Finite State Machines and Timed Automata: A Hierarchical Approach for Integrated Traffic Microsimulations

Frank Lehmann[1], Partha S. Roop[2], and Prakash Ranjitkar[1]
Department of Civil and Environmental Engineering, University of Auckland, New Zealand
Department of Electrical and Computer Engineering, University of Auckland, New Zealand
Email: fleh889@aucklanduni.ac.nz, {p.roop, p.ranjitkar}@auckland.ac.nz

***Abstract*—Microscopic traffic simulations capture the trajectories of individual drivers as responses to stimuli from their surroundings (i.e. other vehicles or road conditions). Mathematically, these models are usually designed with differential equations or as sets of integer-based rules. Since both approaches have disadvantages, we propose an in-between approach built with Timed Automata and Finite State Machines (FSM) to reproduce the human behaviour. The fundamental idea is to model the switches between a limited set of discrete acceleration levels with a FSM and derive all other trajectory features from there. The duration for which this constant acceleration is maintained is not fixed and is modelled by a (probabilistic) Timed Automaton (TA). With this arrangement, the complexity of CF behaviour can be represented with high computational efficiency in large-scale simulations without sacrificing model fidelity. It also captures the intrinsic randomness in human driving and enables the incorporation of directly observably statistical CF properties. This paper identifies the best-correlated stimulus-response factors, analyses state machine properties of certain trajectory features and finally shows how several state machines can be hierarchically organised with the subsumption architecture.**

***Index Terms*—microsimulation, Finite State Machine, FSM, timed automata**

I. Introduction

Due to the complex interactions of road participants, simulations rather than analytical approaches are used to tackle traffic planning and network optimisation tasks. These underlying models are commonly classified as macroscopic (traffic flow is considered a stream exhibiting liquid-like properties), mesoscopic (vehicle trajectory data like average velocity or space gap are aggregated) and microscopic (each road user is represented and modelled individually). Building on Differential Equations (DE), the latter can replicate non-linear flow and perturbation phenomena but their increased fidelity comes at the cost of higher computational demand and more parameters to calibrate. Being spatially and temporally discrete, Traffic Cellular Automata (TCA) can replicate driver behaviour but are generally considered being too coarse from a driver's perspective. TCA are very popular in the transportation research community nevertheless, but are not available in commercial simulations and therefore not widely used in practice [1].

The duality of physically unrealistic TCA or fully continuous models with unobservable parameters, dependency on numerical solvers and high computational demand is not a necessity. Some simplistic "hybrid" models with discrete and continuous components were proposed but are feature-wise incomplete. How would a model have to be constructed to meet the following requirements?

- Be computationally efficient.
- Describe variations in driver attention and reaction time.
- Be spatially continuous.
- Contain only parameters relevant for the driving task.
- Extensible with route-choice and lane-changing features.

Our contribution is to develop a CF model which meets the outlined requirements and thus bridges the efficiency/accuracy gap between TCA and continuous microsimulations. From several combinations of responses and stimuli we identify which pairs are most suitable to predict CF behaviour for the next few moments, evaluate the temporal distributions of constant behaviour and describe how quantisation shrinks the model's state space. We also develop an exemplary implementation based on hierarchically organised FSM and a Timed Automaton.

Although our approach for modelling CF behaviour has not been proposed before, several hybrid microsimulations with other objectives and construction principles exist. Together with an introduction to FSM and TA concepts, some semi-discrete approaches are listed in Sec. II. Subsequently, we work out the best stimuli-response correlations which will govern the driver reactions (Sec. III), measure how long drivers maintain the same acceleration (Sec. IV) and investigate the construction of a state-machine representation for acceleration levels (Sec. V). Building on the insights gained, Sec. VI finally provides a detailed description of

Manuscript received July 30, 2018; revised November 6, 2018.

doi: 10.18178/jtle.6.2.25-36

the FSM-TA model. The paper concludes with a summary of the findings; it also discusses potential extensions and improvements within the proposed framework.

II. Related Work

From the earliest attempts in the 1960's until now, hundreds of CF models were proposed and implemented in simulations. Development of new models is mainly driven by better psychological understanding, technological advancements (e.g. cruise control, autonomous driving), improved observation techniques, higher data resolution, as well as the demand to model more features. Independent from their individual scope and objectives, CF models attempt to capture the *driving relation* which is abstracted as a reaction (e.g., adjustments of speed or acceleration) to one or more stimuli perceived from the leading vehicle (hence the name) or other environmental factors. Strength, linearity and time between stimulus and response (*reaction time*) have been modelled from numerous angles. Because vehicular trajectories are spatially and temporally continuous, the most obvious way of representing them mathematically are (sets of) partial or ordinary differential equations (PDE, ODE). For an overview, genealogy and benchmarks over (O)DE-based approaches see sources [2-5] and references therein. Significant research efforts also went into more abstract *minimalist models* which are usually constructed as sets of integer-equations and are evaluated at evenly spaced time steps (*ticks*). These fully discrete Traffic Cellular Automata (TCA) do not suffer from some problems their continuous counterparts plague (stability, scalability, de- pendency on numerical solvers) but suffer from a significantly reduced model fidelity. Due to their abstraction, TCA can be implemented very efficiently, only require few parameters, can standardise topologies and reproduce macroscopic phenomena well [6].

In addition to the fully discrete/continuous models, several variants with varying numbers of discrete and continuous properties were proposed to replicate spatio-temporal inter- actions between dynamical systems. Except for the time- discrete Gipps [7], Krauss [8] and Newell [9] models, most of these approaches were relegated to a niche existence in transportation engineering. They have found wide application in the research investigation particle hopping approaches and the simulation of lattice-based road networks [10], [11]. Some examples are the Coupled Map Lattices [12]-[16], Zero-Range-Processes and Chipping Models [17]-[21], or some of the interacting particle systems [22]-[25].

Also referred to as finite-state automata, FSM are mathematical models of machines which can be in exactly one state at any given time. State changes (*transitions*) are instantaneous and occur in response to external inputs ("reactive system") [26]. In reference [27] they are defined as "abstract model describing a synchronous sequential machine" while reference [28] defines them as a *model of computation* which consists of a set of states ***N***, a start state, an input alphabet, and a transition function. Over time, several FSM variants were used in linguistics, computer science, philosophy, biology, mathematics, and other fields. FSM are highly flexible and some of their adaptations (non-determinism [29], [30], hierarchical structures [31], [32] or multiple outputs (*Mealy machines*) [33]) make them very suitable to represent car- following activities. The downsides of their simplicity are maintainability (transitions of all connected states need to be adjusted when a state is added/removed), scalability (state space explosion and loss of graphical readability for bigger systems) and lack of reusability (strong coupling between states means that parts of a more complex FMS cannot be simply used separately) [27].

One way to deal with the architectural problems of simple FSM is the *subsumption architecture*. It was developed as a control structure to deal with concurrent processes and decomposes "simple independent task-achieving behaviours" into parallel sets of control systems. While controllers operate in parallel from input to output, higher priority controllers *subsume* or (*suppress*) the lower priority ones based on their output conditions. This approach allows to incrementally add features to a model from a small set of "general purpose behaviours" and simplifies debugging and specifying features for individual *modules* which can be capsuled and reused [34]-[36]. Despite its popularity in developing autonomous robots and wide application in agent-based simulations, the subsumption architecture has – to the best of our knowledge – not been used to develop a CF model.

