SUMO User Conference 2020
https://doi.org/10.52825/scp.v1i.103
© Authors. This work is licensed under a Creative Commons Attribution 3.0 DE License
Published: 18 Jun. 2022

Pre-study and insights to a sequential MATSim-SUMO tool-coupling to deduce 24h driving profiles for SAEVs

Henriette Triebke[1,2], Markus Kromer[1], and Peter Vortisch[2]

[1] Corporate Reasearch, Robert Bosch GmbH, Renningen, Germany

henriette.triebke@de.bosch.com

[2] Institute for Transport Studies, KIT, Karlsruhe, Germany

peter.vortisch@kit.edu

Abstract

New mobility concepts such as shared, autonomous, electric vehicle (SAEV) fleets raise questions to the vehicles' technical design. Compared to privately owned human driven cars, SAEVs are expected to exhibit different load profiles that entail the need for newly dimensioned powertrain and battery components. Since vehicle architecture is very sensitive to operating characteristics, detailed SAEV driving cycles are crucial for requirement engineering. As real world measurements reach their limit with new mobility concepts, this contribution seeks to evaluate three different traffic simulation approaches in their ability to model detailed SAEV driving profiles. (i) The mesoscopic traffic simulation framework MATSim is analyzed as it is predestined for large-scale fleet simulation and allows the tracking of individual vehicles. (ii) To improve driving dynamics, MATSim's simplified velocity profiles are enhanced with real-world driving cycles. (iii) A sequential tool-coupling of MATSim with the microscopic traffic simulation tool SUMO is pursued. All three approaches are compared and evaluated by means of a comprehensive test case study. The simulation results are compared in terms of driving dynamics and energy related key performance indicators (KPI) and then benchmarked against real driving cycles. The sequential tool-coupling approach shows the greatest potential to generate reliable SAEV driving profiles.

1 Introduction

SAEV load profiles and technical requirements are expected to differ fundamentally from conventional private cars. While the latter feature (a) small daily mileages, (b) long times of non-use, (c) high driving ranges and (d) have access to a dense refueling infrastructure, SAEV operating characteristics are rather opposite when used for urban passenger transport. Higher daily mileages and shorter (battery-limited) driving ranges entail the need for frequent recharging. This, however, is counteracted economically by the request for little idling times and technically by long charging durations within a comparatively thin network of charging stations. As the complexity of vehicle development increases, detailed SAEV driving profiles become more and more important for virtual prototype testing. For this purpose, they need to meet the following **key requirements**: **(KR1)** The profiles need to mirror the vehicles' movement throughout

entire metropolitan areas for 24 hours accounting for all range and charging constraints as well as for different routing, dispatching and pricing strategies. **(KR2)** They need to provide information on the vehicles' states such as *idling*, *relocating*, *charging* or *occupied* to enable optimal climate control or battery preconditioning. **(KR3)** The driving cycles must be accurate enough to derive reasonable velocity profiles that reflect for autonomous driving, road congestion and diverse transport infrastructures. **(KR4)** Depending on the road network's topography or the driving cycle's purpose, further time-series such as altitude or occupancy profiles are also of interest. To the authors' best knowledge, the problem of deriving representative SAEV driving profiles that meet all above stated requirements has not been tackled by the scientific community yet. There are many publications that deal with conventional driving cycle generation, the modeling of autonomous driving behavior or large-scale SA(E)V fleet simulation. However, no holistic approach is known that combines all three areas.

For the automotive industry **driving cycles** play a major role in state-of-the-art emission modeling, performance prediction and virtual prototype testing. Driving cycles most commonly designate second-by-second time-velocity profiles and can be distinguished in *modal* and *transient* cycles. Modal cycles are highly simplified and consist of different idling, straight acceleration and steady speed phases. They often feature unrealistic dynamics in the transition zones [1, 9, 24]. Transient cycles in contrast, reflect real-life driving behavior under on-road conditions [12]. A common technique to derive new driving cycles comprises four steps: **route choice, data collection, data clustering and cycle generation** [1, 40, 43]. Route choice involves selecting the route on which data are to be collected. The driving data are gathered by means of *on-board measurement*, *GPS-tracking* and/or *chase car method*. As stated in [43], on-road measurements reflect the selected route most accurately but feature a strong bias due to unusual congestion pattern which entails the need for repetitive measurements. The chase car method is less cost-intensive and involves randomly following target vehicles by imitating their driving behavior. This approach, however, comes at the price of route choice. The collected profiles are often decomposed into micro trips[1] which are clustered according to traffic condition, vehicle type or other KPI. Common trip clustering techniques are *k-means cluster algorithms* [15, 41] or hybrid approaches of *k-means* and *support vector machine (SVM)* clustering [43]. Despite their validity, cluster methods often require large computational resources [1]. The final cycle is typically constructed from a pool of available micro trips [1, 40, 43]. The idea of the **micro-trip-based methodology** is to find those micro trips that reflect the diversity of real world driving well enough but in a more compressed manner to be practical and cost effective [1, 40]. Generally, the micro trips are selected by algorithms based on predefined performance measures. Alternatively, *Monte Carlo engines* serve to generate multiple candidate cycles by randomly picking several micro trips and determining their KPI. The best fit in performance is then finally chosen.

Another statistical approach consists in using real world driving databases to generate synthetic driving cycles by means of **Markov chain processes**. As done in [18, 35, 36], the measured velocity profiles serve to construct a transition probability matrix of a Markov chain. At this, each matrix element corresponds to a certain state (denoted by current velocity and acceleration) and within each state, the transition probabilities to jump from one state to another are stored. Yet another data-driven approach of driving cycle deduction is referred to as **route information mapping**. A new concept of defining automotive driving cycles is introduced in [12] by stressing the need to incorporate external conditions such as weather, traffic and terrain data. This is also done in [16] by joining data on slope, road curvature and speed

[1] A micro trip denotes a trip between two idling phases.

limit with traffic information and driver models to form a control problem that is numerically solved to generate velocity profiles. However, prerequisites for such data-driven approaches are (a) large databases of GPS-tracked driving cycles, (b) detailed maps and/or (c) access to traffic information.

As all previous methods rely on measured or historical data, they are not suited to deduce driving cycles for future autonomous vehicles. **Autonomous driving behavior** is often approached by applying **filter or smoothing techniques** on human driven profiles [2, 19, 28]. In this context, the smoothing approach is justified by *kinetosis prevention*[2] on the one hand and by the vehicle's improved perception on the other hand. Advanced sensors and car2x-communication will enable autonomous vehicles (AVs) to respond more smoothly to ambient traffic conditions. However, smoothing techniques tend to annihilate idling times and cannot reflect for platooning effects or connected driving in a methodologically sound manner.

As conventional approaches to deduce representative driving cycles reach their limit with new mobility concepts, **microscopic traffic simulation** became increasingly popular in this regard. Microscopic frameworks have been used for cost-optimized driving cycle deduction [1] and to assess the impact of automated driving on fuel consumption [10, 21, 37]. In [21], the capability of VISSIM[3] to model real world driving cycles is evaluated. Compared to human-driven cycles, the simulated profiles fit well in aerodynamic speed but poor in acceleration: human drivers tend to have higher acceleration rates at lower speeds and the simulation neglects stochastic oscillations around the target velocity. Similar conclusions are drawn in [1] which combines microscopic traffic simulation and micro-trip-based methods to deduce representative driving cycles. According to the authors, default parameters from micro-simulation produce **unrealistic driving behavior**: simulated velocity profiles are too aggressive as their gradients are often set to the vehicle's maximum capability. This is also evidenced in [37] by emphasizing that the driving cycles' quality is directly tied to a well calibrated traffic model. Due to the same reason, the relevance of microscopic traffic models for estimating the impact of traffic strategies on fuel consumption is questioned in [10]. The authors pinpoint the fact that **microscopic traffic simulation models have a validation problem** when driving dynamics are concerned: even though they produce detailed velocity profiles, microscopic traffic models are usually designed to meet macroscopic objectives such as signal timing or transportation planning. Consequently they are calibrated by traffic flow parameters like speed, density or queue length rather than instantaneous speed and acceleration [10, 37]. Thus, speed profiles are often too simplified and therefore might not be applicable for environmental studies or requirement engineering. However, even though microscopic traffic simulation tools have weaknesses in capturing human driving behavior, they are likely to cope well enough with fully automated driving as fewer stochastic terms are involved.

