BAILENSON: *Immersive virtual environment technology as a methodological tool for social psychology*. Psychological Inquiry, 13(2):103–124, 2002.

[6] BOVY, PIET HL and ELIAHU STERN: *Route choice: Wayfinding in transportation networks*, 1990.

[7] CARSTEN, OLIVER and A. HAMISH JAMSON: *Chapter 7 - Driving Simulators as Research Tools in Traffic Psychology*. In PORTER, BRYAN E. (editor): *Handbook of Traffic Psychology*, pages 87 – 96. Academic Press, San Diego, 2011.

[8] CODECA, L., R. FRANK and T. ENGEL: *Luxembourg SUMO Traffic (LuST) Scenario: 24 hours of mobility for vehicular networking research*. In *Vehicular Networking Conference (VNC), 2015 IEEE*, pages 1–8, Dec 2015.

[9] COOPER, JOEL M, IVANA VLADISAVLJEVIC, NATHAN MEDEIROS-WARD, PETER T MARTIN and DAVID L STRAYER: *An investigation of driver distraction near the tipping point of traffic flow stability*. Human Factors: The Journal of the Human Factors and Ergonomics Society, 51(2):261–268, 2009.

[10] D. KRAJZEWICZ, J. ERDMANN, M. BEHRISCH AND L. BIEKER: *Recent Development and Applications of SUMO - Simulation of Urban MObility*. International Journal On Advances in Systems and Measurements, 5(3&4):128–138, December 2012.

[11] GAJANANAN, K., A. NANTES, M. MISKA, A. NAKASONE and H. PRENDINGER: *An Experimental Space for Conducting Controlled Driving Behavior Studies based on a Multiuser Networked 3D Virtual Environment and the Scenario Markup Language*. IEEE Transactions on Human-Machine Systems, 43(4):345–358, July 2013.

[12] GLENN W. HARRISON, ERNAN HARUVY, E. ELISABET RUTSTRÖM: *Remarks on Virtual World and Virtual Reality Experiments*. Southern Economic Journal, 78(1):87–94, 2011.

[13] GRASSO, CHRISTOPHER J., MICHAEL J. MCDEARMON and YOSHIHIRO KOBAYASHI: *Virtual Driving and Eco-simulation: VR City Modeling, Drive Simulation, and Ecological Habits*. In *Proceedings of the 2010 Spring Simulation Multiconference*, SpringSim '10, pages 199:1–199:7, San Diego, CA, USA, 2010. Society for Computer Simulation International.

[14] GREEN, PAUL A, HEEJIN JEONG and TE-PING KANG: *Using an OpenDS Driving Simulator for Car Following: A First Attempt*. In *Adjunct Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 1–6. ACM, 2014.

[15] JAHN, OLAF, ROLF H. MÏ¿½HRING, ANDREAS S. SCHULZ and NICOLÏ¿½S E. STIER-MOSES: *System-Optimal Routing of Traffic Flows with User Constraints in Networks with Congestion*. Operations Research, 53(4):600–616, 2005.

[16] KOUTSOPOULOS, HARIS N., TSIPPY LOTAN and QI YANG: *A driving simulator and its application for modeling route choice in the presence of information*. Transportation Research Part C: Emerging Technologies, 2(2):91 – 107, 1994.

[17] LEISER, DAVID and ELIAHU STERN: *Determinants of subjective time estimates in simulated urban driving*. Transportation Research Part A: General, 22(3):175 – 182, 1988.

[18] LUJAK, MARIN, STEFANO GIORDANI and SASCHA OSSOWSKI: *Route guidance: Bridging system and user optimization in traffic assignment*. Neurocomputing, 151, Part 1:449 – 460, 2015.

[19] MCFADDEN, DANIEL and GREGORY K. LEONARD: *ISSUES IN THE CONTINGENT VALUATION OF ENVIRONMENTAL GOODS*, chapter 6, pages 165–215.

[20] MISKA, MARC and MASAO KUWAHARA: *Nanoscopic Traffic Simulation with Integrated Driving Simulator to Investigate the Sag Curve Phenomenon*. SEISAN KENKYU, 63(2):153–158, 2011.

[21] MÜHLBACHER, DOMINIK, JUTTA ZIMMER, FLORIAN FISCHER and HANS-PETER KRÜGER: *The multi-driver simulator–a new concept of driving simulation for the analysis of interactions between several drivers*. Human Centred Automation, pages 147–158, 2011.

[22] NAKASONE, ARTURO, HELMUT PRENDINGER, MARC MISKA, MARTIN LINDNER, RYOTA HORIGUCHI and MASAO KUWAHARA: *OpenEnergySim: A 3D Internet Based Experimental Framework for Integrating Traffic Simulation and Multi-user Immersive Driving*. In *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, SIMUTools '11, pages 490–498, ICST, Brussels, Belgium, Belgium, 2011. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[23] NILSSON, LENA: *Behavioural Research in an Advanced Driving Simulator - Experiences of the VTI System -*. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 37(9):612–616, 1993.

[24] NOTH, SEBASTIAN: *A multi-user driving simulator for studying human driving*. PhD thesis, Bochum, Ruhr-Universität Bochum, Diss., 2014, 2016.

[25] NOVACO, R.W.: *Transportation, Psychology of*. In BALTES, NEIL J. SMELSERPAUL B. (editor): *International Encyclopedia of the Social & Behavioral Sciences*, pages 15878 – 15882. Pergamon, Oxford, 2001.

[26] TILAHUN, NEBIYOU Y. and DAVID M. LEVINSON: *A Moment of Time: Reliability in Route Choice Using Stated Preference*. Journal of Intelligent Transportation Systems, 14(3):179–187, 2010.

[27] WATSON, GS, YE PAPELIS and LD CHEN: *Transportation safety research applications utilizing high-fidelity driving simulation*. Publication of: WIT Press, 2003.

[28] YASAR, ANSAR-UL-HAQUE, YOLANDE BERBERS and DAVY PREUVENEERS: *A computational analysis of driving variations on distributed multiuser driving simulators*. In *IASTED International Conference on Modelling and Simulation*, number 19, pages 178–186. ACTA Press, 2008.

[29] YOUN, HYEJIN, MICHAEL T. GASTNER and HAWOONG JEONG: *Price of Anarchy in Transportation Networks: Efficiency and Optimality Control*. Phys. Rev. Lett., 101:128701, Sep 2008.

15 Enhanced Emergency Calls for Next Generation Vehicle Communications: Performance Analysis using SUMO with NS3 and a Dedicated Testbed

Leandro D'Orazio[1], Davide Domeniconi[2], Alessandro Marchetto[1], Fabrizio Granelli[2], Marco Darin[1]

[1] *Centro Ricerche FIAT (CRF), Trento Branch, Infotelematic System Unit,*
Via della Stazione, 27, 38123, Trento (ITALY)
[2] *University of Trento, Department of Information Engineering and Computer Science (DISI),*
Via Sommarive, 9, 38123, Trento (ITALY)

{leandro.dorazio, alessandro.marchetto, marco.darin}@crf.it
{d.domeniconi, fabrizio.granelli}@unitn.it

Abstract

Modern vehicles are increasingly equipped with advanced driver assistance systems (ADAS) and connectivity features for coping with the market requirements and meeting the customers' expectations. The mandatory introduction of additional devices (e.g., the eCall platform) will reinforce this trend and could be exploited by car manufacturers as an opportunity for enhancing the safety and entertainment features of their products through the newly available computational capabilities (which can be, e.g., allocated for improving the positioning accuracy of the vehicle or its communication capabilities). Nevertheless, analyzing the performance of new vehicular services based on communication still represents an open challenge, due to mobility and network modeling. This paper investigates the possibility to use a SUMO/NS3-based simulation platform for supporting the performance analysis of new vehicular services based on wireless communication. The ADAS application developed in the e2Call research project is taken as example for highlighting the benefits deriving from performance analysis based on simulation and supported by testbed measurements. This methodology could be used for supporting the development phase of new automotive features. In fact, the traditional extensive testing phase which is usually completed with a limited number of vehicles could be enriched with specific hints coming from the simulation activity since the early prototyping stage (with a particular emphasis on the identification of communication problems which could arise in congested traffic or network conditions).

Keywords: SUMO, NS3, LTE, ADAS.

15.1 Introduction

Enhanced vehicular communications and especially emergency calls are expected to play a relevant role in the future of telematics services for road safety. Indeed, timely and effective delivery of information about accidents is expected to provide most efficient intervention and appropriate medical support to injured people. Given the required coverage and pervasiveness of such applications, the usage of Wide Area Networking solutions, such as LTE and LTE-Advanced cellular networks, represents a proper candidate to support such services. However, while waiting for the fifth generation of cellular networks (5G) to be standardized and deployed (expected in 2020), capacity requirements and request for provisioning of a satisfactory level of service quality (in terms of transmission capacity and latency) from existing technology introduce a big challenge along the road to make such services a reality in the next years. Performance analysis can be performed in two ways:

- Through testbed measurements: as LTE is already deployed, it is possible to equip vehicles with cellular modems and capture actual traces to enable a realistic performance evaluation. Here, the major limitation is the cost for achieving significant results, which might limit the number of vehicles and the possibility for scaling the evaluation to a high vehicle density.

- Through software simulations: simulation platforms exist that allow to study vehicular mobility (such as SUMO (4)) and mobile communications (such as NS3 (16)). By merging both environments it is possible to perform complex simulations with a high degree of freedom in controlling the environment variables, e.g., number and characterization of vehicles, mobility patterns. However, in this case, it is not easy to understand how achieved results are near to what can be expected in case of a real deployment of the service.

The paper is intended to bridge the gap between the two approaches above and provides a comparison in terms of performance evaluation provided by simulation, using SUMO and NS3, and by testbed measurements, developed in the framework of the e2Call project (6). The simulation-related activities have been conducted within a collaboration between the Centro Ricerche FIAT (CRF (3)) and the Department of Information Engineering and Computer Science (DISI) of the University of Trento (5). The paper is structured as follows: Section 15.2 presents the scenario of the e2Call project, while the following Sections 15.3 and 15.4 describe the project testbed and the corresponding simulation implementation, respectively. Section 15.5 provides an analysis of the achieved results and a comparison to evaluate whether simulations and testbed measurements are aligned. Finally, Section 15.6 concludes the paper with final remarks.