Inside an FSM, transitions may occur with arbitrary but fixed delays without altering the system's behaviour: FSM are invariant under *stretching* of the time axis [37]. As such, FSM are not suited to model temporal aspects of a dynamic system. For this reason, Alur and Dill [38] extended FSM with real-valued clocks and mathematically formalised their dynamics. Similar to FSM, state transitions are annotated with actions (what is done?), guards (when is it done?) and reset sets (which clocks are reset?) [39]. Although not as ubiquitous as FSM, TA are used widely to replicate asynchronous applications and circuits, finding logical errors in communication protocols [40], design pacemakers [41], model Bitcoin contracts [42], mimic home care plans [43], simulate biological pathway dynamics [44] and implement Publish-Subscribe structures [45]. TA transitions can also be stochastic or be triggered by external factors making them a suitable tool to model the human reaction time and attention states [46]. Despite their inherent aptitude, TA have only been used to model particle hopping models [25] but not to replicate more complex CF dynamics.

Mathematically, the TA $\mathbf{A}$ is a tuple with the components $\mathbf{Q}$ (set of states), $\mathbf{\Sigma}$ (alphabet), $\mathbf{C}$ (set of clocks), $\mathbf{E}$ (transitions), and $\mathbf{q_0}$ (initial state). $\mathbf{A}$'s transitions $\mathbf{E}$ are defined as $\mathbf{E} \subseteq \mathbf{Q} \times \mathbf{\Sigma} \times \mathbf{B}(\mathbf{C}) \times \mathbf{P}(\mathbf{C}) \times \mathbf{Q}$ where $\mathbf{B}(\mathbf{C})$ denotes the set of Boolean clock constraints and $\mathbf{P}(\mathbf{C})$ is the power set of $\mathbf{C}$. $\mathbf{\Sigma}$, $\mathbf{Q}$, $\mathbf{C}$ and $\mathbf{E}$ are finite sets; $\mathbf{q_0}$ is one element of $\mathbf{Q}$. The exemplary

edge (**q**, **a**, **g**, **r**, **q'**) defines a transition from q to q_0 with action **a**, guard **g** and clock resets **r**. Like most FSM, **A** has a starting state $\mathbf{q_0}$. Once this is entered, all clocks start to increase with the same speed so their values can be compared with guards to trigger (instantaneous) transitions along the edges.

III. Stimuli and Responses: Correlations and Feature Distributions

To develop a car-following model requires knowledge about the behaviour of drivers: CF decisions should be based stimuli and responses which are highly correlated at some preferably consistent delay. To find these, we employ an experimental dataset which recorded ten drivers following each other in a platoon. The measuring equipment consisted of high-precision RTK GPS receivers and Doppler-effect-based speedometers. The experiment took place on a test track without opposing traffic or interruptions and the platoon's leader performed several driving maneuvers to simulate different traffic conditions. The dataset does not include interrupted facilities nor free- flow conditions but, due to its accuracy, was used in several publications already [3], [47]-[55].

In an exclusive CF situation (no surrounding effects), drivers may base their decisions on space gap g, headway h, relative speed Δv or the lead vehicle's acceleration a_{n+1}. Stimuli-wise, the follower then adjusts speed v_n or acceleration a_n. Are the stimuli all equally important? What is the delay in processing them? These questions are answered by Fig. 1 which contains four subplots (one for each stimulus) and three curves for each (average) driver response. The axis of abscissa indicates how much the respective stimulus and responses were shifted against each other, i.e. their delay (time shift > 0) or anticipation (time shift < 0). On the ordinate axis is plotted how strongly (linear) stimulus and response are correlated measured as Pearson correlation coefficient [56]). Kendall Tau and Spearman correlation coefficients have been calculated, too: results only differ in their scales and are therefore not included. Fig. 1 makes immediately clear, that Δv and g exhibit the highest correlation with acceleration and speed, respectively. The peak for both correlations is 1.5 s.

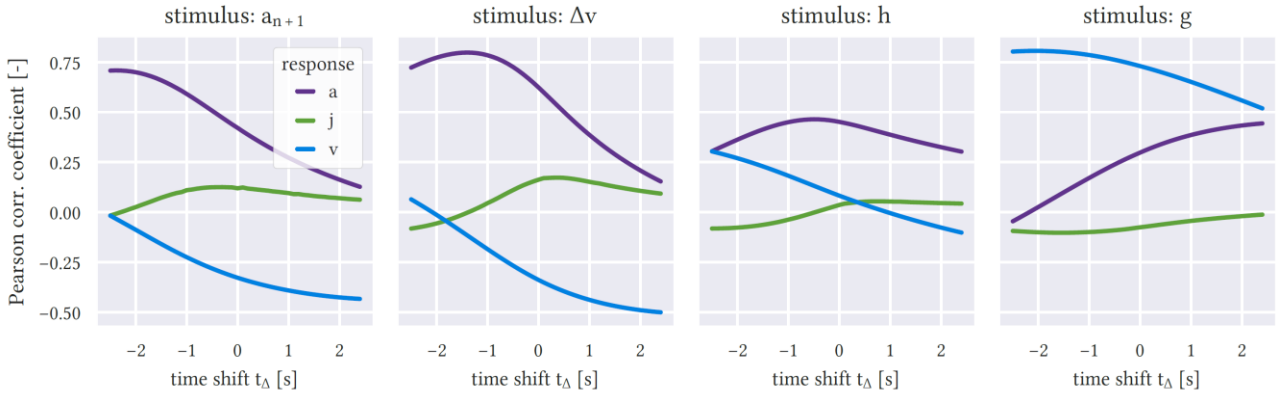


Figure 1. Pearson correlatin for different stimuli and responses shifted in steps of 100ms

Having found the response-stimuli pairs with the highest correlation and the associated total delay (human reaction processing and mechanical lags) we proceed to identify the range of acceleration, headway and relative speed drivers tend to select. Respective Kernel Density Estimations are shown in Fig. 2. All three trajectory feature distributions are highly symmetrical and exhibit distinct maxima. Overall, the 90% quantiles are $a_{90\%}$ = [0.80, +0.83] ms^{-2}, $\Delta v_{90\%}$ = [2.22, +2.20] ms^{-1} and $h_{90\%}$ = [1.25, +3.03] s.

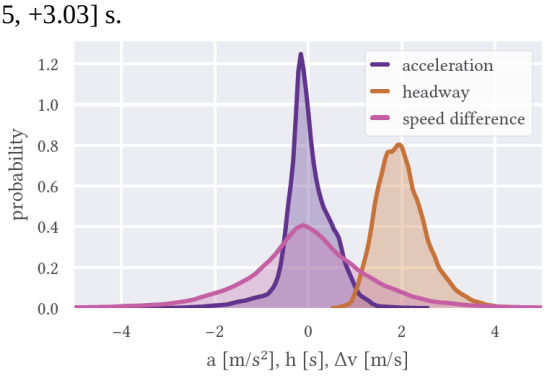


Figure 2. Kernel density estimations for acceleration, headway and speed difference

One key property of microscopic traffic models, is the crash avoidance ($g_{min} > 0$, $h_{min} > 0$). Models should avoid these situations but not render them completely impossible (like early TCA). To determine which minimal headways or space gaps are selected by drivers for a given speed, we quantised all velocities in the dataset with a mid-tread quantiser, collected all space gaps / headways falling into each group and selected the respective minima. Results are shown in Fig. 3. To demonstrate that the quantisation step size ΔQ does not have a major influence, Fig. 3 contains the results with ΔQ = 0.1m and ΔQ = 1.0m. The smaller step size produces more outliers but does not significantly alter the general progression. Interestingly, the minimal space gaps and headways form very different relationships with the follower's speed. As shown in Fig. 3a with thicker yellow line, $g_{min}(v)$ can be approximated with two piecewise linear functions:

$$g_{\min}(v) = \begin{cases} 0.5v + 5 & v \in [0, 25]\,\mathrm{m\,s^{-1}} \\ -10v - 233 & v > 25\,\mathrm{m\,s^{-1}} \end{cases} \quad (1)$$