There is plenty of literature dealing with the **acceptance, simulation and impact of autonomous vehicle fleets**. For one thing, AV fleets are expected to improve network capacity due to connected driving and improved safety [34]. Then again, AVs may also increase traffic volumes due to induced travel demand arising from improved travel comfort, additional empty rides and smaller vessel sizes in contrast to public transport means [22]. Due to their disruptive character, AV fleet simulations have been analyzed from many different perspectives. In this context, especially the mesoscopic *Multi-Agent Transport Simulation framework*[4] (MATSim) [20]

[2] To ensure the passenger's well being, the lateral and longitudinal acceleration is limited.
[3] https://www.ptvgroup.com/de/loesungen/produkte/ptv-vissim/
[4] https://www.matsim.org/

is well established. In [4] and [14], for example, the city-wide replacement of private cars with shared autonomous vehicle (SAV) fleets is simulated for Berlin and Austin. Both studies conclude that each SAV could potentially replace ten privately owned cars. Further contributions evaluate the impact of different SAV pricing schemes on mode choice [23, 27] or deal with SAV electrification and its implication for charging infrastructure planning [5, 8, 29, 42]. The influence of routing and dispatching algorithms on taxi services are extensively discussed in [7, 31, 32]. However, even though MATSim has its strong points in large-scale fleet simulation, mesoscopic traffic simulation tools generally lack the necessary level of detail to simulate reasonable dynamics of individual vehicles [38].

To conclude, there are numerous publications dedicated to partial solutions but as those approaches are often too narrow in their objective, they either lose viability or lack feasibility in a broader context. This contribution seeks to elaborate an overall concept to deduce **representative 24h SAEV driving cycles that meet all above stated key requirements**. To this end, three different traffic simulation approaches are evaluated and discussed. To gain deeper insights in terms of large-scale feasibility, the methods are applied to a set of test cases. To reduce modeling effort, several simplifications are made: the pre-study has no fleet character yet, nor does it reflect for autonomous driving behavior. These limits, however, do not affect this study's validity: The main objective at this stage is to quantify the approaches' suitability by means of different evaluation criteria, such as (a) their ability to model detailed driving dynamics, (b) their capability to simulate large-scale areas and (c) the approaches' feasibility in terms of data availability and automation capacity **(KR5)**.

2 Methodological approach

This section serves to outline each of the three simulation approaches in more detail as their understanding is essential for the test case analysis in Section 3.

2.1 MATSim's capabilities and limits in drive cycle deduction

MATSim is a open-source framework for large-scale, agent-based traffic simulation. Its traffic assignment relies upon a co-evolutionary algorithm where so-called agents optimize their daily activity schedules in an iterative fashion by varying their initial departure time, transport mode or route choice to maximize their personal benefit. At this, they compete with other agents for space-time resources in the transportation network until a quasi equilibrium state is reached[5]. MATSim allows the deduction of vehicle trajectories and status profiles by design. Every action an agent performs – such as entering or leaving a certain road segment (*link*) – is recorded. Based on this information, daily status and speed profiles can easily be derived as exemplarily shown in Figure 1. However, as MATSim uses a simplified queue model to approximate traffic dynamics, the framework does not provide any reasonable information on a vehicle's position on a link itself. Only average link-speeds can be extracted. The queue model further leads to limitations in congestion modeling [3] as the tool's primary purpose is to simulate large scenarios in decent time which requires simplifications in traffic and driving dynamics. As the understanding of those shortcomings is essential for this work, a brief recap of MATSim's traffic dynamics is given next. MATSim relies on the discrete cell transmission model (CTM) [11] and the queuing model described in [17]. In the CTM, the length of the homogeneous network cells

[5]For more information on the user equilibrium, replanning process or plan scoring please refer to [20].

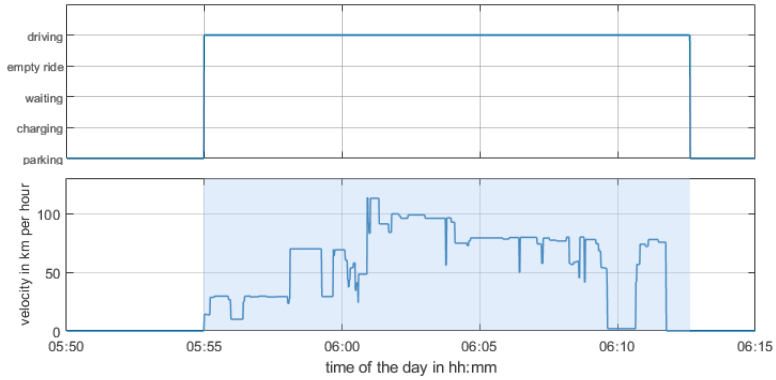

Figure 1: Status (top panel) and average link speed profile (bottom) for a chosen MATSim car

is defined by the distance a vehicle travels in the time step T at free-flow velocity. As defined in Equation 1, the number of vehicles $n_i(t+T)$ in a cell i depends on the number of cars $n_i(t)$ in that cell at the previous time step and the difference of inflowing and outflowing vehicles.

$$n_i(t+T) = n_i(t) + Y_{i-1}(t) - Y_i(t) \tag{1}$$

Here, the vehicles' movement $Y_{\mathrm{i}-1}$ into the cell i is limited by three restrictions as depicted in Equation 2, where the *flow capacity* Q_{i} represents the maximum number of vehicles allowed to enter a cell and the *storage capacity* N_{i} the cell's capacity to store vehicles.

$$Y_{i-1} = \min \begin{cases} n_{i-1} \\ Q_i \times T \\ N_i - n_i \end{cases} \quad \text{with } n, N, Q \in \mathbb{N} \tag{2}$$

With the improved queue model by [17], the road network is represented by so-called *links* of different length instead of homogeneous cells. Additionally, *priority queues* are introduced in MATSim that sort vehicles on a link according to their order of entrance or earliest exit time.

Under certain conditions MATSim's queue model leads to false congestion patterns and therefore misleading vehicle dynamics especially on short links or in sample runs[6]. The flow capacity basically acts like a batch system: A flow capacity of 600 cars/h means that only every sixth second a vehicle is allowed to leave a link. Otherwise the exit is blocked. Consequently, newly arriving vehicles queue up on the link and wait for their turn to leave which sometimes leads to unrealistic long passing times. Consider, for example, two subsequent vehicles on a 15 m link: even with a free flow velocity of 50 km/h the rear car would need at least 6 s to pass 15 m as the exit is blocked this long by the first vehicle. The *stucktime parameter*[7] complicates this even further as it temporarily allows a car surplus on a link: 10 % sample runs reveal vehicle

[6]Sample runs increase computational performance, as only a subset of agents is simulated. In a 10 % run for example, each simulated vehicle gets the weight of ten and therefore occupies a net-space of 75 m on the network (the default vehicle length in a 100 % sample is 7.5 m) [20]. To preserve traffic dynamics, the flow and storage capacities are adjusted accordingly and multiplied by a factor $f_{\mathrm{f,s}} = 0.1$.

[7]To counteract gridlocks, the *stucktime parameter* has been introduced to bypass the storage capacity constraint in case the first vehicle in the queue is stuck too long. In doing so, a minimal flow even under very congested traffic conditions is maintained [20].

queues of 300 m length on a single link 10 m long. At this, the second vehicle needs at least 1 min to pass the link, the third a minimum of 2 min and the third even 3 min[8]. Technically, even four vehicles of weight 100 (which sum up to a queue of 3 km) can be enforced to stand on a single short link without throwing an error. It has to be stressed at this point, that under those circumstances the queues do *not* line up on upstream links, which hinders MATSim to model spatial congestion patterns in detail (even though they might be correct on a pure temporal level as the flow capacity has it's methodical legitimacy).
To conclude, short links act as temporary vehicle sinks, storing too many vehicles which otherwise would have spilled back in upstream links. Consequently, the average link-speed profiles are faulty under congested traffic conditions as they often show average link speeds near zero on short links but nearly free flow velocities on links prior to those error-prone short links.