15.2 Project Scenario

The number of road deaths decreased by 52.4% in Italy between 2001 and 2014, resulting in 55.6 deaths per million inhabitants in 2014 (12). A similar trend has been registered in the whole European Union (EU28), with an overall average value of deaths which is lower than Italy (i.e., 51) and even better for several other member states (e.g., United Kingdom, Spain, Germany and France). Between 2010 and 2014, the number of victims in the EU28 is decreased of 18% and a similar trend has been registered in Italy with a 17.8% decrease. Although the achieved good results and the continued European measures to do better in terms of performance evaluation for road safety linked to death reduction, the road accidents problem still has an high social impact in terms of suffering, medical costs, lost productivity, insurance costs and so on. For this reason the recent European programs for road safety and the new 2020 EU targets have set new standards, over the halving of the number of fatalities, that also includes, with an harmonized definition, the impact of serious injury.

At the same time, all vehicles are expected to be equipped with the pan European eCall (emergency call) platform. The mandatory introduction of this device on each new type approved vehicle will enable the possibility to potentially support additional advanced driver assistance systems and mobility services. In Europe the iHeERO project (9) is working for the upgrade of the existing European infrastructure for enabling eCall based on 112 to be handled correctly across all member states, defining operational and functional requirements needed to integrate PSAPs (Public-Safety Answering Points) at different levels. The eCall could be the key to move towards a standardized European solution.

Nowadays the cooperative mobility network should enable the use of different communication technologies with prioritized channels depending on time critical safety application as real time warning or potential danger warning and so on. In this context, the in-vehicle eCall platform will offer the possibility to integrate long range 2G/3G/4G (LTE: Long Term Evolution) with short range communication (ITS-G5, (7)) and with more accurate positioning systems. This roadmap is promising a future in which the vehicles will be always connected and localized with high precision. In the e2Call project (6), co-funded by the local Public Authority of the Autonomous Province of Trento (PAT, Provincia Autonoma di Trento) in Italy, in collaboration and Telecom Italia (19), Centro Ricerche FIAT is evaluating innovative solutions based on accurate localization and on broadband networking, with the final goal to improve road safety. In particular, three different solutions have been analyzed throughout the project: an Advanced Driver Assistance System (ADAS) (i.e., technological on board solutions for assisting the driver and thus enhancing its safety), the Enhanced Emergency Call (e2Call) (i.e., the capability of reporting detailed information on an accident to the answering points in order to maximize the eCall effectiveness), and the Road Safety Information Service (RSIS) (i.e., the possibility to use the data collected from the car and other sources for providing to the drivers real time information related to the road status).

This paper will focus on the ADAS application, highlighting the opportunities offered by the SUMO/NS3 based software simulator for analyzing its performances under different traffic and environment conditions. The system architecture is described in Figure 15.1.
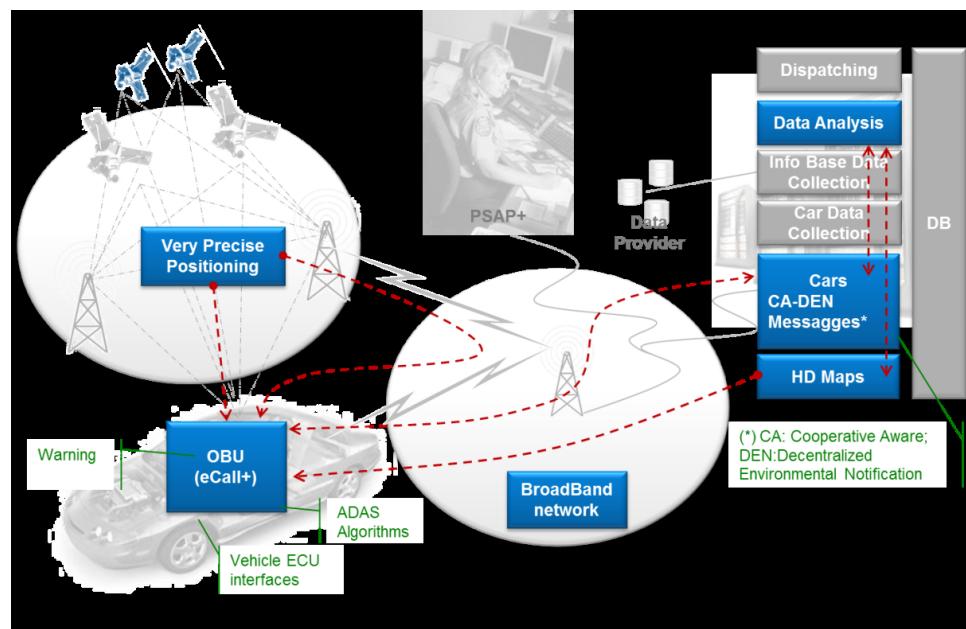

Figure 15.1: Concept of ADAS architecture

The novel ADAS application, improved with precise positioning and vehicle to vehicle information exchanged at low latency (below 100 msec) via broadband network, is managed within the on-board unit with the objective to analyze, in real time, the recognition of critical situations in different road
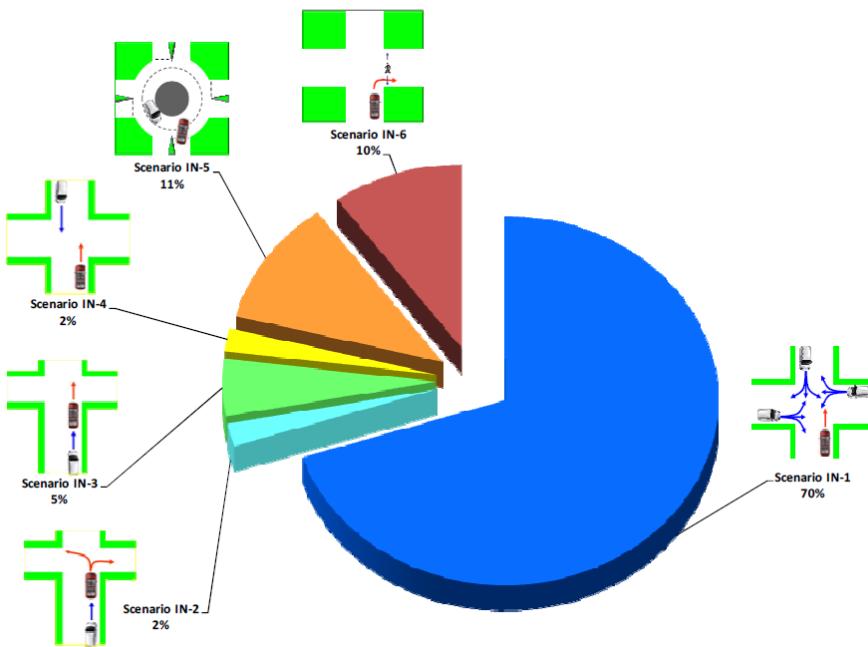
Figure 15.2: Road accident occurrences

scenarios such as: intersections, with or without traffic light, and roundabouts. These scenarios
have been selected considering the statistical results of public European projects that have identified
accident causation factors according to individual pre-accident driving situations (Figure 15.2).
Dedicated algorithms, based on a layered control architecture with human-like vehicle primitive
(already presented in (14)), have been implemented to predict the potential vehicle trajectories and
the probabilities of intersections with the surrounding vehicles that are exchanging, at the same
time, their own predictions. The output for the driver is an internal warning, available from the
in-vehicle cockpit, related to vehicle dynamics and consequently risk levels.

15.3 Testbed Description

A territorial test-site has been created with the aim of testing the previously described applications
under real operative conditions, with a particular emphasis on the ADAS solution. For doing that,
three different zones with the required road geometries have been identified in the sub-urban area
of the Trento city, the necessary fixed infrastructure has been deployed or modified for offering
both precise positioning and broad-band communication services (i.e., RTK and LTE, respectively),
and two vehicles have been equipped with the components required for realizing the applications of
interest. In the following, the three main components of the implemented testbed (i.e., territorial
zones, fixed infrastructures, and vehicles) will be described.

The urban and sub-urban territory of the Trento (Italy) city has been analyzed for identifying the
most suitable intersections and roundabouts to be used for our testing activities. The main criteria
we considered for the final selection have been the LTE/RTK network coverage and the traffic
density. In fact, the LTE commercially available network has nowadays a good coverage in urban
areas, while the RTK corrections can be received up to about 10 km from the base station which
will be described later on. The decision to identify areas which are characterized by medium traffic
load is mainly driven by the need to conduct real on-road testing in safe conditions (i.e., avoiding
strongly congested areas) but still under real operative conditions (i.e., with the presence of other
cars in the selected environment). These criteria led to the identification of the following testing
zones located in the sub-urban area of Trento: an intersection and a roundabout in Mattarello

Figure 15.3: Satellite image of the Mattarello intersection (taken from (11)) and data acquired for its detailed mapping

Figure 15.4: Satellite image of the Mattarello roundabout (taken from (11)) and data acquired for its detailed mapping

(Trento), an additional roundabout in Romagnano (Trento). Their satellite images are depicted in Figure 15.3, Figure 15.4, and Figure 15.5, respectively. The data acquired for realizing the detailed map of each area are also reported in the corresponding figure.

As far as the fixed infrastructure is concerned, two different deployment activities have been executed for the positioning and communication services. An RTK base station has been installed at the Trento Branch CRF premises. It is aimed at calculating the corrections which can be used by the mobile rover stations (i.e., the vehicles) for improving their positioning accuracy.

The collaboration between CRF and Telecom Italia gave us the possibility to obtain an ad-hoc modification of the existing commercial LTE network for implementing a sort of broadcast solution between the vehicles registered to this service (i.e., the packets sent by a certain vehicle are redirected to all the other vehicles simultaneously registered to the network). A dedicated server has been installed in the core network for realizing this feature. More details on this customization process can be found in (10).

The remaining component of the developed testbed is represented by the vehicles. Two cars (i.e., a Fiat 500L and a Lancia Delta) have been fully equipped with the hardware and software components required for implementing the solutions of interest. The main installed components are the RTK

Figure 15.5: Satellite image of the Romagnano roundabout (taken from (11)) and data acquired for its detailed mapping

Figure 15.6: Antennas of the RTK rover station mounted on the CRF car rooftop

rover station (for measuring the position of the car by using the corrections received from the base station), an application unit (an industrial PC which contains the software components which implements the various applications) and an LTE transceiver. One of the two equipped cars is depicted in Figure 15.6, with a detailed view of the antennas mounted on its rooftop. We are conscious that using more than two vehicles could be appreciated to emulate realistic scenarios, however, for the project purpose, we have equipped only two vehicles to have a preliminary even if quite representative scenario in the e2Call project.