Using the same technique as described before, we plotted the minimum headways drivers select depending

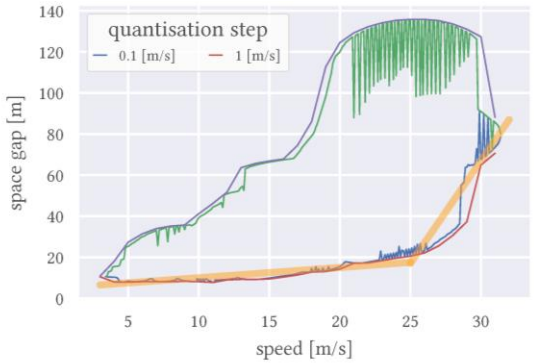
on their own speed. The result is shown in Fig. 3b and unlike g(v), is not suited to be approximated by linear functions. Instead, we estimate the observations with a Beta distribution. With α = β = 0.5, the probability density function was calculated as

$$h_{\min}(v) = \frac{v^{\alpha-1}(1-v)^{\beta-1}}{\mathrm{B}(\alpha,\beta)} \qquad (2)$$

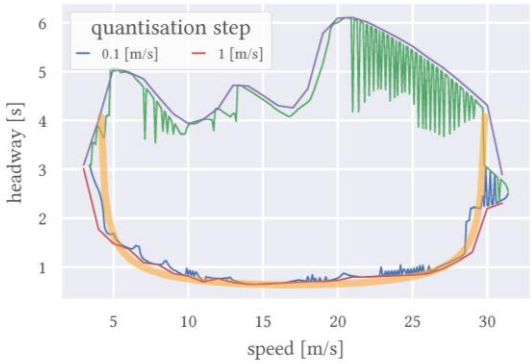
where the denominator is defined as shown in eq. 3.

$$\mathrm{B}(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \qquad (3)$$

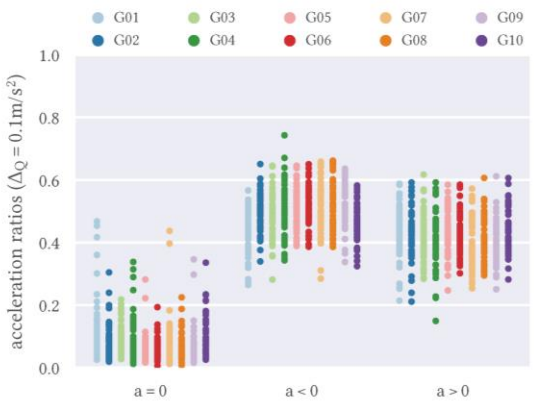
Eq. 1 and Eq. 2 determine the lower thresholds for triggering deceleration transitions in our subsumption architecture CF model. The upper thresholds were calculated, too but are less meaningful because drivers were instructed to follow each other. Usually, the driving behaviour is considered being independent from the leading vehicle for g ≈ 65m (200 ft) [57]. The irregular shape of maximum headways indicates that more data is required to draw final conclusions. Yet, since 95% of all headways in the dataset were <3.03 s (99 %: <3.64 s), 3.5 s is a reasonable threshold for drivers to catch up to their lead vehicle.



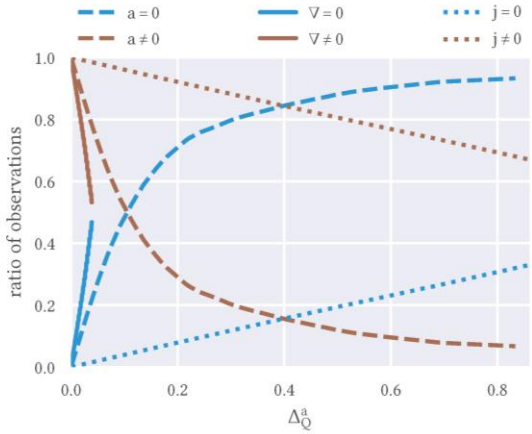
(a) Minimum space gaps as function of speed.



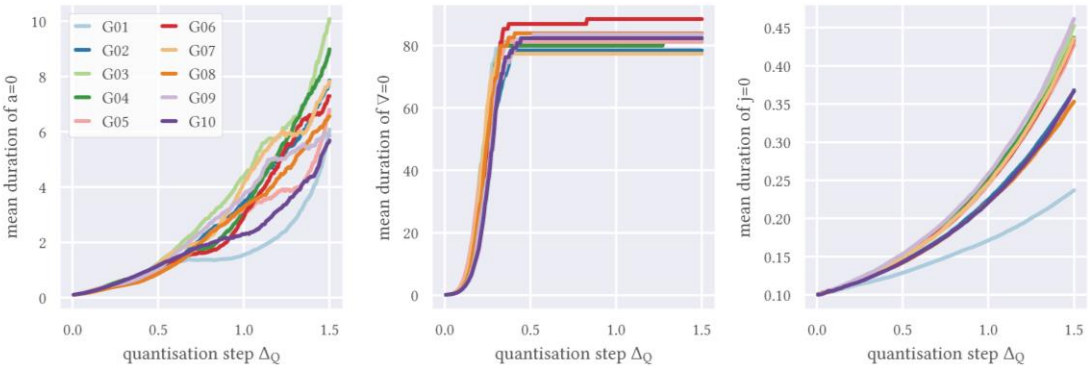
(b) Minimum headways as function of speed.



(c) Shares of positive, negative and 0 occurrences for a after quantisation with a mid-tread quantiser with $\Delta_Q = 0.1\,\mathrm{m\,s^{-2}}$.



(d) Relationship of Δ_Q and the average shares constant a, j and ∇



(e) Mean duration of a, j and $\nabla = 0$ plotted against $\Delta_Q \in \{0 \ldots 1.5\}$.

Figure 3. By binning the measured speeds the minimum space gaps and headways for each bin can be determined

IV. Acceleration Properties

Sec. III showed that acceleration and speed correlate highly with Δv and g. Since the proposed microsimulation model shall have better precision than TCA, drivers must be able to adjust their vehicles' acceleration instead of hopping between different velocity levels. To better understand how drivers control their acceleration, this section analyses how long they maintain constant levels of acceleration and shows the ratios of deceleration, constant speed and acceleration. To obtain these details, we quantise the dataset and visualise how this process affects model fidelity.

All values stored and processed in a digital computer are by definition discrete, i.e. stem from a finite set of values. With sufficiently fine resolutions, continuity can be approximated. At a sampling rate of 10 Hz, temporal discretisation of the dataset is already suitable but the resolution of **a** is so high that repeated observations of the same value are highly unlikely. For this reason, it cannot be measured how long drivers maintain a constant speed because the likelihood of *a=0* approaches zero. In fact, the dataset does not contain any observation with *a=0*. The solution to this is quantisation, i.e. the process of converting an object in the continuum into a finite bit stream. It is a "non-linear, irreversible and memoryless many-to-few mapping" whose quality depends on the number of quantization levels L and thus on the quantisation step size Δ_Q [58]. For sake of simplicity and practicality (symmetry around the origin, inclusion of zero in the discrete state space) we employ a *mid-tread quantiser*. Its mathematical formulation is shown in eq. 4 [59].

$$Q_{\text{mid-tread}}(x) = \lfloor 2^{M-1}x + 0.5 \rfloor / 2^{M-1} \qquad (4)$$

By measuring the duration of consecutive movements for each vehicle-driving pattern combination and classifying the results as positive, negative or zero, the nature of consecutive movements can be explored more deeply. Because the recording time varies for individual trajectories, shares of acceleration, deceleration and zero movement ought to be expressed as ratios to the individual total. The shares of acceleration, deceleration and constant movement are shown in Fig. 3c for ΔQ = 0.1ms^{-2}. One dot in each strip plot marks the acceleration percentage for each driver per driving pattern. Albeit not visually obvious due to overlapping marks, all data points from each group add up to 100 %. Other visualisations eliminate this shortcoming but fail to convey the information as efficiently. Fig. 3c shows, that drivers spend approximately the same share of time with speeding up, breaking and maintaining the same speed. Driver tt01 (platoon leader) spent the most time at the same speed – apparently an artefact from trying to reproduce the different speed patterns. While individual drivers differ and can be identified by their behaviour, the shares of *a>0, a<0 and a=0* exhibit similar patterns.