2.2 MATSim drive cycle enrichment with real-world driving profiles

To improve driving dynamics, MATSim's average link speed profiles are enhanced with synthetic and real-world driving cycles. For this, five different driving cycles are chosen that mirror a wide range of driving maneuvers and road types. All together, they account for a total driving time of 228 min. The cycles' normalized velocity and acceleration distributions are given in

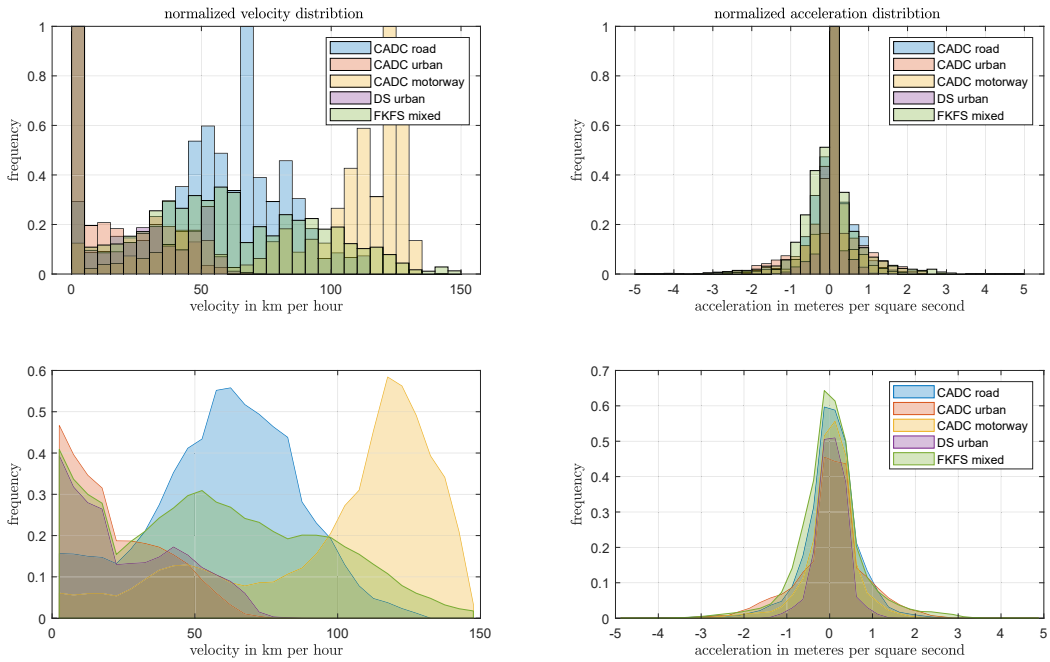

Figure 2: Normalized velocity (left) and acceleration distribution (right) of the considered drive cycles. All distributions are normalized to their local maximum and flattened by moving average.

[8]Given a nominal flow capacity of 600 cars/h which corresponds to 60 cars/h in a 10% sample run.
[9]Common Artemis Driving Cycles (CADC): https://dieselnet.com/standards/cycles/artemis.php
[10]The *DS urban* is a RB-internal cycle through Stuttgart city used for load collective deduction.
[11]The FKFS cycle was conceived by the *Research Institute of Automotive Engineering and Vehicle Engines Stuttgart* as representative driving cycle for the Stuttgart region.

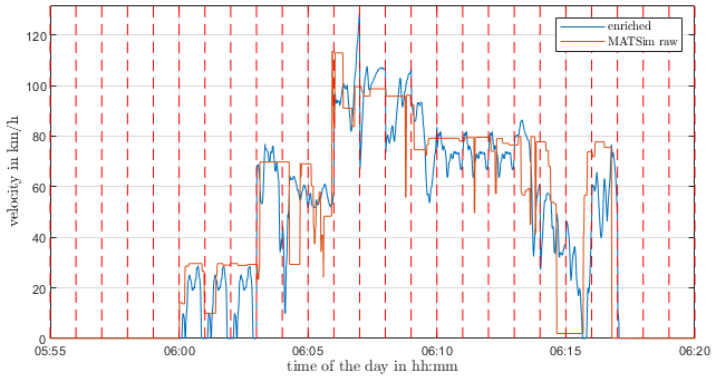

Figure 3: Exemplary representation of an enriched velocity profile without improvement measures. The red dashed lines indicate the transition of two consecutive 1 min-segments.

approach, an algorithm goes through all MATSim segments and identifies the CADC/DS/FKFS segment with the lowest discrepancy in average speed without bothering about unrealistic driving dynamics in the transition zones. In case the maximum speed limit of the MATSim segment is lower than the corresponding tabulated one, the segment with the next best fit in average speed is chosen. This prevents congested motorway cycles from being mixed into urban MATSim profiles. Figure 3 displays the outcome of this approach. At this, the orange and blue line represent MATSim's simplified and enriched profile respectively. As expected, the latter looks more realistic, but still features unrealistic acceleration rates between consecutive segments that require further improvement: (i) As discussed in Section 2.1, MATSim often features velocities near zero on short links. As those are hard to match with real driving cycles, the average speed of those 1 min-segments is set to zero if $\overline{v}_{seg} < 0.5$ km/h. (ii) To make up for the lost distance, the chosen synthetic driving cycles are allowed to exceed MATSim's speed limit by 20 %. This is further justified by the fact that real world drivers tend to overspeed as well. (iii) Moreover, acceleration rates in the transition zones are limited to realistic values. If the acceleration exceeds 5 m/s^2, the identified CADC/DS/FKFS segment is discarded and a better one is iteratively chosen. The so generated profile is considered acceptable if the daily traveled distance of both profiles $v_{\text{raw}}(t)$ and $v_{\text{enr}}(t)$ have a relative error of less than 5 %. The relative error $e_{\text{veh}}^{\text{day, rel}}$ is calculated as follows

$$e_{\text{veh}}^{\text{day, rel}} = \frac{\left| \left(\sum_{n=1}^{N} \int_{t=1}^{T} v_{\text{enr, veh}}(n,t)dt \right) - \int_{t=1}^{NT} v_{\text{raw, veh}}(t)dt \right|}{\int_{t=1}^{NT} v_{\text{raw, veh}}(t)dt} \tag{3}$$

where veh is the vehicle's identification number, N the maximum of 1440 1 min-segments per day and T the total of 60 s per minute.

2.3 Microscopic drive cycles from sequential tool-coupling

Another approach to enhance MATSim's speed profiles consists in subjecting the simulated vehicle trajectories to an additional microscopic traffic simulation. In this context, *Simulation*

of Urban Mobility (SUMO)[30] constitutes a rather natural choice as it is the most popular open-source microscopic traffic simulation framework[12]. SUMO is well established in the fields of traffic management, traffic light evaluation and (in recent years) the simulation of vehicular communications. It provides many interfaces that allow external applications to interfere online with the traffic simulation. In this work, the *Traffic Control Interface* (TraCI) is used to retrieve and instantaneously manipulate object attributes.

Network generation To build a SUMO network based on an existing MATSim model, the geographical area of interest is independently imported from *OpenStreetMap* with SUMO *NETCONVERT*[13]. Network differences in MATSim and SUMO are exemplarily depicted in Figure 4 for the *Bergheimer Steige* in Stuttgart. In MATSim, networks can be imported via the *OsmNetworkReader* with varying degree of resolution, e.g. rather simple networks with reduced number of links (4b) or more complex ones which account more accurately for curved road shapes (4c)[14]. In general, it can be noted that MATSim paths (regardless of their import resolution) already account for corrective measures for road geometry and altitude differences. Consequently, the path lengths fit r ather w ell i n d irect c omparison w ith *G o ogleMaps*. SUMO networks in contrast, feature the most sophisticated network design but additional length gains by altitude differences are not projected to the 2-dimensional network by default. In our work, those data are loaded from an additional elevation model.

Travel demand transfer MATSim-SUMO The travel demand in our SUMO simulation comes entirely from MATSim. For that purpose, all MATSim links bordering a chosen test case are identified. N ext, a ll v ehicles p assing t hose l inks a re r ecorded d uring M ATSim simulation with (i) *vehicleID*, (ii) vehicle route and (iii) time of test case entrance and exit. In case of a MATSim sample run, the travel demand in SUMO is upscaled accordingly by injecting *cloned* vehicles. To prevent severe gridlocks in SUMO, a random time offset (sampled from a Gaussian distribution) is added to the network entering time of the cloned cars. Having all departure

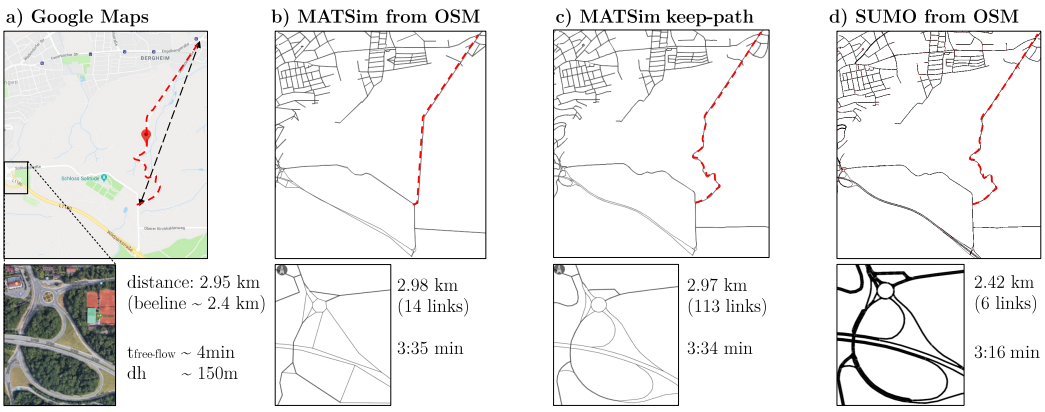

Figure 4: Network differences based on the Bergheimer Steige test case as defined in Section 3.