The whole testbed resulting from the combination of the selected territorial areas, the dedicated fixed network infrastructures and the vehicles has been used for calibrating the software simulator described in the next section following the procedure described in the forthcoming part related to the experimental results.

15.4 Simulation Platform

The simulation process aims to study the vehicular communication in specific test-sites by using LTE technology. To reach this objective, we used SUMO (4) to implement the mobility part, in cooperation with Network Simulator 3 (NS3 (16)) for the communication part. The process has involved 4 different steps as shown in Figure 15.7: (1) the generation of a realistic map of the considered testing area, (2) the definition of vehicular mobility, (3) the realization of mobility trace files for a given configuration, and finally (4) the simulation of vehicular communication based on the traces generated in the previous step.

The SUMO simulator was used to implement the traffic model and generate the output files listing the position of vehicles in a given configuration at each timestep of the simulation. Each testing area was first selected in OpenStreetMap (17) and then converted to a SUMO-map using the Netgenerate tool. As example, Figure 15.8 shows a zoom of the previously described Romagnano roundabout (depicted in Figure 15.5) imported in SUMO.

The mobility part was led following two different approaches: the routes of target vehicles (i.e., vehicles simulating the two CRF cars crossing the on-road test-sites) were specifically defined while the traffic flow was implemented with randomTrips.py (to randomly define the departure and arrival points of vehicles) and subsequently Duarouter has been used to convert the trips in routes suitable for SUMO. We are conscious the random definition of vehicle's routes could lead to traffic scenarios that are only partially realistic, but we argue that this is enough for our experimentation since we mainly aiming at simulating the presence of a quite large and realistic number of vehicles in a given area, for estimating the LTE communication infrastructure capability related to novel ADAS applications. Once defined the map and the set of route files, we ran SUMO for each vehicle configuration to obtain the corresponding trace files and used traceExporter.py to convert traces from .xml to .tcl format importable in NS3. Relevant SUMO parameters characterizing a typical
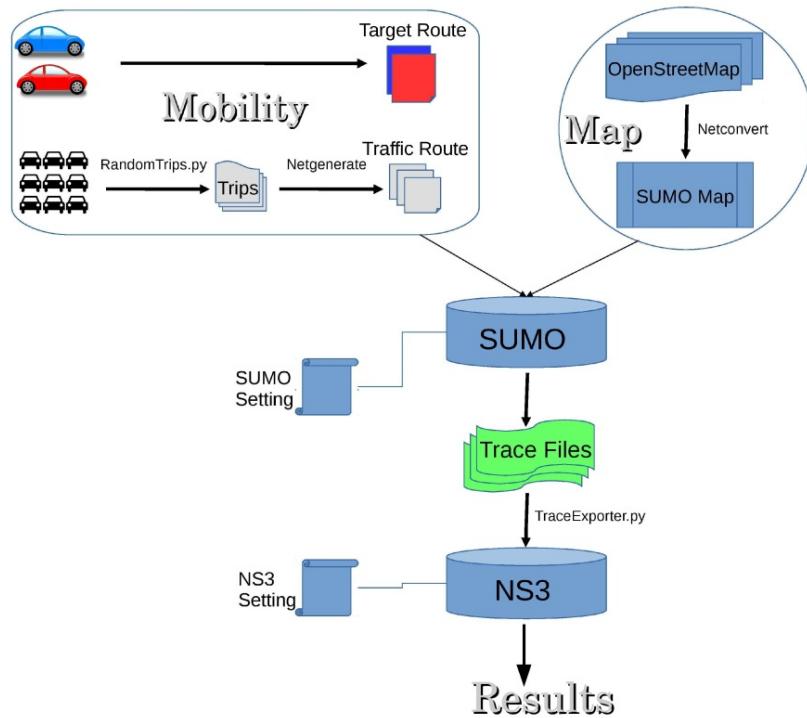
Figure 15.7: Logical flow of the simulation process

traffic scenario involve:

- vehicle type (i.e. Car, Bus, Bicycle, Pedestrian, Motorcycle, Truck, Rails);

- mobility features as maximum speed (Max Speed), Acceleration and Deceleration;

- driver attitude corresponding in responsiveness (Driver Reaction Time τ) and driving imperfection (Driver Random Behavior σ);

- vehicle geometry as Length and minimum distance from preceding vehicle (Minimum Gap);

- how vehicles join the scenario in terms of position in the network (Depart Position) and speed (Depart Speed);

- traffic model as Car Following Model and Lane Change Model.

The setting of the SUMO-based simulation environment was done to represent typical traffic conditions and suite the specificity of the selected test-site. NS3 was involved to simulate vehicular communications using LTE given the vehicles configuration previously defined in SUMO by importing the corresponding trace file, to monitor the effectiveness of wireless transmission. The NS3 code basically implements a LTE message application based on vehicle to infrastructure (V2I) and infrastructure to vehicle (I2V) architecture in which packets are transmitted both from vehicles (UEs) to the base-station (eNB), V2I architecture on uplink direction, and backwards, I2V on downlink. To collect the wireless statistics we used FlowMonitor, which is a native data collector module of NS3 that outputs xml files listing message flows and with average packet loss and delay. Relevant NS3 aspects that user can define are:

- architecture in terms of number of UEs, eNBs and how they are connected;

- antennas in terms of radiation pattern, height, gain, noise figure, transmission power;
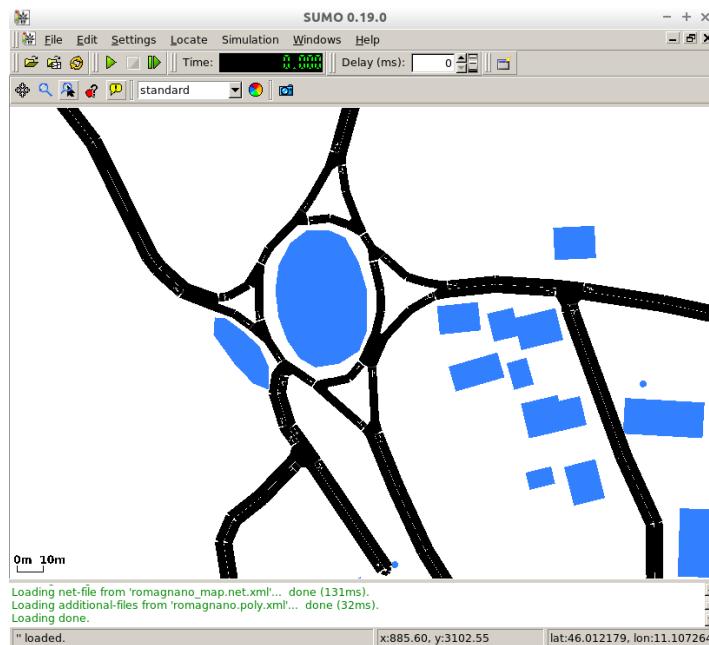
Figure 15.8: Detail of the Romagnano roundabout imported in the SUMO simulator

- frequency usage defining carrier (EARFCN) and bandwidth (Resource Blocks) for uplink and downlink;

- LTE protocol in terms of Medium Access Control (MAC) in particular with the choice of scheduler, Radio Link Control (RLC) supporting Packet Data Units (PDU) and channel control, Radio Resource Control (RRC) supporting in particular radio bearer setup, transmission mode, handover and the Sounding Reference Signal (SRS) configuration index, X2 interfaces considering interconnections of multiple eNBs;

- wireless channel by selecting the propagation loss and fading model;

- information data in terms of packet size, transmission packet interval and quality of service (QoS);

- topology by considering buildings and constructions;

- simulation scenario in terms of simulation time and simulation step;

- data output collections depending on the metrics the user wants to analyze (native NS3 module Gnuplot may be useful).

The effort in setting the environment was carried out to match the real conditions found in test-sites and to implement a realistic message application and a realistic wireless communication based on LTE protocol.

15.5 Experimental Results

In this section we briefly report an experiment we carried out, in both the on-road test-site and in the "virtual" one built on the simulator, with the aim of: (i) characterizing the simulator according to the observations derived from the test-site, and (ii) using the simulator to scale-up the analysis conducted in the on-road test-site by increasing the involved number of vehicles and by varying the environmental conditions. All the described results have been solely obtained by CRF without any involvement of Telecom Italia.

15.5.1 Procedure

We have first executed some test sessions by considering two vehicles that cross the on-road test-site areas of interest several times and in different environmental situations (e.g., early morning and evening). During these tests, we continuously monitored the communication traffic for collecting a set of interesting parameters concerning the test-site and that were then helpful for adequately configuring the "virtual" test-site in the simulator. Even though results were collected for all test-sites, only the results related to the Romagnano roundabout will be presented throughout this Section for sake of brevity.

Table 15.1 summarizes relevant parameters characterizing the simulated scenario. The area of interest for our simulation is the same of the on-road test-site and it consists in (about) $1\ Km^2$ close to Trento (Italy). The map of the area has been downloaded from OpenStreetMap (17). Table 15.1 shows that the considered scenarios is characterized by one evolved Node B (eNB), representing the base station where the LTE antenna is positioned, and a variable number of vehicles (ranging from 5 to 200) that cross the area of interest. The vehicles composing the vehicular flow of our scenario have been randomly distributed in the considered area by paying attention to preserve a realistic vehicle density (i.e., number of cars per kilometer of street) for a typical Italian urban scenario, i.e., by avoiding strong congestion as well as a totally-free scenario, and by checking that at least some vehicles cross the roundabout as expected. As content to be distributed in the scenario, we adopted textually encoded information of different size (i.e., 512 B, 1024 B, and 5000 B). The adopted vehicle mobility model is the Krauss model implemented by SUMO, in which each vehicle has a variable speed that ranges from 0 to 90 Km/h when crossing the area. The Krauss model is a microscopic mobility model in which the flow of vehicles is defined by simulating the movement of every vehicle on the street, thus assuming that: (i) an overall flow of vehicles can be defined by modelling the behavior of each vehicle; and (ii) the behavior of each vehicle depends on a set of low-level vehicle characteristics (e.g., vehicle acceleration and deceleration, driver behavior, vehicle's length, vehicle speed) that have to be defined. As simulation time, a time of 40 sec has been considered in our simulations for what concerns the execution of the scenario in SUMO (i.e., without considering the wireless communication).