The acceleration ratios depicted in Fig. 3c are just a snap- shot with $\Delta_Q = 0.1\text{ m s}^{-2}$. With increasing Δ_Q, the number of values falling into the *a*=0 bin increases, while the percentage in the other categories shrinks. Smaller Δ_Q, on the other hand, decrease the bin width for *a*=0, thus having the opposite effect by reducing the number of acceleration levels and increasing the share drivers move with constant speed, acceleration or steering angle. Fig. 3d depicts this relationship for the three features by plotting the average share of zero and non-zero a, j, and ∇ against Δ_Q. The ratio of positive and negative is visually extremely similar and were therefore summarised in one group. The curves in this chart are symmetric to *y=0.5* because *0* and *≠0* observations must sum up to 1. While consecutive observations of and j are linearly correlated with Δ_Q, larger Δ_Q increase the share of observations with *a=0* near-linearly up to Δ_Q=0.15ms^{-2} before the slope significantly decreases and then approaches 100% in a log-like fashion. Under the prevailing conditions, >95% of all accelerations are consecutive for $\Delta_Q = 0.8\text{ms}^{-2}$. Not shown is the plot of inter-driver differences, but as can be deducted from Fig. 3e, variations between drivers are not very pronounced for $\Delta_Q < 0.6$. Thus, Δ_Q is a suitable reference for the expected vehicular dynamics of a discrete acceleration microsimulation.

While the share of $a = j = \nabla = 0$ gives a good indication about the general microscopic system dynamics as function of Δ_Q, it does not predict how Δ_Q affects the mean duration. This is shown on a per-vehicle basis in the three subplots of Fig. 3e. The selected range of $\Delta_Q \in \{0 \ldots 1.5\}$ is representative for all three features but remains well below the quantisation step size of most TCA ($\Delta_Q = 7.5\text{ms}^{-1}$ in [60]). Steering direction and jerk were included to provide some context and demonstrate the applicability of the approach for other trajectory features. In the leftmost subplot, $\Delta_Q \in \{0, 0.6\text{ms}^{-1}\}$ only shows minor differences between drivers while $\Delta_Q > 0.6\text{ m s}^{-1}$ fans out the average time spent without changing speeds in a highly non-linear fashion. Similar trends are exhibited in the rightmost subplot but with a much smoother progression and 20-fold smaller scale. The mean duration of a_{G01} and j_{G01} grows slowest by a significant margin with Δ_Q. This may result from the driver's objective to change speed regularly while the followers do not have to react immediately but can maintain their speed slightly longer. This is not necessarily the case as shown by driver **G10** (the last in the platoon) who also keeps accelerating/braking despite having the most information about the traffic conditions ahead. The middle plot of Fig. 3e has a very different dynamic: the mean time vehicles move straight ahead increases sharply with Δ_Q and then remains almost constant. Increasing Δ_Q beyond $0.4^{\circ}/s$ does not change the system dynamics anymore; resolution becomes so coarse that vehicles do not change direction (in a model of that resolution).

The previous paragraphs showed that, $\Delta_Q < 0.6\text{ms}^{-2}$ affects all drivers similarly. But which discrete acceleration levels are actually be detected by human drivers? Without visual clues, speed cannot be felt at all but acceleration (linear and rotational) and its change,

can. Perception of acceleration mostly depends on the vestibular system: without it, detecting rotational stimuli and linear motion is still possible through cutaneous and other somaesthetic mechanoreceptors but loses precision in the order of a magnitude [61]. In [62] thresholds of 0.03m s^{-2} and 0.06m s^{-2} were observed for horizontal and vertical movements, respectively. These values were collected with sinusoidal linear oscillation at 0.03Hz [62]. Because perceptions change with the frequency of oscillation, thresholds vary. Participants in the experiment described in [61] exhibited differing perceptions but on average were able to perceive accelerations of $\approx 0.02 ms^{-2}$. In [63], vestibular observation thresholds of $0.121 ms^{-2}$ (linear ramp $j = 0.028 ms^{-3}$), $0.192 ms^{-2}$ (linear ramp $j = 0.079\ ms^{-3}$), $0.048\ ms^{-3}$ ($j>>0$) and $0.167 ms^{-2}$ (parabolic ramp $j = 0.0152 ms^{-3}$) were observed. The authors conclude that large j facilitate motion perception. In reference [64] the threshold for detecting angular rotation was determined to be 0.12°/s. These thresholds form the lower bounds for designing a discrete-acceleration microsimulation; it is reasonable to assume that drivers accelerate faster than their own perception threshold.

V. An Acceleration State Machine

It is the nature of quantisation to reduce a continuous, uncountable state space to a smaller, enumerable set of symbols. This section describes the properties of an *acceleration state machine*: the number of states as function of Δ_Q and how drivers switch between levels. When a continuous range of numbers is discretised through quantisation and sampling, its individual members can be identified and counted. That also means we can consider individual values as start and endpoints for actions in the acceleration state machine; their symbols become a graph which maps relations between objects (*nodes, vertices, points*). Transitions between nodes are termed *edges* (arcs or links) and the *degree* of a node denotes how many edges connect to it [65]. Due to the uniform quantisation, distances between all nodes are constant. Graphs are one option to represent state machines and can abstract the transitions between states. Because acceleration oscillates around zero and is only governed by driver actions, we can use the state machine abstraction to reason about human driving behaviour. The gained insights will also be incorporated into the design of the proposed microsimulation.

While the average duration drivers spend in states of a, j and increases with Δ_Q, the state space of each trajectory feature shrinks. The details of this inverse relationship are presented in Fig. 4; it shows the averaged number of states for a, j and in dependence of Δ_Q. The general pathway for all three features and their DCP is similar with the notable noise which likely stems from recording inaccuracies and small relative changes of the feature itself. For an acceleration state machine with a coarseness in the range of $0.1 ms^{-\Lambda 2}$ to $0.6 ms^{-2}$, between 15 and 75 states are needed to represent the different acceleration levels. By selecting Δ_Q appropriately, computational complexity, output precision and model fidelity can be balanced.