[12] https://sumo.dlr.de/docs/index.html

[13] In principle, SUMO networks can also be imported from MATSim. This proceeding, however, proved not beneficial for our purpose as MATSim discards some network information which is required in SUMO.

[14] This network, however, behaved poorly in our simulation, as it has too many short links where the artefacts discussed in Section 2.1 occur.

times settled, the *linkIDs* from MATSim are translated into corresponding *edgeIDs* in SUMO to write the final t rips-file fo r SU MO si mu lation. Fi nally, al l *ve hi cle tr ip s* in SU MO ar e converted in *vehicle routes* by *DUAROUTER*. Due to the small test case sizes in this work (see Sect. 3), the SUMO routes match with those in MATSim.

Microscopic traffic dynamics Traffic dynamics in SUMO are realized by car-following models (such as *Krauss* [26] or *Intelligent Driver Model (IDM)* [39]) and lane-change models (such as *LC2013* [13]). In this contribution we use the default Krauss-model according to which the vehicles drive as fast as possible while maintaining a perfect safety distance to the leading car. The safe speed is computed as follows [25]:

$$v_{\text{safe}}(t) = v_l(t) + \frac{g(t) + v_l(t)\tau}{\frac{\overline{v}}{b} + \tau} \tag{4}$$

where $v_l(t)$ represents the speed of the leading vehicle, $g(t)$ the gap to the leader, τ the reaction time, b the maximum deceleration of the follower and $\overline{v}$ the mean velocity of following and leading vehicle. As v_{safe} may exceed the legal speed limit of the road or surpass the vehicle's capability, the actual targeted velocity is limited to the minimum of those three. On top of that, a *driver imperfection* σ has been introduced in SUMO that causes random deceleration to model speed fluctuations that lead to spontaneous jams at high traffic densities. Furthermore, each vehicle draws an individually chosen *speedFactor* from a normal distribution to represent a wider variety of human driving styles, e.g. drivers that notoriously stay above or below the legal speed limit. Figure 5 displays an exemplary velocity profile extracted from SUMO simulation by also providing information on the current speed limit and the vehicle's elevation profile.

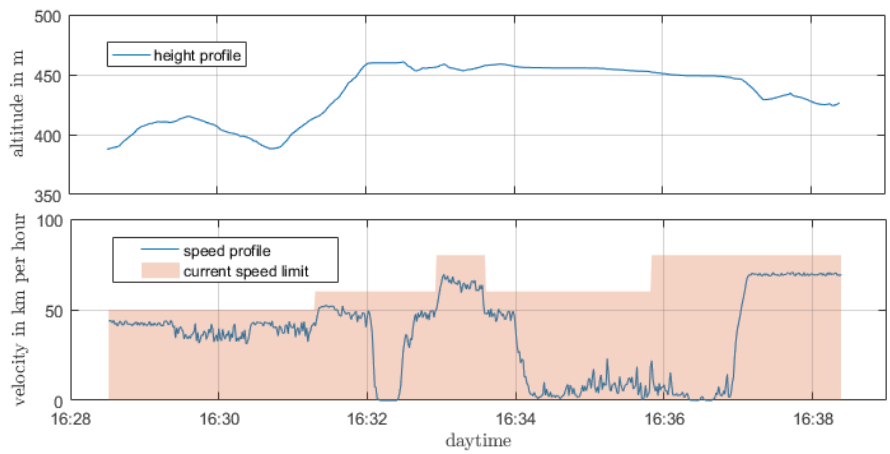

Figure 5: Elevation (top panel) and velocity profile (bottom) for an exemplary vehicle.

3 Test case analysis

This section evaluates all approaches elaborated in Section 2 in their ability to deduce reasonable velocity profiles. The test case analysis relies on an existing MATSim model for the Stuttgart region. Following an approach similar to [6], the MATSim model has been built (by RB)

on the basis of a *mobiTopp* [33] travel demand model for the Stuttgart region provided by the *Verband Region Stuttgart* as part of a research collaboration. In total three different test cases were identified t hat d iffer i n r oad t ype, n etwork t opology a nd r ight o f w ay r u les: (i) The *Bergheimer Steige* features no crossroads but sharp turns and road gradients up to 15 %. This test case seeks to analyze to what extent slope and curves influence v ehicle s peed i n simulation. (ii) The *Motorway A8* (Kreuz Stuttgart to AS Stuttgart Möhringen) allows the analysis of traffic dynamics on motorways. (iii) The *Kräherwald* test case (leading from Kräherwald/junction Zeppelinstraße to University of Stuttgart) is of mixed inner-city and highway character and is part of the FKFS cycle as illustrated in Figure 6 (center and right panel).

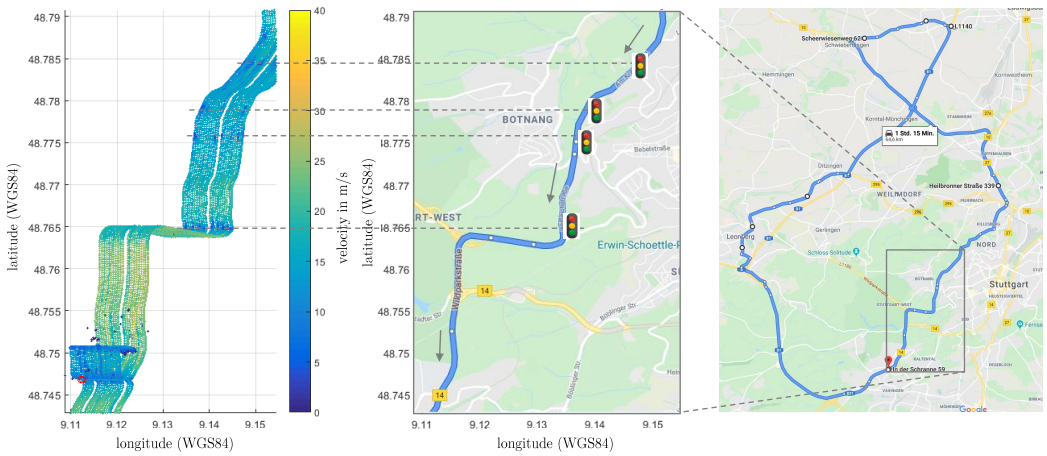

Figure 6: The right plot illustrates the FKFS circuit. Here, the grey rectangle borders the actual *Kräherwald* test case whose zoomed trajectory is given at center. To the left, the spatial velocity profiles of 22 measured FKFS cycles are provided for the *Kräherwald* test case.

All test cases are simulated in MATSim and SUMO for one day. For each test case, driving cycles are deduced by means of (a) pure MATSim simulation, (b) enhanced MATSim simulation with real driving cycles and (c) sequential MATSim-SUMO tool-coupling. The driving cycles are then compared based on **aggregated dynamics** such as velocity and acceleration distributions, overall traveled time and distance as well as average congestion ratio and energy consumption (Sec. 3.1). As further evaluation criteria serve the accuracy of **time- and space-dependent velocity profiles** (Sec. 3.1 and 3.2) as well as the simulation approaches' **large-scale feasibility** and **automation capability** (Sec. 3.3). Energy related KPI are derived from vehicle simulation in *GT-Suite*[15]. For this purpose, the following vehicle specifications have been used: vehicle mass (including battery and powertrain components) $m = 1545\,\mathrm{kg}$, constant tire rolling resistance $c_\mathrm{R} = 0.011$, vehicle front area $A_\mathrm{f} = 2.2\,\mathrm{m}^2$, vehicle air resistance coefficient $c_\mathrm{W} = 0.27$, road friction coefficient $c_\mathrm{fric} = 1$, battery capacity $E_\mathrm{bat} = 60\,\mathrm{kWh}$ and engine power $P_\mathrm{eng} = 200\,\mathrm{kW}$. Driving dynamics and energy related KPI are additionally compared to 22 measured and GPS-tracked FKFS cycles for the *Kräherwald* test case[16]. To ensure comparability, the measured velocity profiles are equally passed to GT-Suite simulation.

[15] https://www.gtisoft.com/gt-suite/gt-suite-overview/

[16] All data were gathered by the *Research Institute of Automotive Engineering and Vehicle Engines Stuttgart* based on a contract research *"Bordnetzmessungen am Elektrofahrzeug (cZero)"* with the *Robert Bosch GmbH*.