Table 15.2 summarizes the parameters we used to characterize the LTE communication network. We considered the Evolved Packet Core (EPC) architecture of LTE. We adopted the unicast communication protocol for the communication between vehicles and the base station, while both unicast and broadcast schemes have been considered for the communication between the base station and vehicles. Both downlink and uplink channel bandwidths have been fixed to 20MHz (corresponding to 100 resource blocks - RB) and the LTE frequency band of 1800 MHz has been considered. The maximum transmission power for user equipment and eNB have been defined to 23 and 46 dBm respectively. The eNB node has been equipped with an antenna of height 30mt, typical for urban areas. We adopted the Friis propagation loss model, typically used to model the performance of a wireless network channel by applying a deterministic path loss model that calculates quadratic path loss as it occurs in free space. As MAC scheduler for the eNB, we adopted the PfFfMacScheduler implementation of NS3, a proportional fair scheduler that tries to balance between two interests: trying to maximize total wireless network throughput while at the same time allowing all users to receive at least a minimal level of service. For the eNB, we used the Ideal radio resource control (RRC) protocol provided by NS3 and we fixed the values of two RRC parameters: (i) EpsBearerToRlcMapping equals to RlcUmAlways (segmentation and reassembly of RLC data, RLC header is added, no delivery is guaranteed), with the aim of specifying the unacknowledged mode as the radio link control (RLC) model used for each EPS bearer (i.e., data traffic flow in the LTE network); and (ii) sounding reference signal (SRS) periodicity equals to 320, with the aim of specifying the frequency in which user equipped nodes have to send the SRS signals to the eNb, that can use it to estimate the channel quality over a wider bandwidth and to frequency selective scheduling over the channel. Finally, the link between the eNB and the first internet server has been characterized

Table 15.1: Parameters related to the simulated scenario

Parameter	Variable / fixed	Units	Value
Number of vehicles (UEs nodes)	Variable	Number	5, 10, 30, 40, 50, 100, 125, 150, 200
Number of evolved Node B (eNB)	Fixed	Number	1
Type of content	Fixed	Text, Video, Map	Text
Size of content (packet size)	Variable	Bytes	512B, 1024B, 5000B
Type of the street	Fixed	Motorway, urban, rural, extra-urban and secondary	Urban
Coverage area	Fixed	Km^2	Area of about 500mt around the roundabout of interest (about 0.8 Km^2)
Road map	Fixed	Simulated, random, manually built, real	Real map from Open-StreetMap
Type of node	Fixed	Bus, train, car, motorcycle	Car
Car speed	Variable	Km/h	0-90
Node mobility model	Fixed	Krauss model, intelligent driver model, Kerner's three-phase model, Wiedemann model	Krauss model
Node trip/route definition	Fixed	Random, semi-random, manually, map-based, based on real data, Shortest-path based, ...	Semi-random
Sumo simulation time	Variable	sec	40

by: a data rate of 100 Gb/sec, an MTU (Maximum Transmission Unit) of 1500 B, and for being consistent with the observation done in the on-road test-site, we introduced an artificial delay in this link of 50 msec. The adopted parameters are in line with the ones typically adopted in the literature concerning vehicular simulation (e.g., (18), (2), and (13)) while some of them have been defined according to the data observed in the on-road test-site.

Table 15.3 presents the main aspects investigated in our simulation-based analysis. We were interested in studying the impact on the communication of the two following aspects:

- Type of communication protocol: we considered both unicast (the base station sends the received information to all connected downlink nodes using one-to-one connections) and broadcast (the base station sends the received information to all connected downlink nodes by using the same channel for each of them) communications. This aims at understanding limits and advantages of using such communication protocols in different traffic and context conditions.

- Content size: we considered different packet size of the content sent between vehicles and base station with the aim of studying the impact on the communication of different type of content, such as: simple textual information (e.g., traffic information), geographic and map-based information, and video-based content.

Table 15.2: Parameters related to the LTE-based communication system considered in the simulation

Parameter	Variable / fixed	Units	Value
Type of LTE	Fixed	Simple, EPC	EPC
Communication protocols	Variable	Unicast, Broadcast, Multicast	Unicast, Broadcast
Dl Bandwidth	Fixed	Resource block (MHz)	100 (20 MHz)
Ul Bandwidth	Fixed	Resource block (MHz)	100 (20 MHz)
DL EarFcn	Fixed	EarFcn number (MHz)	1575 (1800 MHz)
UL EarFcn	Fixed	EarFcn number (MHz)	19575 (1800 MHz)
Ue phy :: TxPower	Fixed	dBm	23
Enb phy :: TxPower	Fixed	dBm	46
Height Antenna	Fixed	mt	30
Mac Scheduler	Fixed	FdMtFfMacSheduler, TdMtFfMacScheduler, PfFfMacScheduler, ...	PfFfMacScheduler
Propagation model	Fixed	Friis, Three-log distance, Nakagami, Two-ray ground, Jakes	Friis
Rrc model	Fixed	Ideal, real	Ideal
Rrc :: EpsBearerToRlcMapping	Fixed	RlcSmAlways, RlcUmAlways,	RlcUmAlways
Rrc :: SrsPeriodicity	Fixed	msec	320
Delay between eNB and server	Fixed	msec	50
Data rate between eNB and server	Fixed	Gb/s	100
MTU between eNB and server	Fixed	Bytes	1500

Table 15.4 reports the main metrics measured in the simulation-based analysis to evaluate the communication performance of the LTE network in different traffic and context conditions. To this aim, in each simulation run we measured Latency and Packet Error Rate (PER): Latency (in milliseconds) expresses the time required to transmit and receive a message, and PER expresses the percentage of corrupted packets among the ones that have been sent. By measuring both Latency and PER we were able to study how the two aspects of interest (type of communication protocol and content size) impact on LTE-based communication network.

15.5.2 On-road Testing Results

The vehicle-to-vehicle communication latency offered by the LTE-based service has been measured on the available test site in several measurement campaigns. The obtained results are quite homogenous on the three zones included in our testbed. As example, one of the results obtained on the Romagnano roundabout will be hereinafter presented. The latency values have been measured for about 25 minutes on the 500L prototype vehicle continuously moving on the area of interest. The obtained results are reported in Figure 15.9, where the latency of each packet is depicted in blue and the corresponding 10 seconds moving average in red. As can be seen, the 100 msec requirement is almost completely fulfilled. In fact, the overall mean latency value is equal to 59.7 msec (with a

Table 15.3: Main aspects investigated in the simulation

Investigated aspect	Value
Type of communication protocol	Unicast, broadcast
Content size (packet size)	Increasing packet size (512B, 1024B, 5000B)

Table 15.4: Output metrics collected in the simulation

Metrics	Units
Latency	msec
Packet error rate (PER)	%

standard deviation of 8.9 msec) and only 0.37% of the samples assume values greater than 100 msec. This behavior is probably due to the routing and resource allocation policies adopted by the LTE architecture, which can rapidly vary according to the instantaneous network conditions, especially in the control plane signaling. The number of lost packets in the considered time frame is equal to zero, thanks to the fact that only two equipped vehicles were using simultaneously the dedicated network service. This indicator is clearly expected to worsen when the number of vehicles increases significantly.

The link quality provided by the LTE network has been evaluated by measuring the three following quantities related to the signal strength: Reference Symbol Received Power (RSRP), Received Signal Strength Indicator (RSSI), and Reference Signal Received Quality (RSRQ). These measurements have been acquired on the LTE module mounted on the Fiat 500L prototype. According to (8), RSRP can be defined as the linear average of the cell-specific reference signals across the channel bandwidth. RSSI represents the average total received wide band power, including thermal noise, co-channel power and noise generated in the receiver. RSRQ combines RSRP and RSSI with the information related to the number of used resource blocks in order to obtain an indication of the quality of the received reference signal. The measurements of these three quantities (i.e., RSRP, RSSI, and RSRQ) acquired on the Romagnano roundabout are reported in Figure 15.10, Figure 15.11, and Figure 15.12, respectively.
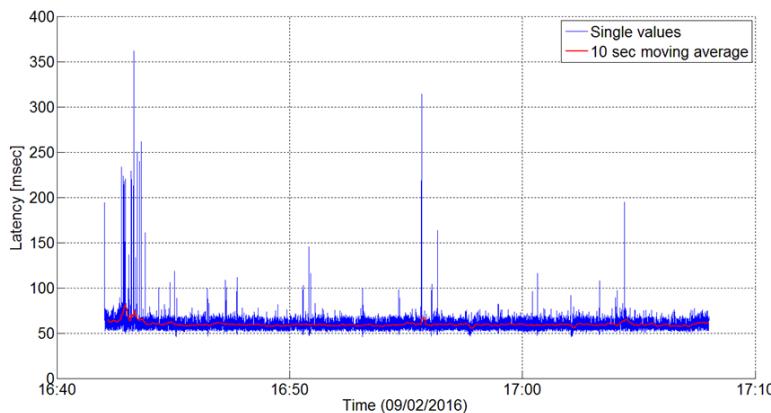