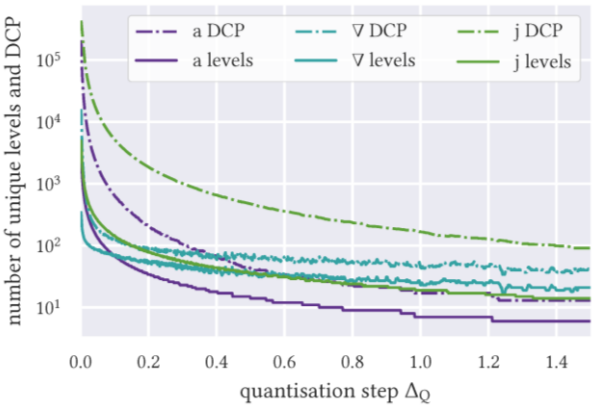


Figure 4. Number of states and DCP as function of Δ_Q

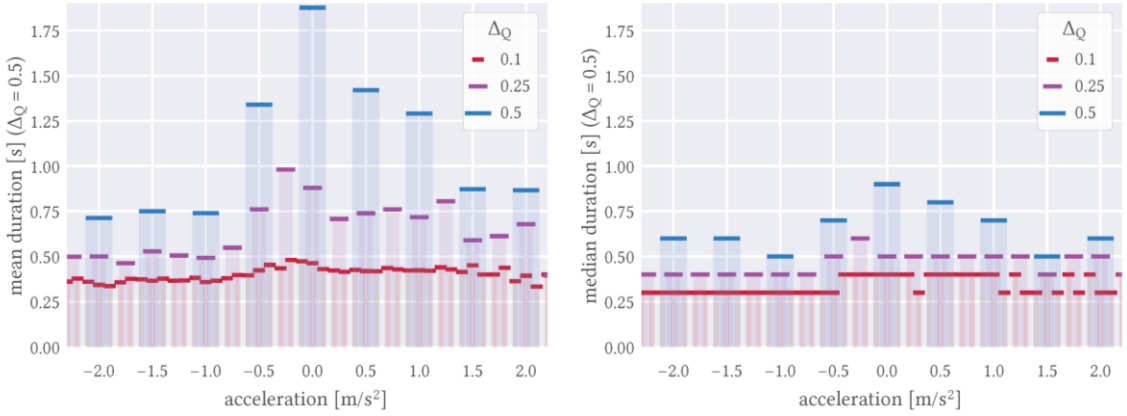
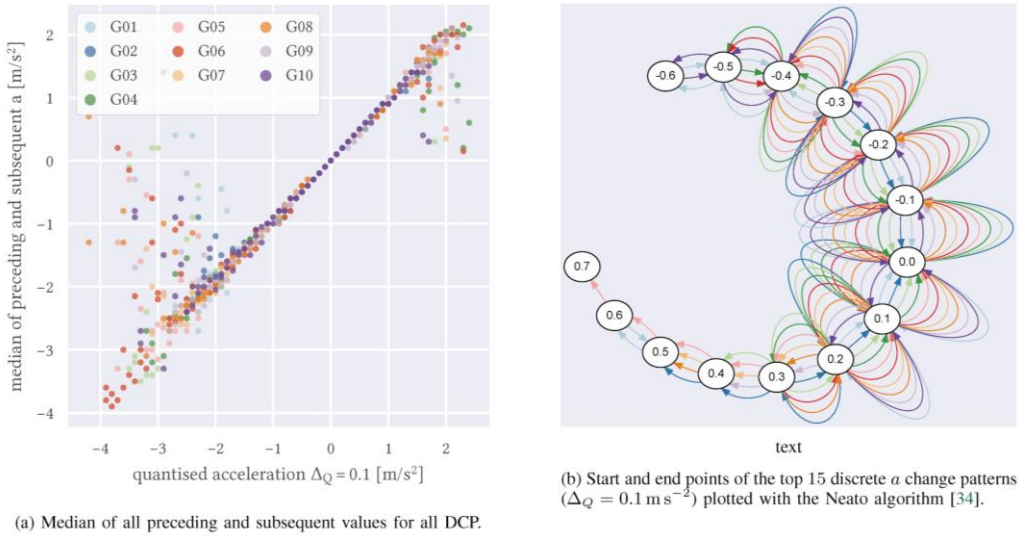


Figure 5. Mean and median duration for different of acceleration levels and three Δ_Q

The second property shown in Fig. 4 is the number of *discrete change primitives* (DCP). Organised as 2-tuples, DCP denote the start and end point for each action in the quantised trajectory data. The total number of DCP is significantly lower than the cross product of all unique values and describes driver behaviour in *states*. This is because vehicular acceleration cannot change from e.g. $3.5 ms^{-2}$ to $1.9 ms^{-2}$ unless the chosen Δ_Q is large. In our model, the driver's attention is separately modelled from the accelerations by a Timed Automaton. But while it has

been shown that acceleration/deceleration are distributed similarly, driver heterogeneity for the mean duration of $a = 0$ is negligible for Δ_Q=0.6 and the number of states and DCP as determined, too. Still missing is the variability of the mean duration of individual acceleration. The left plot of Fig. 5 shows these distributions for Δ_Q = 0.10, Δ_Q=0.25 and Δ_Q=0.50. Peaks are very differently pronounced for the different quantisation steps: Δ_Q=*0.1* produces a ragged shape with numerous local minima and maxima while Δ_Q=*0.5* may be approximated with a skewed but unimodal distribution. A very different picture is produced when medians instead of means are used to measure the central tendency (right plot of Fig. 5): peaks are less pronounced and the median duration is shorter. A visual analysis of durations for individual acceleration levels revealed exponential distributions across for all values and selected Δ_Q. The TA governing the durations of constant acceleration should therefore be probabilistic with an underlying exponential distribution.



(a) Median of all preceding and subsequent values for all DCP.

(b) Start and end points of the top 15 discrete a change patterns ($\Delta_Q = 0.1\,\mathrm{m\,s^{-2}}$) plotted with the Neato algorithm [34].

Figure 6. How do drivers iterate through a finite set of discrete acceleration levels?

As last step and before finally introducing the model, we want to analyse the order in which acceleration is selected by drivers. Fig. 6b represents the *transitions* between all quantised acceleration values contained in the Tomakomai dataset (mid-tread quantiser, Δ_Q = *0.1*). The plot was created with Graphviz [66], [67] and, more specifically, using the Neato algorithm which runs an iterative solver to find low energy configurations [68]. In accordance with Fig. 4 where the number of unique DCP drops quickly before slowly decreasing further, adjusting Δ_Q does neither change the number of nodes nor the general graph. The *directed edges* in Fig. 6b are colour-coded and represent individual drivers; nodes indicate the discrete a values which are contained in the 15 most frequent acceleration DCP. Due to the differing DCP per driver, the total number of nodes is 15. Fig. 6b shows a strong order of acceleration preferences; drivers tend to maintain $a \approx 0 ms^{-2}$ and iteratively move to acceleration levels further from 0.

At a sampling rate of 100ms and given that acceleration changes continuously with time, this is expected. Not shown in Fig. 6b are loops, i.e. when drivers maintain the same acceleration. How frequently and for how long this happens, was discussed in the previous paragraphs already. Nevertheless, Fig. 6b is incomplete because it only contains the top 15 DPC. This shortcoming is fixed by the complimenting view on the state machine perspective provided by Fig. 6a which plots all quantised acceleration values against the median of their respective preceding and succeeding states. Medians were chosen over averages to maintain the same state space for predecessors, successors and reference points. The straight line shows that drivers tend to maintain their current acceleration for $a \in \{1.5, +1.5\} ms^{-2}$. In other words, in this range drivers behave very predictably and only make minor adjustments. As pointed out in Sec. IV, >90% of all acceleration activity falls into that range and is therefore a suitable target for modelling. The larger number of outliers in the deceleration range is also not surprising: decelerating when approaching the leader is much more important than accelerating when the lead vehicle is driving away. Furthermore, cars tend to be equipped with brakes much more powerful than the engine and thus decelerating can take place at more extreme values than accelerating the car. Δ_Q=0.1ms^{-2} was selected because it is near the human perception threshold but still large enough to mainly contain driver-intended acceleration changes. Increasing or decreasing Δ_Q does not significantly alter the linear relationship; it only changes the distance between observations.