3.1 KPI comparison

The following assessments refer to Table 1 which summarizes for each test case and simulation approach the most important aggregated KPI. For each test case, only a sample of simulated vehicles have been tracked microscopically. The exact numbers of tracked and simulated vehicles are indicated within the table as well.

Aggregated vehicle dynamics and energy related KPI In general, the average **traveled distance** of all tracked vehicles is similar in all simulation scenarios. Differences mainly arise due to different network designs and import functionalities. Every time a road attribute changes in OSM both MATSim and SUMO create a new link/edge. In contrast to SUMO, MATSim links are represented by straight lines only. In case this straight line deviates strongly from the actual road shape, MATSim inserts artificial n odes t o p reserve t he n etwork g e ometry. B y consequence, one SUMO edge often represents several MATSim links which leads to longer SUMO distances especially in small test cases like ours. The calculated distances of the enriched scenario are purely artificial a s t hey d o n ot c orrespond t o t he a ctual t arget t r ajectories. N evertheless, they are reasonable enough considered the little effort it took to implement the enrichment procedure. Solely the *Motorway A8* test case reveals discrepancies in traveled distance higher than the desired 5 % error margin. This however, is not the fault of the enhancement method itself. Those imperfections are caused by an insufficient number of available fast-driving 1 min-segments in Section 2.2 which also lead to low average velocities and energy consumptions. The validity of the enrichment procedure is therefore directly tied to a wide range of underlying measured driving cycles.

The **average travel time, velocity and energy consumption** are strongly congestion dependent. As the approaches base on different traffic dynamics (queue vs. car-following model) and network attributes (node vs. signaled intersection), the same ego-vehicle is differently delayed throughout the network which leads to different traffic conditions. Naturally, this affects average travel time, velocity and energy consumption. The inconsistencies in congestion modeling

Table 1: Aggregated KPI comparsion for all three test case
(no. of tracked/simulated ego-vehicles)

	KPI	MATSim	enriched	SUMO
Bergheimer Steige (359/7170)	average traveled distance in km	2.9	2.8	3.1
	average traveled time in min	4.3	4.8	4.6
	average congestion rate	0.85	0.72	0.84
	average velocity in km/h	43	36	42
	average energy consumption in kWh/100km	9.2	11.6	16
Motorway A8 (385/379591)	average traveled distance in km	6	5	6.8
	average traveled time in min	3.7	4.1	3.7
	average congestion rate	0.88	0.55	0.97
	average velocity in km/h	92	59	101
	average energy consumption in kWh/100km	23.8	15.8	24
Kräherwald (444/110779)	average traveled distance in km	5.5	5.4	5.7
	average traveled time in min	7.8	8.2	6.6
	average congestion rate	0.82	0.76	0.85
	average velocity in km/h	52	48	53
	average energy consumption in kWh/100km	11.8	12.4	15.6

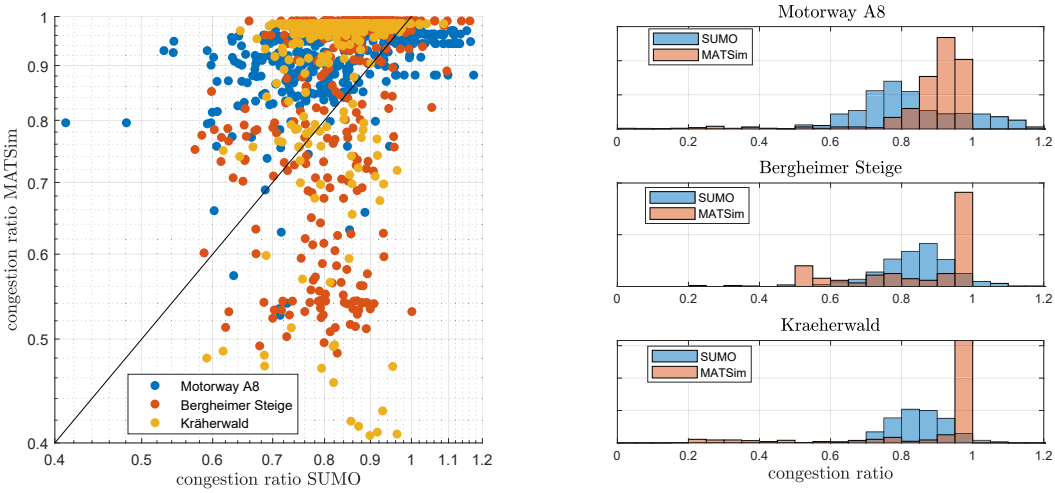

Figure 7: Inconsistencies in congestion modeling in MATSim and SUMO for all three test cases illustrated as log-log plot where each dot represents a tracked vehicle (left) and congestion ratio histograms (right). A congestion ratio of *one* corresponds to free-flow driving conditions.

are further illustrated in Figure 7 on the left, where the congestion ratio for both MATSim and SUMO are compared. At this, each dot represents a tracked vehicle. The congestion ratio is defined as the ratio of actual travel time and the free-flow travel time simulated in MATSim. A congestion ratio of *one* corresponds to free-flow driving conditions, whereas a ratio near *zero* signifies a blocked road[17]. A perfect match would theoretically result in a diagonal line. As depicted in Figure 7 this is seldom the case and needs to be investigated further. The histograms on the right show that the traffic conditions in MATSim are often too optimistic (presumably on links where the spatial queue did not propagate due to the artefacts discussed in Section 2.1) or way too pessimistic (presumably on short links).

When comparing the **speed and acceleration distributions** of all simulations, considerable differences in all approaches become apparent. Figure 8 displays the normalized velocity and acceleration histograms of all 359 tracked vehicles for the *Bergheimer Steige* test case. As expected, pure MATSim simulation exhibits unrealistic driving dynamics as it only accounts for average link speeds with no oscillations around the target velocity. Consequently the acceleration rate is predominately zero. In between two links however, the acceleration may jump from zero to an value predefined by the next link's speed limit. The enriched profiles feature more realistic driving dynamics, but as will be shown in Section 3.2, they are only as good as MATSim's capability to model spatial congestion patterns (which is limited at the moment). SUMO, in contrast, features more bell-shaped distributions (around local maxima) which, however, have not been validated yet. In the enriched MATSim and SUMO simulation the maximal acceleration is limited by design to $\pm 5\,\mathrm{m/s^2}$ absolute. However, compared to real-world driving, the acceleration rates in SUMO are distributed too perfectly as equally stated in [1, 21, 37].

[17]As SUMO allows overspeeding (here: up to 20 %) and as SUMO link lengths do not match those of MATSim perfectly, congestion ratios greater *one* may result.

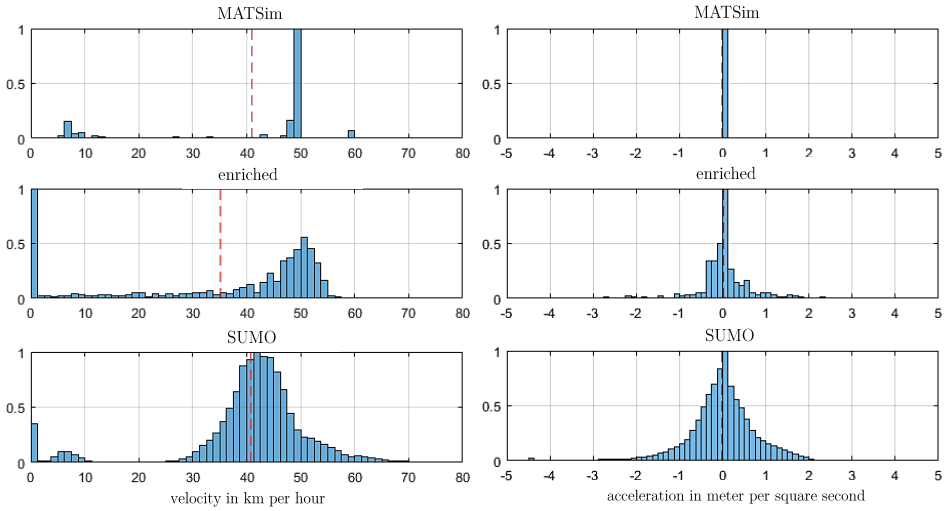

Figure 8: Normalized velocity (left) and acceleration distributions (right) for 359 tracked vehicles within the Bergheimer Steige test case with a temporal resolution of $\mathrm{d}t = 1\,\mathrm{s}$. The red dashed lines represent the mean values.