Figure 15.9: Latency measured on the Romagnano roundabout
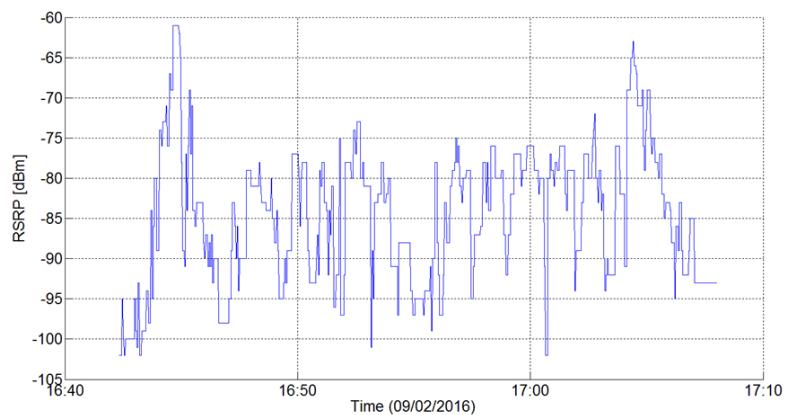
Figure 15.10: RSRP measured on the Romagnano roundabout
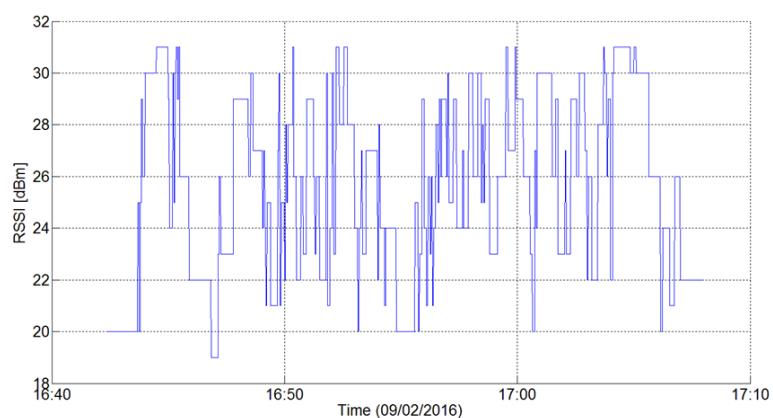
Figure 15.11: RSSI measured on the Romagnano roundabout
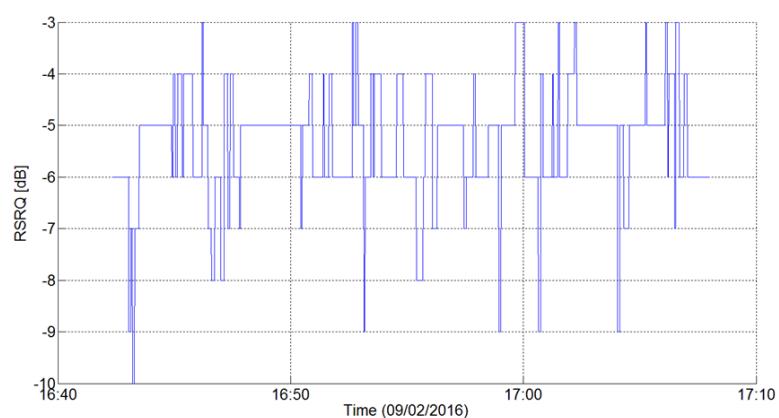
Figure 15.12: RSRQ measured on the Romagnano roundabout

15.5.3 Simulation Results

Table 15.5 summarizes the parameters we combined for the simulation runs. In the rest of this section we report and discuss the results achieved in these simulations.

Table 15.5: Summary of the parameters adopted in the simulation runs

Vehicles	Packet Size	PathLoss Model	MAC Scheduler	UL/DL Bandwidth	DL Ear-Fcn	UL Ear-Fcn	Delay
5, 10, 30, 40, 50, 100, 150, 200	512 B, 1500 B, 5000 B	Friis	PfFf Mac-Scheduler	100 / 100	1575 (1800 MHz)	19575 (1800 MHz)	50 msec

Type of communication protocol

Figure 15.13 plots Latency and PER obtained in the simulation conducted for comparing the communication performance achieved by adopting unicast and broadcast communication protocols. The plots in the figure show that Latency in the unicast case is varying in the range between (about) 68 and (about) 296 msec, while between (about) 68 and (about) 183 msec in the broadcast case, if we consider to send a limited amount of content (small packet size corresponding to 512 B). Both solutions have a similar behavior up to about 50 vehicles while the difference increases if we consider more vehicles. In terms of PER, the performance of the two communication protocols is comparable if we consider less than 40 vehicles. If more vehicles are involved, the performance of the unicast solution gets notably worse while the behavior of the broadcast one remains almost acceptable.

The obtained results are completely in line with similar studies already published in the literature (see, e.g., (1), (15), and (18)). The added value provided by this work is the possibility to fine tune the simulation parameters following the measurements gathered on the available testbed. In this way, it is possible to increase the quality and accuracy of the results obtained in more complex scenarios, which could be not easily tested in real conditions due to cost and practical constraints (e.g. it'd be difficult to make extensive measurements on a real cross-road due to permissions and limited control on traffic flows). For example, the scenario with 2 vehicles considered in our activity could appear not completely realistic in terms of number of involved actors (i.e., the probability to find only 2 vehicles in the selected area is quite low, if we assume an acceptable penetration rate for our ADAS solution). However, it provides relevant indications on the performance of the communication component of the system (in terms of channel characteristics, fading, packet-level performance of LTE) as well as the system as a whole (i.e. from physical to application and mobility levels). Based on such measures it is possible to "fine-tune" the simulation environment and consider more interesting scenarios, where the number of involved vehicles is allowed to increase to hundreds and scenarios can become arbitrarily complex.

Content size

Figure 15.14 and Figure 15.15 plot Latency and PER obtained in the simulations conducted for comparing the communication performance achieved if content of different size is sent, for both unicast and broadcast communication schemes, respectively. In these runs, we considered a frequency band of 1800 MHz while we changed the packet size of the sent content. As expected, the plots of the collected results confirm that an increase of the content size till 5000 B decreases the performances.
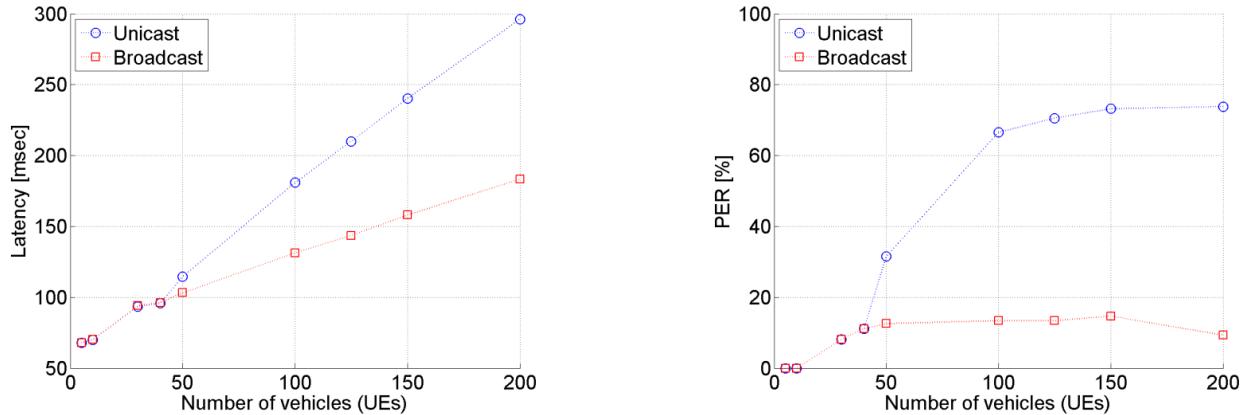
Figure 15.13: Latency and PER for unicast and broadcast communication protocol
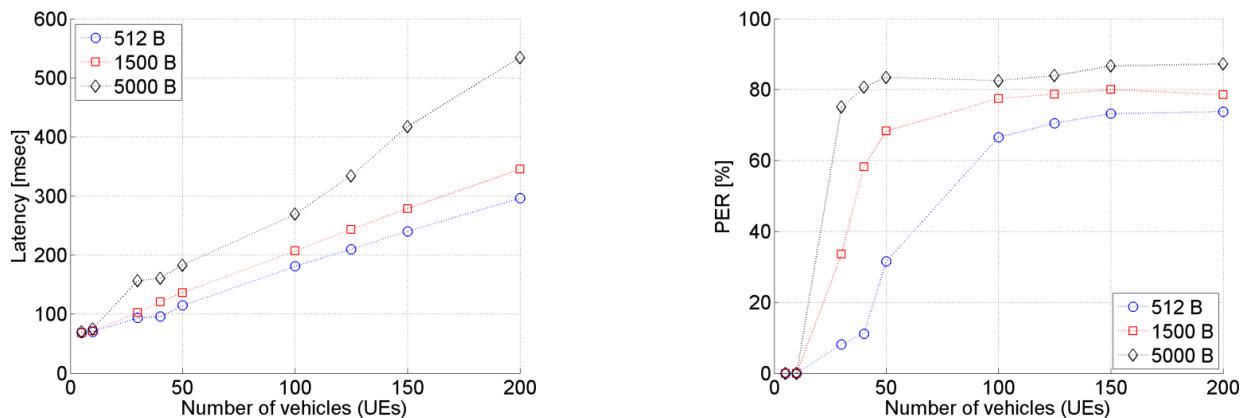
Figure 15.14: Latency and PER for different content size (unicast scheme)

We can also see that Latency obtained by sending 512 B and 1500 B seems comparable, instead PER decreases notably when we increase again the packet size. For instance, by considering the unicast communication protocol we see that for 50 vehicles PER is (about) 84% for 5000 B, (about) 68% for 1500 B, and (about) 31% for 512 B. Conversely, the maximum PER for 50 vehicles is (about) 40% if we adopt the broadcast communication protocol and we send 5000 B.

15.6 Conclusions

The paper describes the experience gained through the e2Call research project for analyzing the performance of cellular-based in-vehicle applications. Among the three innovative solutions previously proposed in the project (i.e., ADAS, e2Call, and RSIS), the next-generation advanced driver assistance system has been presented and analyzed as a case study. Such application uses the enhanced communication capabilities offered by the LTE network for exchanging data related to the vehicles' dynamic and positioning. This information is then used by each vehicle for evaluating the probability to collide with the other surrounding ones and take safety decision accordingly. The presented methodology proposes a joint exploitation of the results obtained in the real testbed and the ones collected through the simulation platform based on SUMO and NS3. The measurements collected with two equipped vehicles circulating on real roads and connected to the commercial LTE network have been used for fine tuning the simulation parameters. This process enabled the possibility to
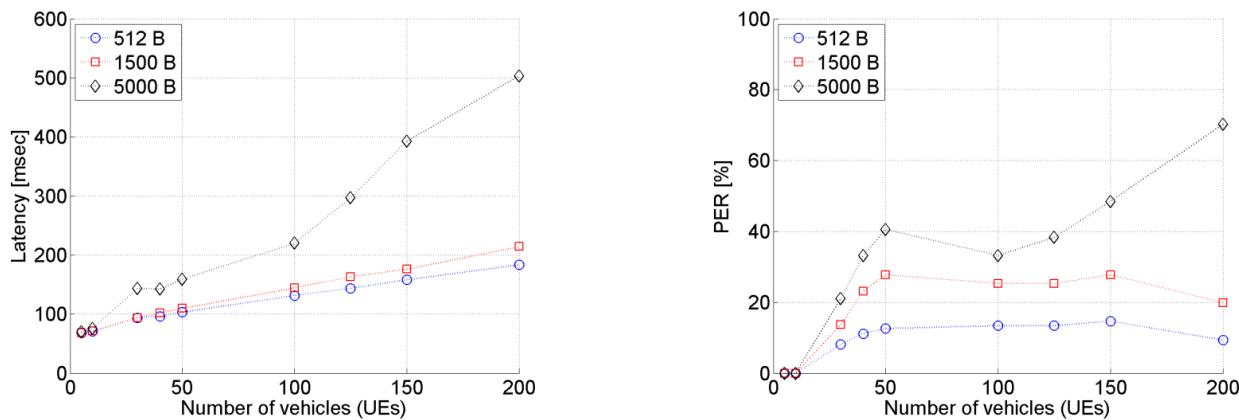
Figure 15.15: Latency and PER for different content size (broadcast scheme)

obtain more accurate and realistic results, as simulation is employed to scale-up the considered scenario and to analyze the performance of the system with more degrees of freedom. This approach will be further extended by considering some additional metrics to be measured on the testbed and then used for parameterizing the simulation platform. As example, the channel measures acquired on the testing area could be included into the simulator for enhancing the employed channel model. Moreover, vehicular traces acquired on the field through satellite-based positioning system could be analyzed for better modeling the vehicle behavior of the vehicles in the SUMO platform. The same traces could be also used for creating customized maps of the area of interest. The obtained result will be characterized by a very high precision with respect to the freely available maps and, once imported into the simulation platform, will contribute in increasing its overall accuracy without affecting the complexity level.