VI. Exemplary Implementation

This section finally introduces the hierarchical subsumption model. It consists of Timed Automaton which governs the driver's attention, individual Finite State Machines which react to headways ($\mathbf{A}_{hw}$: Fig. 10), to the relative speed ($\mathbf{A}_{\Delta v}$: Fig. 9) and the space gap for crash avoidance ($\mathbf{A}_{break}$: Fig. 11). The other fundamental parts of our CF model are the Timed Automaton $\mathbf{A}_{TA}$ which governs the driver's attention (Fig. 8) and the FSM

$\mathbf{A}_{acc}$ which switches between different discrete acceleration levels (Fig. 12). Fig. 7 depicts the relationship of all components in the pro- posed CF model. On the left side, the Timed Automaton $_{TA}$ controls the times between actions of the drivers. Unless there is feedback from the break light of the leader, the automaton selects times of "inactivity" based on the distributions worked out in the preceding sections. The perception of the brake light plays a major role in reducing the reaction time [69] and also the time between driver actions. This separation between the attention modelling and acceleration representation has the advantage that some time-related properties can be formally verified [38]. The third component of the subsumption architecture shown in Fig. 7 are the *modules*. They can be composed of more complex tasks or activities, but we intentionally only employ hierarchically organised state machines for letting the driver react to certain CF stimuli. The notion of hierarchy is visualised by the stair-like arrows: the lead vehicles' break light increases attentiveness in the follower but does not directly trigger a driver reaction (no step). $\mathbf{A}_{\Delta v}$, on the other hand, can cause the acceleration to change by $1\Delta_Q$ unless this decision is overruled by A_h which would trigger if the headway was too small or the driver was more than 3.5s behind their leader. While deceleration is assumed to be always a safe action, an acceleration decision from too large headways or speed differences would be dominated by $\mathbf{A}_{max}$ if the vehicle already exceeded the speed limit. The acceleration decision is finalised by $\mathbf{A}_g$ which checks if the space gap undercuts g_{min} as defined in eq. 1. The change of acceleration Δa coming from one of the parallelly executed state machines is used as input for $\mathbf{A}_{acc}$. The state space of Δa is {-1, 0, 1}. In consequence, acceleration cannot jump more than $1\Delta_Q$ in one *attention cycle*. This is physically accurate and extreme decelerations have to be captured within A_{TA}: by repeatedly iterating through the whole decision cycle the extremes of $\mathbf{A}_{acc}$ can be reached. The advantages are that the physical reality is captured well and acceleration cannot exceed the limits imposed on $\mathbf{A}_{acc}$. Neither numerical solvers nor stability analyses as commonly performed for models based on differential equations are required.

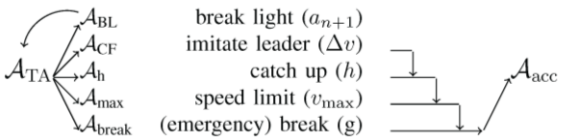


Figure 7. The complete CF model with all automata who govern the driver decisions and longitudinal vehicle dynamics.

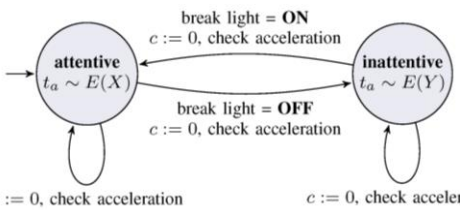
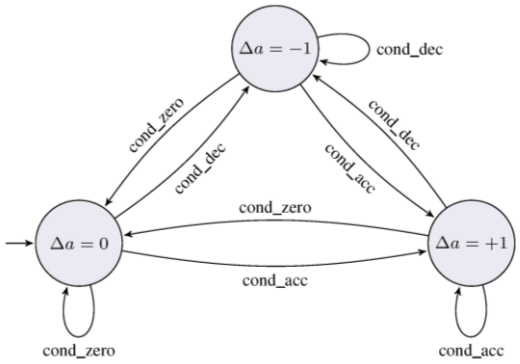
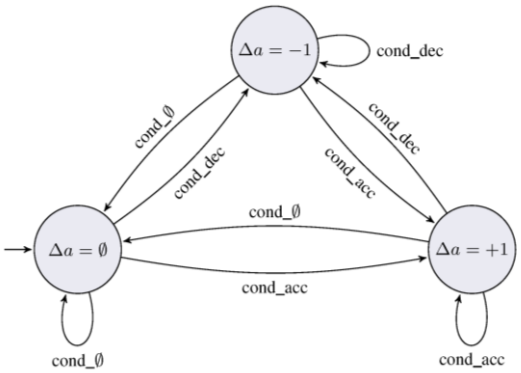


Figure 8. $\mathbf{A}_{TA}$ draw durations of constant acceleration and trigger a check when the time is exceeded.



condition	
cond_acc	$\Delta v > 1\,\mathrm{m\,s^{-1}}$
cond_dec	$\Delta v < -1\,\mathrm{m\,s^{-1}}$
cond_zero	$-1\,\mathrm{m\,s^{-1}} \leq \Delta v \leq 1\,\mathrm{m\,s^{-1}}$

Figure 9. FSM to imitate the leading vehicle by synchronizing speeds



condition	
cond_acc	$h > 3.5\,\mathrm{s}$
cond_dec	$h < h_{min}$ (eq. 2)
cond_∅	$h_{min} \leq h \leq 3.5\,\mathrm{s}$

Figure 10. $\mathbf{A}_{hw}$: Catch up if $h \geq 3.5\,\mathrm{s}$, slow down if h, is smaller than the minimum headway defined in eq. 2, or else do not overrule any previously made acceleration decisions

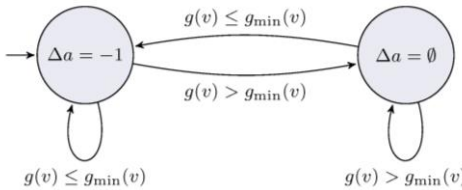


Figure 11. $\mathbf{A}_{break}$: Slow down if g is smaller than the minimum space gap defined in eq. 1

Fully discrete traffic models are usually designed around the notion of *time steps* (*ticks*) at which, in a synchronised fashion, all simulated vehicles attempt to move forward or carry out other driving activities. Continuous microsimulations, on the other hand, are iterated with numerical solvers at a frequency of usually 10Hz [70]. Our approach, as shown in Fig. 8, differs in this respect because durations of constant acceleration are repeatedly drawn from two continuous distributions. Building on the insights gained in previous sections, these *acceleration times* $\mathbf{t}_a$ are exponentially distributed with a mean/median that can be estimated from Fig. 5. Since

the ratios presented there contain all observations, i.e. attentive and inattentive states, the expectation value E of the distributions X and Y will have to be determined by a parameter analysis or a more specialised statistical study. The driver attention TA $\mathbf{A}_{TA}$ is nevertheless equipped with two states: When the brake lights of vehicle $n+1$ are on, the mean/median is lower and drivers are more attentive. As a result, stronger acceleration values are achieved by quickly iterating through the states of $\mathbf{A}_{acc}$. Each transition of $\mathbf{A}_{TA}$, the internal clock c is reset and another cycle through $\mathbf{A}_{acc}$ and the preceding CF automaton is triggered ("check acceleration"). The change between $\mathbf{A}_{TA}$'s two states is activated through feedback from the FSM $\mathbf{A}_{BL}$ which is executed in parallel with the other CF FSM. Since the model has to go through a full cycle before deceleration sets in and acceleration cannot jump arbitrarily, the average cycle time and the number of cycles it takes to reach the more extreme states of $\mathbf{A}_{acc}$ is equivalent to the total reaction time t_{rct}. $\mathbf{A}_{CF}$ as visualised in Fig. 9 is the first FSM which directly influences a driver's acceleration behaviour; it takes the relative speed as input and decides to either increase/decrease acceleration by one level or maintain the current level of acceleration. The result is indicated by the state $\mathbf{A}_{CF}$ is in and therefore can attain the values $\Delta a = +1$, $\Delta a = 1$ or $\Delta a = 0$. The thresholds which are labelled on the edges and trigger acceleration or deceleration are 1ms^{-1} and 1ms^{-1}, respectively. Any value $\Delta v \in \{-1,0,1\}\text{ms}^{-1}$ does not lead to a change in acceleration.