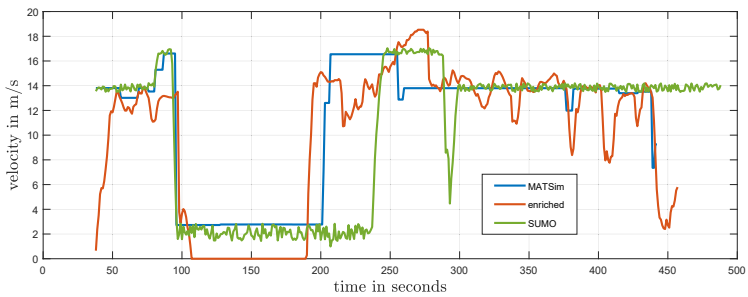

Figure 9: Time-dependent speed profile of a chosen vehicle of the Bergheimer Steige test case as simulated by the different simulation approaches.

Time-dependent speed profiles In Figure 9 the time-dependent velocity profiles for the same vehicle are shown. Even though the starting times are identical for all scenarios for the chosen vehicle[18], the car is differently delayed due to discrepancies in traffic conditions, network distances, traffic signals and right-of-way rules. Whereas MATSim's velocity profile is rather steplike due to the average link speed, SUMO shows strong oscillations around the target velocity (possibly arising from the driver imperfection σ). However, compared to real world driving, SUMO's oscillation amplitude seems too homogeneous and the frequency too high-frequent. This may be solved by a better parametrized car-following model, but as our approach aims at autonomous driving (AD) applications in future no further effort was put into this task.

[18]This might not always be the case. If strong congestion occurs on a vehicle's departure link, the moment of network entering can be delayed artificially.

3.2 Comparison against GPS-tracked FKFS cycles

In this section, the simulated driving cycles for the *Kräherwald* test case are compared against 22 measured FKFS cycles to assess the quality of the simulated results. To do so, only the part of the FKFS cycle is considered that overlaps with the *Kräherwald* test case as displayed in the right panel of Figure 6.

Space-dependent speed profiles All simulated (in SUMO only) and measured driving cycles of the *Kräherwald* test case are spatially compared in Figure 10 top panels. The bottom panels provide additional information on the vehicles' minimum, mean and maximum velocity at each location of the test case. As indicated in Figure 6, the trajectory undergoes first four successive traffic lights, becomes then a west-heading highway and is finally merged into another arterial road before turning abruptly south. Those characteristic become clearly visible in the both data sets in the form of sudden drops in velocity. In contrast to the FKFS data (that unfortunately reflect free-flow driving conditions only), the SUMO simulation on the right side exhibits some congestion during the day which leads to longer waiting queues in front of the traffic signals and especially when both arterial roads meet. Moreover, in real life locals tend to anticipate upcoming speed limit changes and adjust their velocity accordingly before the actual traffic sign occurs . This is especially true when the speed limit rises. In our simulation, however, the rise and fall of the speed limit is rather step-like. In the context of autonomous driving this simplification is not necessarily disadvantageous as future AVs might adapt to speed limits in a similar manner.

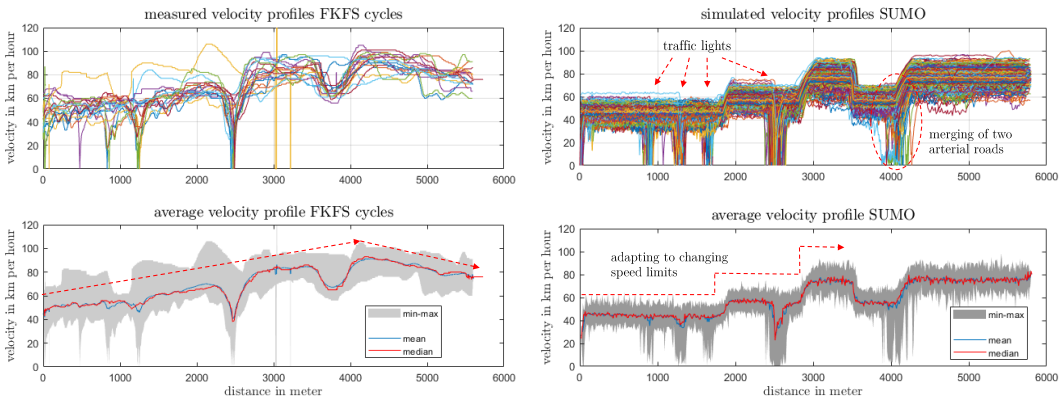

Figure 10: Stacked velocity-space profiles for 22 measured FKFS cycles (left) and 444 simulated SUMO vehicles (right) for the *Kräherwald* test case.

Aggregated vehicle dynamics For further plausibility checks, only those vehicles from the traffic simulation are benchmarked with FKFS data that exhibit similar traffic conditions. Unfortunately all measured cycles feature free-flow driving conditions, consequently no conclusions to the partly or fully congested state can be drawn. Table 2 summarizes selected **aggregated KPI** for a chosen, simulated vehicle and compares them with three different FKFS vehicles. Generally, all listed KPI match rather well for the non congested state regardless of the driving cycle deduction approach. A slightly different picture emerges when regarding the **velocity distribution** under free-flow driving conditions. As evidenced in Figure 11 on the

Table 2: FKFS benchmarking for a chosen simulated vehicle for the non-congested state.

KPI	**simulated cycles via**			**FKFS cycles**		
for the non-congested state	MATSim	enriched	SUMO	car 1	car 2	car 3
distance in m	5489	5460	5698	5951	5955	5942
average velocity in m/s	16.4	15.2	15.6	15.0	14.5	17.2
travel time in min	5.6	6.0	6.1	6.6	6.9	5.8
energy consumption in kWh/100km	20.1	18.3	18.8	18.8	22.0	19.6
congestion ratio	0.95	0.88	0.9	0.87	0.84	0.99

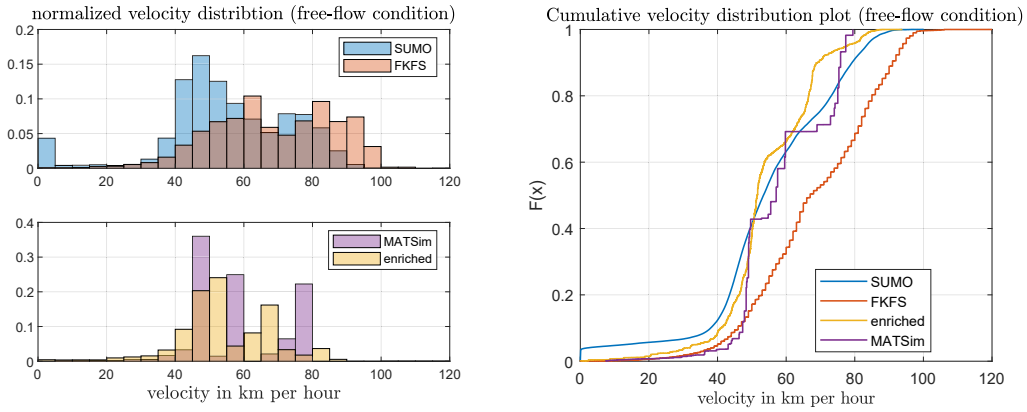

Figure 11: Probability density (left) and cumulative velocity distribution function (right).

left side, real world drivers (represented by the FKFS cycles) tend to drive faster than those simulated in SUMO. Whereas the SUMO simulation exhibits velocity peaks around 45 and 75 km/h, the measured data reach their local maxima around 62 and 82 km/h. Beyond that, the SUMO simulation features many velocities near zero which are not present in the measured data. Simulated vehicles obviously stand a higher chance to hit at least one of the four traffic lights. This also shows as an offset in the cumulative velocity distribution in the right panel of Figure 11: Whereas the graph gradients of SUMO and FKFS match rather well, SUMO's cumulative velocity distribution is shifted considerably more to lower velocities due to the traffic light downtimes. As further expected, MATSim's velocity distribution correlates poorly with the corresponding FKFS data due to the simplified queuing model. The enrichment technique compensates some of those shortcomings, but follows MATSim's trend still too closely. Using a larger sample of measured driving cycles for the enrichment, will likely lead to more realistic velocity distributions.

At this point, however, it has to be emphasized that the simulated driving cycles cannot be validated with the measured FKFS cycles for two reasons: (i) The 22 measured drive cycles are statistically not significant enough to represent the driving behavior of the *Kräherwald* test case during one day. (ii) To validate single profiles, the ego-vehicle's exact environment (e.g ambient traffic and traffic signals) needs to be modeled as encountered during measurement campaign. Unfortunately, neither MATSim nor SUMO are capable to model surrounding vehicles in such a manner. Furthermore, radar and LIDAR data are required to collect necessary data.

3.3 Discussion and implications for final concept choice

This section summarizes all quantitative results of the preceding sections, places them into the context of the key requirements postulated in Section 1 and complements them with qualitative remarks on the approaches' large-scale feasibility and automation capability.