Acknowledgements

15.7 References

[1] A. VINEL: *3GPP LTE Versus IEEE 802.11p/WAVE: Which Technology is Able to Support Cooperative Vehicular Safety Applications?* IEEE Wireless Communications Letters, pages 125–128, April 2012.

[2] B. NGUYEN, A. BANERJEE, V. GOPALAKRISHNAN, S. KASERA, S. LEE, A. SHAIKH AND J., VAN DER MERWE: *Towards Understanding TCP Performance on LTE/EPC Mobile Networks*. In *ACM Workshop on All things cellular: operations, applications, and challenges*, USA, 2014.

[3] CRF: *https://www.crf.it*. last access: April 2016.

[4] D. KRAJZEWICZ, J. ERDMANN, M. BEHRISCH AND L. BIEKER: *Recent Development and Applications of SUMO - Simulation of Urban MObility*. International Journal On Advances in Systems and Measurements, 5(3&4):128–138, December 2012.

[5] Department of Information Engineering and Computer Science, University of Trento, Italy: *http://www.disi.unitn.it*. last access: April 2016.

[6] E2Call project - Autonomous Province of Trento (TN) - Italy: *http://www.e2call.eu*. last access: April 2016.

[7] ETSI ES 202 663: *Intelligent Transport Systems (ITS): European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band*. ETSI, 2009.

[8] ETSI TS 136 214: *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer - Measurements (3GPP TS 36.214 version 9.1.0 Release 9)*. ETSI, 2010.

[9] European project iHeERO - Infrastructure Harmonised eCall European Pilot: *http://iheero.eu*. last access: April 2016.

[10] G. Percannella: *e2Call: eCall enhancement based on 4G and inference algorithms*. In *European Congress and Exhibition on Intelligent Transport Systems and Services*, Finland, 2014.

[11] Here Maps: *http://www.here.com*. last access: April 2016.

[12] ISTAT: *Road accidents in Italy, Year 2014*. http://www.istat.it/en/archive/172487 last access: April 2016.

[13] K. Trichias: *Modeling and Evaluation of LTE in Intelligent Transportation Systems*. In *Master of Science Thesis - Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS), University of Twente and TNO*, The Netherlands, 2011.

[14] M. Da Lio, F. Biral, E. Bertolazzi, M. Galvani, P. Bosetti, D. Windridge, A. Saroldi and F. Tango: *Artificial Co-Drivers as a Universal Enabling Technology for Future Intelligent Vehicles and Transportation Systems*. IEEE Transactions on Intelligent Transportation Systems, pages 244–263, Feb 2015.

[15] M. Phan, R. Rembarz and S. Sories: *A Capacity Analysis for the Transmission of Event and Cooperative Awareness Messages in LTE Networks*. In *ITS World Congress*, USA, 2011.

[16] NS3: *https://www.nsnam.org*. last access: April 2016.

[17] Open Street Map: *http://www.openstreetmap.org*. last access: April 2016.

[18] S. Kato, M. Hiltunen, K. Joshi and R. Schlichting: *Enabling vehicular safety applications over LTE networks*. In *IEEE International Conference on Connected Vehicles and Expo*, USA, 2013.

[19] Tim: *http://www.tim.it*. last access: April 2016.

16 Local Emissions Monitoring using vehicular communication

Laura Bieker, Daniel Krajzewicz and Andreas Leich; German Aerospace Center
{Laura.Bieker, Daniel.Krajzewicz, Andreas.Leich}@DLR.de
Martin Dippold; Technical University Graz
{dippold@ivt.tugraz.at}

Abstract

The emission of pollutants is one of the most significant adverse effects of road transport. Legal restrictions and traffic management strategies can help to reduce emissions. However, to identify actions which have a positive effect on the environment the local emissions have to be measured or estimated. The infrastructure to measure pollutant emissions is difficult to set up and expensive. Therefore this paper presents an approach to estimate local emissions. The basic idea is to combine data from vehicular communication (V2X) with the information of induction loops. The simulation results show that the emissions can be estimated with an error of less than 10% for a V2X equipment rate of 1% within the regarded scenario.
Local Emission Monitoring, pollution estimation, connected vehicles, traffic simulation

16.1 Introduction

Transportation takes a major role in the production of pollutant and noise emissions (3). In order to reduce vehicle emissions it is necessary to measure or estimate the real emissions (10). The most common method to monitor pollutant emissions from road traffic is air quality measurement performed at road side locations. This allows for an assessment of the contribution of vehicle emissions when the measured air concentrations are compared with concentrations from a representative background measurement (e.g. for a city area: a close to town off-traffic site). This method is affected with several uncertainties, as

- for several important pollutants (e.g. PM10, CO2), the background concentrations are close to the roadside values;

- air quality measurement is always affected by meteorological boundary conditions (wind conditions, precipitation etc.).

Another method to investigate vehicle emissions in on-road conditions is to use portable emission measurement systems (PEMS) which are directly attached to the tail pipe (6). Vehicles equipped with PEMS have to pass a certain road section several times in each scenario (e.g. in each traffic light control variation or each speed limit) and the measured emissions can then be compared. However, the use of PEMS is costly and in general only few systems are available (usually a single system is owned by a lab). So only a small sample of vehicles can be monitored, which results in uncertainties in the representativeness of the recorded driving behavior and even more of the recorded emission results. The vehicle fleet consists of various different propulsion concepts (e.g. gasoline or diesel

engines, different exhaust gas after treatment systems) which show different sensitivities of emission behavior to particular driving conditions.

Measuring emission in a real test field is time consuming and difficult to set up. Therefore there are other research studies which focus on other ways how emission can be estimated. The research study of (1) compared emission estimations with the measured air quality. The authors state that a detailed knowledge of the vehicle fleet and the speed profiles is needed to proper estimate the local emissions. In (4) an approach for traffic monitoring and the computation of vehicle emissions by Floating Car Data (FCD) is presented. FCD are usually taxis (or other vehicles) which are equipped with an FCD device to send the current GPS (Global Positioning System) position. From FCD the current travel times for the taxi routes can be estimated. In (4) the travel time is used to estimate the current traffic state to compute the estimated emissions.

The advantage of vehicle-to-vehicle or vehicle-to-infrastructure (V2X) communication is that a speed profile for every equipped vehicle is generated that can be directly used for the estimation of the emissions. The problem is only that it will take a long time until all vehicles will be equipped with V2X communication. To overcome this problem this paper presents an approach which combines the detailed speed information from V2X communication with the data from induction loops to estimate the local vehicle emissions.

16.2 Simulation Scenario

The local emission monitoring approach was tested using the traffic simulation SUMO (**?**). For the European Commission project COLOMBO an extension to SUMO for emission simulation was made which is called PHEMlight (5). PHEMlight is a simplification of the model PHEM (Passenger Car and Heavy Duty Emission Model) (9) and provides a simple though accurate simulation technique to derive modal emission quantities for given speeds and accelerations. To avoid high storage demand and long computation times the history driving trajectories (speed and acceleration) transient dynamic corrections, temperature influences on after-treatment-systems and the driver gear shift model to compute engine speeds were not included in the simulation. These functions from PHEM were replaced by generic functions for a single car which describes fleet average values. In combination with SUMO, PHEMlight uses the vehicle speed and acceleration data from SUMO to compute the fuel consumption and emission for each vehicle.

The simulation evaluation was performed with different simulation scenarios which are described in the following. For the investigation of the different approaches a simulation scenario of a single four legged intersection was simulated (see Figure 16.1). In each direction the same static volume of vehicles drive straight forward, turn left and turn right (each 2,000 vehicles). Each monitoring approach was simulated 10 times spanning 7 hours. The emissions were calculated every 5 minutes. The simulation was run with different penetration rates of equipped vehicles: 1%, 10%, 20%, and 50%.