To underline our model's simplicity and modularity, $_h$ demonstrates that a FSM based on headway in principle has the same structure like $\mathbf{A}_{CF}$ (Fig. 10). The only difference is the table with conditions: when the headway is longer than 3.5s, drivers tend to catch up if they want to follow their leader (≈98% in the Tomakomai dataset) thus $\mathbf{A}_h$ goes to the $\Delta a = +1$ state. The minimum headway follows a Beta distribution (Fig. 3b) based on the speed. If the current headway falls below the associated minimum headway (h_{min}), $\mathbf{A}_h$ terminates in the $\Delta a = +1$ state. All headways in-between do not alter the acceleration change suggested by $\mathbf{A}_{CF}$. To implement this behaviour in the top-down structure of the TA-FSM model, we introduced the *empty set* which indicates that previous acceleration choices are not overridden as would be the case with an integer value. Like Fig. 9. Fig. 10 is a *complete automaton*: transitions can take place directly between all states. Crash avoidance has the highest priority and therefore has the power to overrule all previous acceleration decisions. If the current space gap is smaller than the piecewise linear function of Fig. 3a, $\mathbf{A}_{break}$ (Fig. 11) ends in the deceleration state ($\Delta a = 1$). If leader and follower are far enough apart, acceleration/deceleration decisions from $\mathbf{A}_{CF}$, $\mathbf{A}_h$, and $\mathbf{A}_{max}$ are not altered. The break light FSM and the speed limit FSM are not shown but follow the same design principle like $\mathbf{A}_{break}$ (Fig. 11): they a two-state FSM triggered by one binary condition. If the vehicle's current speed is higher than the speed limit the driver then $\Delta a = 1$. In the same fashion, $\mathbf{A}_{BL}$ triggers a switch to a different attention-time distribution inside A_{TA}. Based on these simple mechanisms, we can also check for moving backwards, stopping for traffic signals and turning at intersections without changing the model's general construction.

The last component in our TA FSM car-following model is the $\mathbf{A}_{acc}$ FSM which implements the actual driving states. $\mathbf{A}_{acc}$ may contain an arbitrary but finite number of states (depending on Δ_Q) but a key feature is that drivers only switch iteratively though the states instead of jumping arbitrarily between acceleration levels. An exemplary implementation is shown in Fig. 12. The selected acceleration is maintained for the duration given by $\mathbf{A}_{TA}$. All dependent trajectory features can then be calculated from the time for which a constant acceleration is maintained.

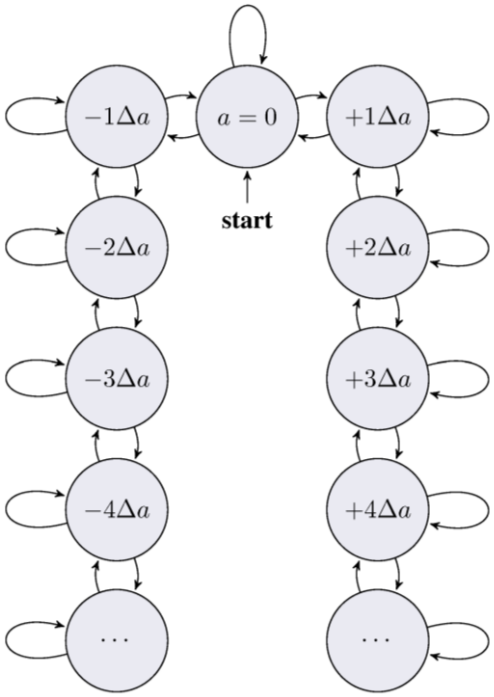


Figure 12. Acceleration state machine $\mathbf{A}_{acc}$

VII. Discussion and Concluding Remarks

In this article we statistically analysed car-following behaviour with the goal to develop a spatially and temporally continuous microsimulation with discrete levels of acceleration. To that end, we analysed the influence of delay on the (Pearson) correlations of different stimuli and responses, measured the share of acceleration, deceleration, and constant speed phases and identified the distributions of the key stimuli and responses. Our main contribution was the statistical description of the potential state machine: how many (acceleration) states and change patterns exist? How long do drivers remain in one state? How do they iterate through the states? These questions were answered in several plots and formed the basis for the proposed TA-FSM model. This microsimulation controls the durations between the (instantaneous) acceleration changes with a

Timed Automaton and organises the stimuli processing with separate finite state machines with a subsumption architecture. Thanks to its modularity, it is easily extendible with lane-changing and route-choice features. The proposed model is not dependent on numerical solvers and for this reason computationally very efficient. Model fidelity can be adjusted through the quantisation step size Δ_Q; blends between the coarse Traffic Cellular Automata and approaches based on differential equations can be selected. Other advantages are the modularity of the concept, the inherent randomness, the focus on observable variables and the lack of *action points* [71], [72].

Although a prototypical implementation exists and already yielded promising results, the description of the calibration process and the role of driver heterogeneity will have to be covered in a future publication. Further research is also necessary with the exact specification of interfaces between the different FSM. Also, the basic ideas could be implemented with Hybrid Automata, a formalism frequently used in Computer Science, Neurobiology, and other domains can resemble the model structure in a simpler way. Hybrid Automata have the advantage of enabling mathematical proofs despite the model's inherent complexity. In fact, Hybrid Automata have been employed to represent vehicular traffic already [73]-[76]. Another way to structure the switching between different tasks is the recently proposed approach of *Behavioural Trees* (BT). They achieve modularity and reactivity by composing complex tasks from simpler tasks without requiring implementation specifics of the subtasks. The notion of tasks (instead of *states*) is one difference of BT and the above-mentioned hierarchical FSM. The other is that BT replace the one-way control transfers of FSM (transitions) with two-way control transfers (function calls). Despite their advantages, BT are mostly used in different Artificial Intelligence tasks, robotics and the development of computer games until now [77], [78]. A BT implementation of our TA-FSM model would have the advantage that the attention TA could be more expressively couple reaction time and subsequent actions (e.g. to represent higher attentiveness during overtaking manoeuvres and shorter attention spans while waiting in jammed conditions). Unlike Hybrid Automata, BT have not been used to microscopically model traffic.

REFERENCES

[1] M. Treiber and A. Kesting, "Cellular automata," in *Traffic Flow Dynamics*, Springer Berlin Heidelberg, January 2013, pp. 225–238.

[2] M. Brackstone and M. McDonald, "Car-following: A historical review," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 2, no. 4, pp. 181–196, December 1999.

[3] P. Ranjitkar, T. Nakatsuji, and M. Asano, "Performance evaluation of microscopic traffic flow models with test track Data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1876, pp. 90–100, January 2004.

[4] F. van Wageningen-Kessels, H. van Lint, K. Vuik, and S. Hoogendoorn, "Genealogy of traffic flow models," *Euro Journal on Transportation and Logistics*, April 2014.

[5] K. Aghabayk, M. Sarvi, and W. Young, "A state-of-the-art review of car-following models with particular considerations of heavy vehicles," *Transport Reviews*, vol. 35, no. 1, pp. 82–105, January 2015.

[6] S. Maerivoet and B. D. Moor, "Cellular automata models of road traffic," *Physics Reports*, vol. 419, no. 1, pp. 1–64, November 2005.

[7] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B: Methodological*, vol. 15, no. 2, pp. 105–111, April 1981.

[8] S. Krauß, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," PhD thesis, University of Cologne, Cologne, 1998.