With respect to city-wide SAEV fleet simulation **(KR1)**, MATSim has advantage over SUMO in scalability and computational performance on the one hand and existing fleet simulation functionalities on the other hand. The enrichment and tool-coupling approach also benefit from MATSim's capabilities in this regard, as the modeled fleet constraints as well as the impact of different dispatching and routing algorithms do equally reflect in those solutions. Regarding individual vehicle states **(KR2)**, MATSim and SUMO prove equally capable. Provided minute-wise drive cycle enhancement, the enrichment procedure should fare well in this regard as well since downtime phases are not altered considerably. Larger enrichment segments, however, increase the chance of annihilating idling periods or inserting additional ones.
The approaches' capability to derive reasonable velocity profiles **(KR3)** has extensively been analyzed in the previous section. Given similar traffic conditions, aggregated trip statistics (e.g. average velocity, traveled distance and time) are well captured by each approach. However, as highlighted in Subsection 3.1, even for a given ego-vehicle the traffic conditions differ considerably between the different approaches due differences in traffic dynamics and network interpretation. A central task in future work therefore relates to the model calibration in terms of (real-world-observed) congestion patterns. Unfortunately, MATSim (and therefore the enrichment approach as well) has some shortcomings in spatial congestion modeling. Another deficit of MATSim is its incapability to model realistic velocity and acceleration profiles due to its simplified queuing model. A satisfying solution that solely relies on MATSim without further enhancement is therefore not conceivable. The velocity profiles obtained from the enrichment procedure closely resemble real world measurements. However, it is not straight forward to transfer this approach to autonomous driving applications, since it depends on measurements as input data. A major drawback for the enrichment approach is therefore its missing sensibility to different driving

Table 3: Approaches' suitability to model detailed SAEV driving profiles
(✓suitable, o limited suitability, - not suitable, * no statement possible)

	key requirements	**pure MATSim**	**enriched MATSim**	**pure SUMO**	**MATSim-SUMO tool-coupling**
KR1:	large-scale, multi-modal SAEV fleet simulation with sensitivity to:	✓	✓	-	✓
	- range & charging constraints	✓	✓	o	✓
	- dispatching, routing & pricing strategies	✓	✓	o	✓
KR2:	vehicle states	✓	✓	✓	✓
KR3:	realistic velocity profiles with sensitivity to:				
	- human/ autonomous driving	-/-	✓/o	o/*	o/*
	- congestion rates	o	o	✓	✓
	- transport infrastructures	o	-	✓	✓
KR4:	further time series such as				
	- height/ occupancy profiles	o/✓	-/o	o/*	o/✓
KR5:	feasibility in terms of:				
	- data availability	✓	o	o	o
	- robustness against critical error	✓	✓	-	-
	- automation capability	✓	✓	o	o
	- computational resources	✓	✓	o	✓

styles or platooning effects. SUMO in contrast, enables the deduction of detailed drive cycles whose drive dynamics prove too artificial to reflect for human driving, but may be reasonable enough for autonomous driving. In contrast to MATSim, SUMO provides many features to tweak driving dynamics in a methodological manner. Another strong point of SUMO is that the simulated vehicles react sensitive to diverse transport infrastructures and are able to mimic different driving maneuvers such as stop&go-patterns or zip merging. Unfortunately, SUMO does not account for reduced velocities in narrow curves. Nanoscopic traffic simulation tools such as *CarMaker*[19] would be required to address these kind of topics. The same applies for road gradients: road slope can technically be modeled in each simulation scenario **(KR4)** but requires access to accurate height data. These data, however, relate to the Earth's surface only and consequently produce invalid results for road tunnels. And even with slope modeled, the latter has so far no impact on the vehicles' driving behavior. Slope only influences energy consumption in a subsequent vehicle simulation. Nonetheless it has to be emphasized at this point, that numerous car-following models exist for SUMO. Some may address those issues already. At this point, those options have not been adequately tested nor investigated yet.

Apart from those quantitative KPI, all simulation approaches differ considerably in practical feasibility and automation capability **(KR5)**. With regard to the key requirements KR1-KR4, the MATSim-SUMO tool-coupling approach seems to be the most promising solution to deduce representative SAEV drive cycles as summarized in Table 3. However, its automation capability remains questionable due to the high effort in setting-up the network. SUMO networks are very detailed and therefore require additional data which OSM does not provide, e.g. detailed elevation information, traffic light positions and control. SUMO's autogenerated networks are sometimes misleading as the underlying OSM attributes are non-existing or error-prone and/or the data are too complex to be interpreted correctly by the default import functionalities. This is shown by (a) faulty turning lanes, (b) poorly guessed traffic light positions, (c) poorly joined complex junctions and (d) uncoordinated traffic light initialization. Manual editing represents a most time consuming task. A further serious drawback for all SUMO related approaches is their proneness to artificial deadlocks. Those gridlocks are created for example by two impeding cars, where the left likes to turn right and vice versa. Those gridlocks do not naturally resolve in SUMO, but can only be counteracted by enabling further options such as *time to teleport* or *ignoring junction blockers*. However, those options do not help if the ego-vehicle selected for drive cycle derivation is affected, as this vehicle then cannot complete its daily trajectory. MATSim in contrast, encounters no data-availability or automation problems due to its simplified network representation. Taken all pros and cons into consideration, the MATSim-SUMO tool-coupling seems most promising despite its automation challenges.

4 Conclusion

This contribution presents different approaches to simulate 24h driving cycles for SAEVs. The approaches are evaluated for a set of test cases. From this, a sequential tool-coupling of meso- and microscopic traffic simulation was found to be most promising with respect to the key requirements defined in Section 1. SAEV driving profiles are derived as follows: Depending on different fleet configurations and pricing concepts, SAEV fleets are implemented and simulated in MATSim on a large-scale, multi-modal network. Based on the simulation results, all SAEV trajectories are analyzed with respect to their daily use patterns, such as driven distance,

[19] https://ipg-automotive.com/de/produkte-services/simulation-software/carmaker/

operating time or number of served trips. Next, representative fleet vehicles are automatically identified and post-processed to be simulated in SUMO. To this end, the time-dependent travel demand of all roads in close proximity to the actual target trajectory is recorded in MATSim and transfered to the SUMO model. To reduce network setup effort, only the trajectories of the chosen vehicles (and their close neighborhood) are modeled in SUMO. Besides, each vehicle tagged as SAEV in MATSim simulation is featured with autonomous driving characteristics in SUMO. The ego-vehicle's speed profile is then derived from SUMO simulation.

At present, this tool-coupling approach works for test cases only as the procedure involves manual network matching and cleaning efforts. Its application to city-wide scenarios necessitates tool-chain automation which, however, constitutes a most challenging task. Further research is therefore required to implement the tool-chain in such a way that – starting from an existing, calibrated MATSim model – the SUMO model is setup, simulated and evaluated without further human intervention. To this end, the following aspects are addressed in future work:
(a) Dealing with inconsistencies in MATSim and SUMO. A sequential tool-coupling requires aligning both frameworks in (i) network representation, (ii) route choice, (iii) traffic dynamics on a macroscopic level and (iv) traffic performance. Otherwise, the travel demand transfer from MATSim to SUMO leads to severe gridlocks in the more congestion-prone microscopic traffic simulation and SAEVs cannot serve their appointed customers in time. Consequently, the frameworks' discrepancies need to be analyzed in more detail to derive alignment measures.
(b) Automated network modeling in SUMO. To solve the bottleneck of tool-chain automation, methods and algorithms need to be elaborated to solve network cleaning, traffic light location and control issues in an automated fashion. As time-dependent traffic volumes on all intersection are known from MATSim simulation, approaches are elaborated that (i) detect and eliminate artificial bottlenecks in the SUMO network that fail to handle the appointed traffic flow and (ii) mirror the decision makings of an actual traffic planner to initialize traffic lights.
(c) Automated travel demand transfer. Another obstacle for tool-chain automation represents the travel demand transfer from MATSim to SUMO simulation. This issue is solved by a robust network matching concept with dynamic meso-micro borders.

5 Acknowledgments

This work was supported by the Reasearch Programme on Automation and Connectivity in Road Transport of the German Federal Ministry of Transport and Digital Infrastructure (funding number 16AVF2147B).

References

[1] G. Amirjamshidi and M. J. Roorda. Development of simulated driving cycles for light, medium, and heavy duty trucks: Case of the toronto waterfront area. *Transportation Research Part D: Transport and Environment*, 34:255 – 266, 2015.

[2] A. Anastassov, D. Jang, and G. Giurgiu. Driving speed profiles for autonomous vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, jun 2017.