16.3 Local Emission Monitoring Approach

The basic idea for local emission monitoring is to use the V2X data of equipped vehicles and additionally the measurements from conventional traffic detectors. In the early adoption phase of V2X communication there will probably be only a low number of vehicles with V2X communication equipment. Therefore, it will be difficult to estimate the total local emissions of an intersection by taking the data from V2X communication only into account. Additionally, the count and optionally vehicle classification data from inductive loops can help to get a picture of the whole traffic situation. While inductive loops are widely used for calculating traffic flows and adapting traffic light phases, inductive loops alone cannot be used for emissions estimations, because they only measure the
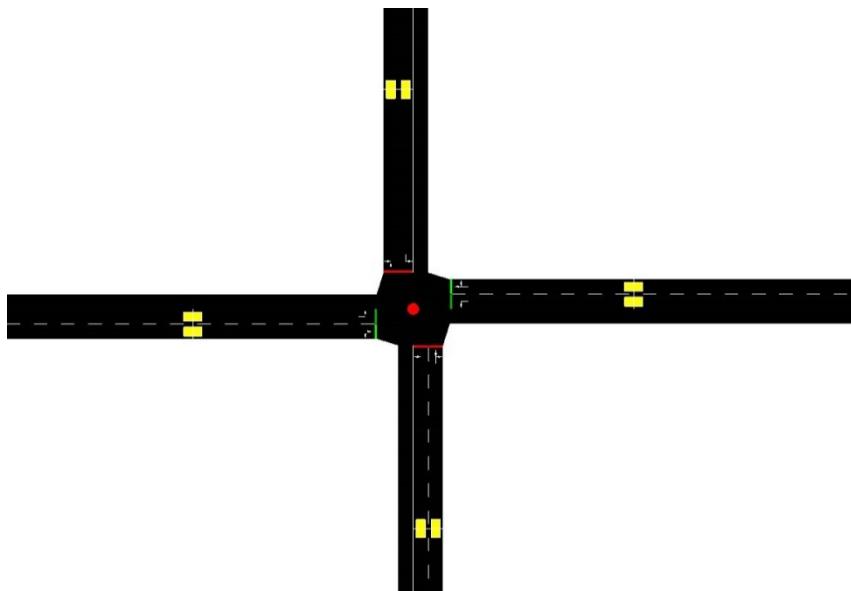
Figure 16.1: Basic intersection in SUMO (yellow: induction loops, red: Road side unit)

number of vehicles at one location. For the emission calculation only one intersection will be monitored. For further research the basic approach can be extended to monitor larger regions as well.

16.3.1 Assumptions

There are some general assumptions for all analyzed approaches which are described here:

- Inductive loops: every incoming lane of the intersections has its own inductive loop detector which determines the vehicle counts and the vehicle types, but not necessarily the speed.

- Intersections: At each monitored intersection a road side unit (RSU) collects the V2X data. For the investigation (the antenna of) the road side unit is placed in the middle of the intersection to provide a good reception range within all approaching lanes of the intersection.

- CAM definition: equipped vehicles are broadcasting so-called "Cooperative Awareness Messages" (CAMs) as definited by ETSI (2). The following data fields from CAMs are used for this research:

  - station characteristics (in this case private vehicle)
  - vehicle type
  - vehicle speed, vehicle speed confidence
  - curvature, curvature confidence
  - signed acceleration in direction of the node heading
  - position confidence ellipse

- V2X vehicle positioning (GPS) is not accurate enough to determine when it is passing the loop or on which lane the vehicle is driving on.

- The local emission monitoring should also work with low penetration ranges.

16.3.2 Basic Approach

The simplest approach analyzed here is called "Basic Approach" in the following. The CAMs of all equipped vehicles are collected in the RSU. Using the speed information from the CAMs, a time-speed series can be computed. The emissions of the equipped vehicle can be calculated by feeding this time-speed series into the emission model PHEMlight which runs as an application inside the RSU. The total number of vehicles which have passed an inductive loop detector at this intersection within a certain time period is counted. From these data the total number of vehicles, the percentage of equipped vehicles, and the emissions of the equipped vehicles are known. Hence, the estimated emissions for all vehicles can be calculated for this:

$$E_{total} = \frac{E_{V2X}}{q_{V2X}} * q_{total} \qquad (16.1)$$

where:

E_{total} : estimated total emissions

E_{V2X}: emissions of equipped vehicles

q_{V2X}: number of equipped vehicles

q_{total}: total number of vehicles

This approach can be refined by distinguishing between the different vehicle types/classes which can be detected by the loops.

$$E_{total=} \sum_{vtype} \frac{E_{V2X}^{vtype}}{q_{V2X}^{vtype}} * q_{total}^{vtype} \qquad (16.2)$$

This basic approach is very simple and does not take into consideration the differentiation of driving patterns. At some intersections it might be that vehicles which have to turn left have to wait longer to pass the intersection because of opposing traffic or there could be a traffic jam at one street of the intersection and vehicles at this street produce more emissions etc.

16.3.3 Clustering Approach

To overcome the uncertainties of the basic approach the hypothesis is introduced that vehicles with a similar speed profile (and similar emission class) produce similar emissions. The basic approach was extended by a clustering of the speed-time series of the equipped vehicles. The road side unit stores the computed speed-time series of every equipped vehicle in the communication range. They are clustered using a Python library called "fastcluster" which provides a hierarchical clustering algorithm (7). Wards method was used as method for the inter-cluster distance. A definition of implementation of the "ward" method can be found in (8). One problem with the clustering algorithm was that the speed-time series had to have the same length of measured data values. Otherwise it was not possible to cluster the speed-time series. To solve this problem the maximum length of all speed-time series was determined. All speed-time series with less data points were enlarged by appending zero-values as representation that the vehicle was not driving anymore in the communication range and as a result is not producing any emissions. This way of solving the problem would be not a good solution for matching the driving profile but for emission estimations it was the simplest way. As a first step the clustering algorithm was tested with the speed-time series of a single simulation run with a penetration rate of 1 %. Afterwards, the clustering algorithm was executed with the speed-time series of the simulation with different sizes of clusters (from 2 to 49 clusters). As an example the different speed-time series with 6 clusters are presented in figure 16.2. It can be seen that the clustering delivers classes of similar behaviour. It seems that a major indicator for the cluster is the actual time the vehicles spent within the communication range. Only the speed-time

series from cluster number 6 are not very similar, hence it is a "junk cluster". The idea was that speed-time series within the same cluster should have similar emissions. Therefore the emissions of the clusters were also analysed. In figure 16.3 the emissions can be seen for 6 clusters. The figures show that the emissions of the speed-time series in the same cluster are also similar, only for one cluster (cluster number 6 in the example) the emissions have a large spreading. This behavior can be seen in all analysed number of clusters. Even for 49 clusters there is one one inhomeneuous cluster. Figure 16.3 shows on the right side the average standard deviation of the emission clusters. It can be seen that for a small number of clusters the standard deviation is very high compared to a large number of clusters. Therefore it is recommended to use this approach with a larger number of clusters to receive good results. Furthermore, it can be seen that between 25 and 50 clusters the average standard deviation is changing only slightly. Thus, for this intersection it can be assumed that a cluster size of 25 should be detailed enough for emission evaluations.

16.4 Simulation Results

The ground truths of the "real" produced pollutant emission were collected to compare them with the estimated emissions "real" means the produced pollutant emissions for all vehicles in the simulation. For computing all emissions PHEMlight is used. For each simulation scenario the relative error of the estimated emissions was calculated as followed:

$$Error = \frac{(E_{approx} - E_{real})}{E_{real}} \tag{16.3}$$

Where:
$Error$: relative error
E_{approx} : approximated emissions
E_{real} : real emissions

16.4.1 Basic Approach

The simulation results show that even with a low penetration rate of 1 % the relative error of the estimated emissions is around 5 % after 1 hour of simulation and emission collecting time (see figure 16.4). The relative error is decreasing with higher penetration rates, this effect can be expected because with a higher penetration rate the knowledge about the produced emissions is also larger. It has to be mentioned that the used simulation scenario is very simple so with more complex traffic conditions the error will probably be higher. Therefore a more realistic simualtion scenario was set up. The traffic infrasture was the same only the vehicle types are adapted to a vehicle fleet prognosis of the year 2040. The year 2040 was choosen because it was expected that in 2040 there will be a realistic amount of equipped vehicles driving on the streets. The simulation was performed also with a vehicle fleet distribution of the years 2020 and 2030 and the results look similar to the one presented in figure 16.4 which shows the ones for the year 2040.