[3] K. W. Axhausen, A. Horni, and H. J. Herrmann. Final report: The risk for a gridlock and the macroscopic fundamental diagram. Technical report, 2015.

[4] J. Bischoff and M. Maciejewski. Simulation of city-wide replacement of private cars with autonomous taxis in berlin. *Procedia Computer Science*, 83:237–244, 12 2016.

[5] J. Bischoff, F. J. Marquez-Fernandez, G. Domingues-Olavarria, M. Maciejewski, and K. Nagel. Impacts of vehicle fleet electrification in Sweden - a simulation-based assessment of long-distance trips. Technical report, MT.ITS, 2019.

[6] L. Briem, N. Mallig, and P. Vortisch. Creating an integrated agent-based travel demand model by combining mobiTopp and MATSim. *Procedia Computer Science*, 151:776–781, Jan. 2019.

[7] R. Chen and M. W. Levin. Dynamic user equilibrium of mobility-on-demand system with linear programming rebalancing strategy. *Transportation Research Record: Journal of the Transportation Research Board*, 2673(1):447–459, jan 2019.

[8] T. D. Chen, K. M. Kockelman, and J. P. Hanna. Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions. *Transportation Research Part A: Policy and Practice*, 94:243 – 254, 2016.

[9] S. Chugh, P. Kumar, M. Muralidharan, M. K. B, M. Sithananthan, A. Gupta, B. Basu, and R. K. Malhotra. Development of delhi driving cycle: A tool for realistic assessment of exhaust emissions from passenger cars in delhi. In *SAE Technical Paper Series*. SAE International, apr 2012.

[10] T. V. da Rocha, A. Can, C. Parzani, B. Jeanneret, R. Trigui, and L. Leclercq. Are vehicle trajectories simulated by dynamic traffic models relevant for estimating fuel consumption? *Transportation Research Part D: Transport and Environment*, 24:17–26, oct 2013.

[11] C. F. Daganzo. The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B*, Vol. 28B(No. 4):pp. 269–287, 1994.

[12] K. P. Divakarla, A. Emadi, and S. N. Razavi. Journey mapping—a new approach for defining automotive drive cycles. *IEEE Transactions on Industry Applications*, 52(6):5121–5129, nov 2016.

[13] J. Erdmann. SUMO's lane-changing model. In *Modeling Mobility with Open Data*, pages 105–123. Springer International Publishing, 2015.

[14] D. J. Fagnant and K. M. Kockelman. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in austin, texas. *Transportation*, 45(1):143–158, Jan 2018.

[15] A. Fotouhi and M. Montazeri-Gh. Tehran driving cycle development using the k-means clustering method. *Scientia Iranica*, 20(2):286 – 293, 2013.

[16] Fraunhofer Institute for Industrial Mathematics (ITWM). Virtual measurement campaign. product flyer, accessed: april 2020.

[17] C. Gawron. An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. *International Journal of Modern Physics C*, Vol. 09(No. 03):pp. 393–407, 1998.

[18] Q. Gong, S. Midlam-Mohler, V. Marano, and G. Rizzoni. An iterative markov chain approach for generating vehicle driving cycles. *SAE International Journal of Engines*, 4(1):1035–1045, apr 2011.

[19] T. Holdstock and M. Bryant. Electric drivetrain architecture optimisation for autonomous vehicles based on representative cycles. In *17th Int. CTI Symp. Automotive Transmissions, Berlin*, 2018.

[20] A. Horni, K. Nagel, and K. W. Axhausen, editors. *The multi-agent transport simulation MATSim.* Ubiquity Press, London, 2016.

[21] Y. Hou, E. Wood, E. Burton, and J. Gonder. Suitability of synthetic driving profiles from traffic micro-simulation for real-world energy analysis: Preprint. *National Renewable Energy Laboratory (NREL)*, 10 2015.

[22] S. Hörl, F. Becker, T. J. P. Dubernet, and K. W. Axhausen. Induzierter Verkehr durch autonome Fahrzeuge. Eine Abschätzung. Technical report, SNF and ETH Zürich, 2019.

[23] I. Kaddoura, G. Leich, and K. Nagel. The impact of pricing and service area design on the modal shift towards demand responsive transit. Submitted to the 9th Int. Workshop on Agent-based Mobility, Traffic and Transportation Models (ABMTRANS), Warsaw, Poland, Apr. 2020.

[24] S. H. Kamble, T. V. Mathew, and G. Sharma. Development of real-world driving cycle: Case study of pune, india. *Transportation Research Part D: Transport and Environment*, 14(2):132–140, 2009.

[25] S. Krauss. Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics. Technical report, 1998. Dissertation. Mathematisch-Naturwissenschaftliche Fakultät, Universität Köln and German Aerospace Center (DLR). LIDO-Berichtsjahr 1999.

[26] S. Krauss, P. Wagner, and C. Gawron. Metastable states in a microscopic model of traffic flow. *Physical Review E*, 5:5597–5602, 1997. LIDO-Berichtsjahr 1997,.

[27] J. Liu, K. Kockelman, P. Bösch, and F. Ciari. Tracking a system of shared autonomous vehicles across the austin, texas network using agent-based simulation. *Transportation*, 08 2017.

[28] J. Liu, K. Kockelman, and A. Nichols. Anticipating the emissions impacts of smoother driving by connected and autonomous vehicles, using the moves model. 01 2017.

[29] B. Loeb, K. M. Kockelman, and J. Liu. Shared autonomous electric vehicle (SAEV) operations across the Austin, Texas network with charging infrastructure decisions. *Transportation Research Part C: Emerging Technologies*, 89:222 – 233, 2018.

[30] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner. Microscopic traffic simulation using SUMO. In *The 21st IEEE IC on Intelligent Transportation Systems*, pages 2575–2582. IEEE, Nov. 2018.

[31] M. Maciejewski. Benchmarking minimum passenger waiting time in online taxi dispatching with exact offline optimization methods, 2014.

[32] M. Maciejewski, J. Bischoff, and K. Nagel. An assignment-based approach to efficient real-time city-scale taxi dispatching. *IEEE Intelligent Systems*, 31(1):68–77, jan 2016.

[33] N. Mallig, M. Kagerbauer, and P. Vortisch. mobiTopp – a modular agent-based travel demand modelling framework. *Procedia Computer Science*, 19:854 – 859, 2013. 4th IC on Ambient Systems, Networks and Technologies (ANT) , 3rd IC on Sustainable Energy Information Technology (SEIT).

[34] A. Moreno, A. Michalski, C. Llorca, and R. Moeckel. Shared autonomous vehicles effect on vehicle-km traveled and average trip duration. *Journal of Advanced Transportation*, 2018, 2018.

[35] P. Nyberg, E. Frisk, and L. Nielsen. Using real-world driving databases to generate driving cycles with equivalence properties. *IEEE Transactions on Vehicular Technology*, 65(6):4095–4105, 2016.

[36] S. Shi, N. Lin, Y. Zhang, J. Cheng, C. Huang, L. Liu, and B. Lu. Research on Markov property analysis of driving cycles and its application. *Transportation Research Part D: Transport and Environment*, 47:171–181, aug 2016.

[37] G. Song, L. Yu, and Y. Zhang. Applicability of traffic microsimulation models in vehicle emissions estimates. *Transportation Research Record: Journal of the Transportation Research Board*, 2270(1):132–141, jan 2012.

[38] I. H. Tchappi, V. C. Kamla, S. Galland, and J. C. Kamgang. Towards an multilevel agent-based model for traffic simulation. *Procedia Computer Science*, 109:887 – 892, 2017. 8th IC on Ambient Systems, Networks and Technologies (ANT), 7th IC on Sustainable Energy Information Technology (SEIT).

[39] M. Treiber, A. Hennecke, and D. Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62:1805–1824, 02 2000.

[40] Z. Xiao, Z. Dui-Jia, and S. Jun-Min. A synthesis of methodologies and practices for developing driving cycles. *Energy Procedia*, 16:1868–1873, 2012.

[41] P. Yuhui, Z. Yuan, and Y. Huibao. Development of a representative driving cycle for urban buses based on the k-means cluster method. *Cluster Computing*, 22(S3):6871–6880, jan 2018.

[42] H. Zhang, C. J. R. Sheppard, T. E. Lipman, and S. J. Moura. Joint fleet sizing and charging system planning for autonomous electric vehicles. 2018.

[43] X. Zhao, Q. Yu, J. Ma, Y. Wu, M. Yu, and Y. Ye. Development of a representative EV urban driving cycle based on a k-means and SVM hybrid clustering algorithm. *Journal of Advanced Transportation*, 2018:1–18, nov 2018.