16.4.2 Clustering Approach

For the local emission monitoring the emission clusters are mapped onto the induction loops or street of the network. According to the assumption that the GPS positions of the CAMs are not accurate enough to know the exact lane and loop a vehicle is driving at, this mapping distinguishes only between the approaching streets, but not the lanes. The errors for the emission calculation with the clustering approach can be seen in figure 16.5. The errors are slightly higher than the results for the basic approach but are still relatively good. When comparing the result for the simulation
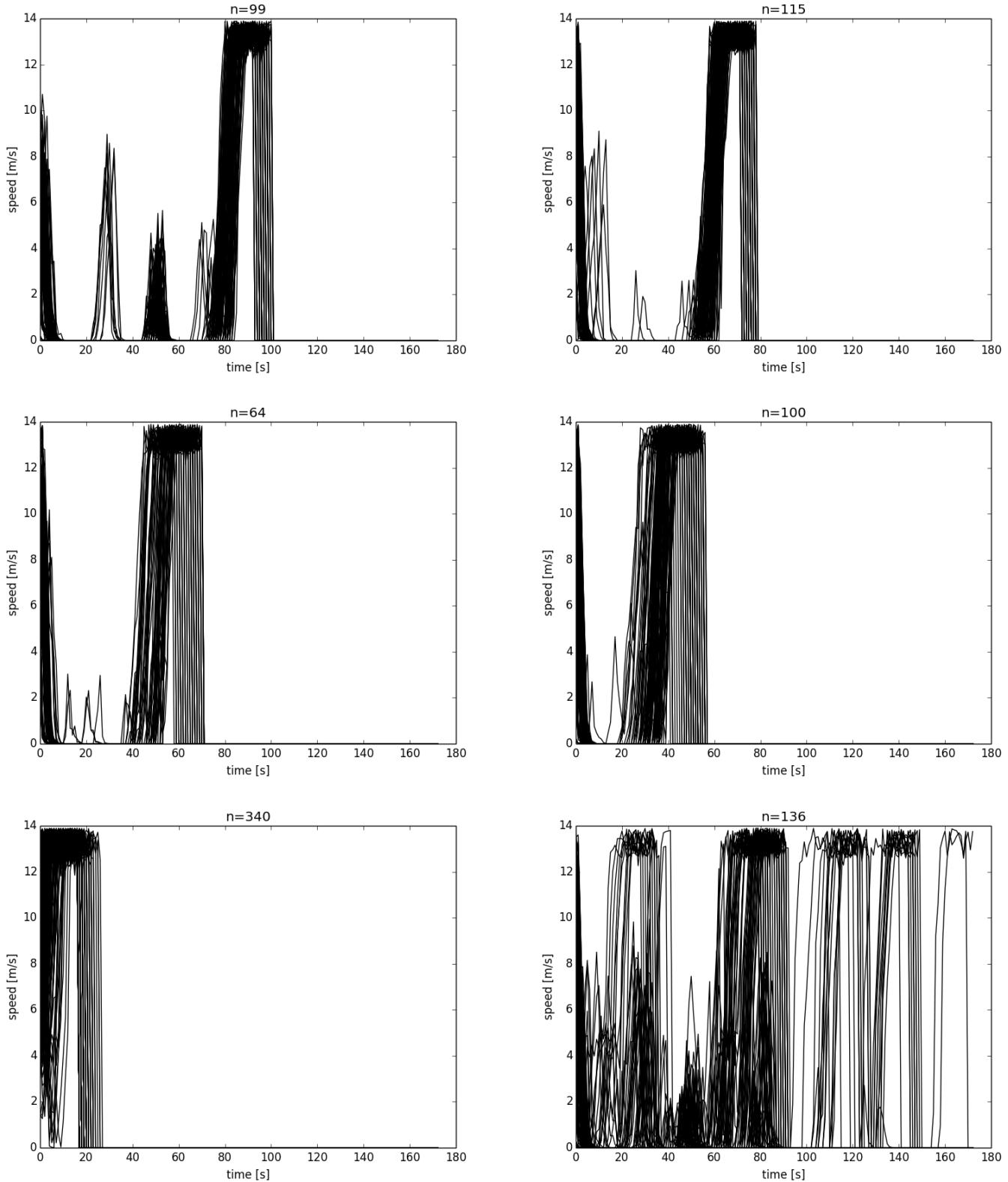
Figure 16.2: Trajectories within 6 clusters
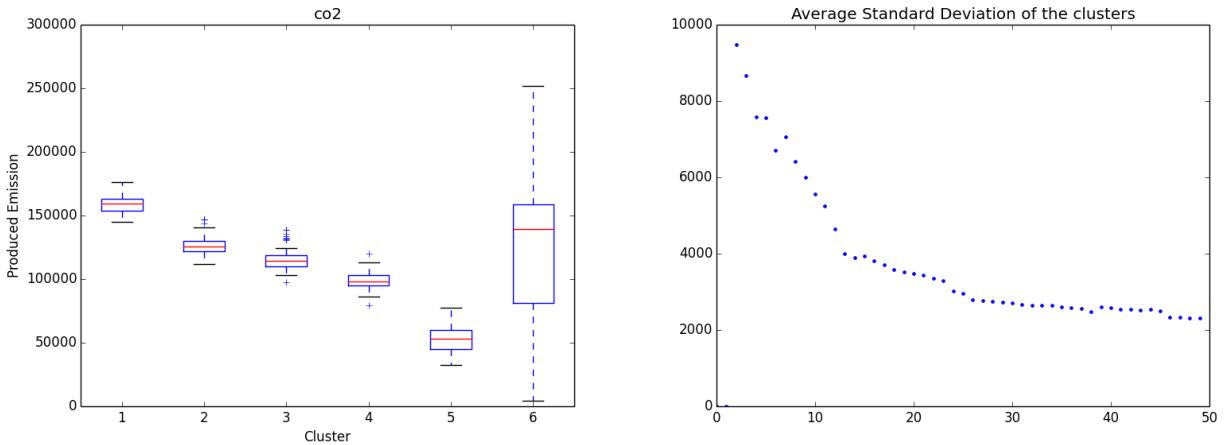
Figure 16.3: Relative Error of the estimated emissions over time (left) Average standard deviation of the emissions for adifferent number of clusters (from 2 to 49) (right)
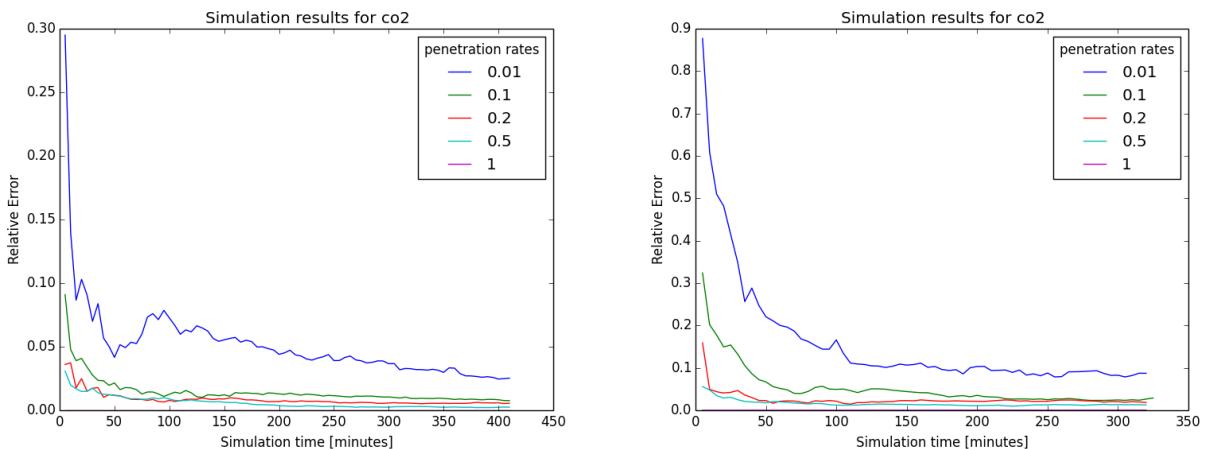

Figure 16.4: Relative Error of the estimated emissions over time (left). With different vehicle types (right)

with different vehicle types the clustering approach performs much better than the basic approach (see figure 16.5). For each penetration rates the error is under 10 %.

16.5 Conclusions

In this work an approach that combines information from inductive loops and vehicular communication to monitor local emissions was presented and evaluated. The basic idea of the approach was to use the vehicle counts from induction loops and combine this information with the speed profile of equipped vehicles. The simulation results show that the algorithm can be used for local emission monitoring. Even with low penetration rates of 1 % an error of less than 10 % can be reached.

But the current state of research has some limitations. The simulation scenario which was used for the evaluation is an artificial one. To have a more realistic traffic situation the approach should also be simulated for a intersection with real world data (an existing intersection and its traffic demand). For this evaluation only simulation results have been considered. Simulation models are normally a simplification of a real world situation. Therefore it can be assumed that the simulated emissions will have a larger error when comparing to measured emissions in real world. Hence, the results of the algorithm should also be compared to real world measurements.
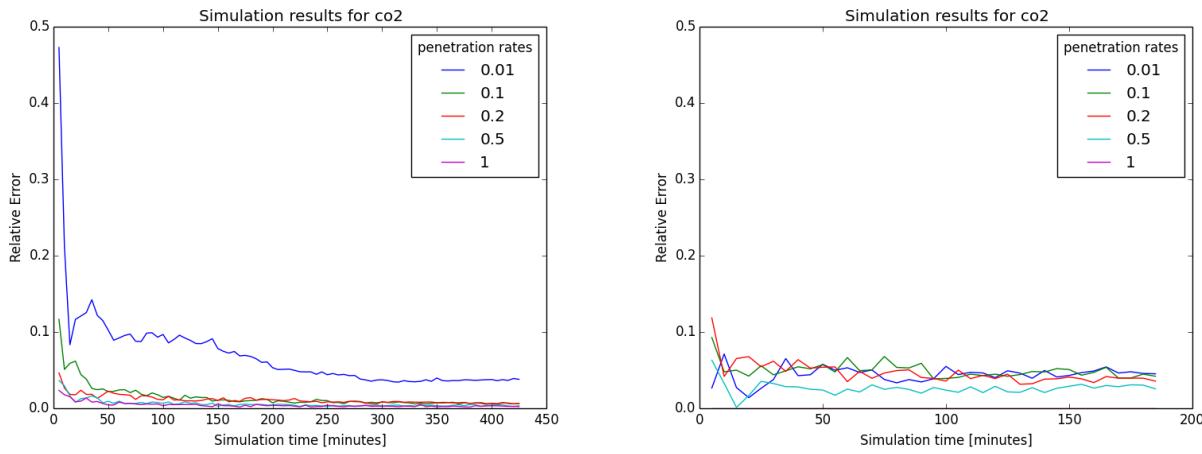
Figure 16.5: Relative Error of the estimated emissions over time for the clustering algorithm (left).
With different vehicle types (right).

The described approaches was evaluated for monitoring a single intersection, but further research has to be done if the approach should be extended for larger regions. One solution could be that CAMs are collected inside the vehicle over a longer time and all the information is send via GSM or when a road side unit is available. In this case it is not needed that every intersection is equipped with a road side unit for local emission monitoring, but the additional costs for GSM had to be considered.

Acknowledgments

16.6 References

[1] Bigazzi, A., J.W.C. Van Lint, G. Klunder, U. Stelwagen and N. Ligterink: *Traffic data for local emissions monitoring at a signalized intersection*. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 210–215, Sept 2010.

[2] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, 2011.

[3] Fuglestvedt, Jan, Terje Berntsen, Gunnar Myhre, Kristin Rypdal and Ragnhild Bieltvedt Skeie: *Climate forcing from the transport sectors*. Proceedings of the National Academy of Sciences, 105(2):454–458, 2008.

[4] Gühnemann, Astrid, Ralf-Peter Schäfer, Kai-Uwe Thiessenhusen and Peter Wagner: *Monitoring Traffic and Emissions by Floating Car Data*. Technical Report, 2004.

[5] Krajzewicz, Daniel, Marek Heinrich, Michela Milano, Paolo Bellavista, Thomas Stützle, Jérôme Härri, Thrasyvoulos Spyropoulos, Robbin Blokpoel, Stefan Hausberger and Martin Fellendorf: *COLOMBO: investigating*

the potential of V2X for traffic management purposes assuming low penetration rates. ITS Europe, 2013.

[6] KRASENBRINK, L. RUBINO; P. BONNEL; M. CARRIERO; A.: *Portable Emission Measurement System (PEMS) For Heavy Duty Diesel Vehicle PM Measurement: The European PM PEMS Program.* SAE International Journal of Engines, 118(3):17, 2009. http://papers.sae.org/2009-24-0149/.

[7] MÜLLNER, DANIEL: *fastcluster: Fast hierarchical clustering routines for R and Python.*

[8] MÜLLNER, DANIEL: *Fast Hierarchical, Agglomerative Clustering Routines for R and Python.* Journal of Statistical Software, 53(9):1pp, 2013. http://www.jstatsoft.org/v53/i09/.

[9] R., LUZ and HAUSBERGER S.: *User Guide for the Model PHEM, Version 11.2*, 2013.

[10] RADL, ULRICH: *Bewertungsmethodik der Verkehrssteuerung innerstädtischer Hauptverkehrsstraßen.* Straßenverkehrstechnik, 59:484–486, July 2015.