[9] G. F. Newell, "A simplified car-following theory: A lower order model," *Transportation Research Part B: Methodological*, vol. 36, no. 3, pp. 195–205, March 2002.

[10] F. Lehmann, P. S. Roop, and P. Ranjitkar, "Microscopic models of vehicular traffic: A classification by degree of discreteness," *Transportmetrica B: Transport Dynamics, Submitted to Journal for Review*, 2018, p. 22.

[11] F. Lehmann, P. Ranjitkar, and P. S. Roop, "A systematic review of discrete lattice models for interrupted traffic," *Physica A: Statistical Mechanics and its Applications*, 2018.

[12] S. Yukawa and M. Kikuchi, "Coupled-Map modeling of one-dimensional traffic flow," *Journal of the Physical Society of Japan*, vol. 64, no. 1, pp. 35–38, January 1995.

[13] S. Yukawa and M.Kikuchi, "Density fluctuations in traffic flow," *Journal of the Physical Society of Japan*, vol. 65, no. 4, pp. 916–919, April 1996.

[14] S. Tadaki, M. Kikuchi, Y. Sugiyama, and S. Yukawa, "Coupled map traffic flow simulator based on optimal velocity functions," *Journal of the Physical Society of Japan*, vol. 67, no. 7, pp. 2270–2276, July 1998.

[15] S. Tadaki, M. Kikuchi, Y. Sugiyama, and S. Yukawa, "Noise induced congested traffic flow in coupled map optimal velocity model," *Journal of the Physical Society of Japan*, vol. 68, no. 9, pp. 3110–3114, September 1999.

[16] M. E. Fouladvand, Z. Sadjadi, and M. Shaebani, "Characteristics of vehicular traffic flow at a roundabout," *Physical Review E*, vol. 70, no. 4, October 2004.

[17] R. D. Vigil, R. M. Ziff, and B. Lu, "New universality class for gelation in a system with particle breakup," *Physical Review B*, vol. 38, no. 1, pp. 942–945, July 1988.

[18] J. Kaupuzs and R. Mahnke, "A stochastic multi-cluster model of freeway traffic," in *Traffic and Granular Flow '99*, Dirk Helbing, Hans J. Herrmann, Michael Schreckenberg, and Dietrich E. Wolf, Eds., Springer Berlin Heidelberg, January 2000, pp. 449–454.

[19] E. Levine, G. Ziv, L. Gray, and D. Mukamel, "Traffic jams and ordering far from thermal equilibrium," *Physica A: Statistical Mechanics and Its Applications*, vol. 340, no. 4, pp. 636–646, September 2004.

[20] J. Kaupuzs, R. Mahnke, and R. J. Harris, "Zero-range model of traffic flow," *Physical Review E*, vol. 72, no. 5, November 2005.

[21] J. Kaupuzs, R. Mahnke, and R. J. Harris, "Metastability of traffic flow in zero-range model," in *Traffic and Granular Flow'05*, A. Schadschneider, T. Poeschel, R. Kuehne, M. Schreckenberg, and D. E. Wolf, Eds., Springer Berlin Heidelberg, January 2007, pp. 461–466.

[22] P. Hrabak and M. Krbalek, "Distance and time headway distribution for totally asymmetric simple exclusion process," *Procedia - Social and Behavioral Sciences*, vol. 20, pp. 406–416, January 2011.

[23] J. G. Brankov, N. C. Pesheva, and N. ZhBunzarova, One Dimensional Traffic Flow Models: Theory and Computer Simulations, March 2008.

[24] C. Arita, M. E. Foulaadvand, and L. Santen, "Signal optimization in urban transport: A totally asymmetric simple exclusion process with traffic lights," *Physical Review E*, vol. 95, no. 3, March 2017.

[25] F. Lehmann, P. S. Roop, and P. Ranjitkar, "Modelling particle hopping models for road traffic with timed automata," *Physical Review E*, 2018.

[26] Y. Hardy and W. H. Steeb, "Finite state machines," in *Classical and Quantum Computing*, Birkhaeuser, Basel, 2001, pp. 229–250.

[27] Z. Kohavi and N. K. Jha, "Switching and finite automata theory," Cambridge University Press, Cambridge, UK: New York, 3rd edition, 2010.

[28] National Institute of Standards and Technology (NIST). Finite State Machines, May 2008.

[29] E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R. C. Carrasco, "Probabilistic finite-state machines - part I," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1013–1025, July 2005.
[30] E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R. C. Carrasco, "Probabilistic finite-state machines - part II," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1026–1039, July 2005.
[31] A. Girault, Bilung Lee, and E. A. Lee, "Hierarchical finite state machines with multiple concurrency models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 6, pp. 742–760, June 1999.
[32] F. Bordeleau, J. P. Corriveau, and B. Selic, "A scenario-based approach to hierarchical state machine design," in *Proc. Third IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, 2000, pp. 78–85.
[33] G. H. Mealy, "A method for synthesizing sequential circuits," *Bell System Technical Journal*, vol. 34, no. 5, pp. 1045–1079, September 1955.
[34] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal on Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.
[35] R. A. Brooks, "A robust programming scheme for a mobile robot," in *Proc. NATO Advanced Research Workshop on Languages for Sensor-Based Control in Robotics*, London, UK, UK, 1987, pp. 509–522.
[36] R. A. Brooks and J. H. Connell, "Asynchronous distributed control system for a mobile robot," in *Proc. International Society for Optics and Photonics*, February 1987, vol. 0727, pp. 77–85.
[37] A. Rabinovich, "Automata over continuous time," *Theoretical Computer Science*, vol. 300, no. 1, pp. 331–363, May 2003.
[38] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, April 1994.
[39] K. G. Larsen, B. Steffen, and C. Weise, "Continuous modeling of real-time and hybrid systems: From concepts to tools," *International Journal on Software Tools for Technology Transfer*, vol. 1, no. 1-2, pp. 64–85, December 1997.
[40] R. Alur, "Timed automata," in *Computer Aided Verification, Lecture Notes in Computer Science*, pp. 8–22. Springer, Berlin, Heidelberg, July 1999.
[41] S. Pinisetty, P. S. Roop, S. Smyth, N. Allen, S. Tripakis, and R. Von Hanxleden, "Runtime enforcement of cyber-physical systems. ACM trans," *Embed. Comput. Syst.*, vol. 16, no. 5, pp. 178:1–178:25, September 2017.
[42] M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek, "Modeling bitcoin contracts by timed automata," in *Formal Modeling and Analysis of Timed Systems, Lecture Notes in Computer Science*, Springer, Cham, September 2014, pp. 7–22.
[43] K. Gani, M. Bouet, M. Schneider, and F. Toumani, "Using timed automata framework for modeling home care plans," in *Proc. International Conference on Service Science*, May 2015, pp. 1–8.
[44] S. Schivo, J. Scholma, B. Wanders, R. A. U. Camacho, P. E. van der Vet, et al., "Modeling biological pathway dynamics with timed automata," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 3, pp. 832–839, May 2014.
[45] V. Valero, G. Dıaz, and M. E. Cambronero, "Timed automata modeling and verification for publish-subscribe structures USING distributed resources," *IEEE Transactions on Software Engineering*, vol. 43, no. 1, pp. 76–99, January 2017.
[46] N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Groesser, and M. Jurdzinski, "Stochastic timed automata," *Logical Methods in Computer Science*, vol. 10, no. 4, December 2014.
[47] G. Gurusinghe, T. Nakatsuji, Y. Azuta, P. Ranjitkar, and Y. Tanaboriboon, "Multiple car-following data with real-time kinematic global positioning system," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1802, pp. 166–180, January 2002.
[48] P. Ranjitkar, T. Nakatsuji, G. S. Gurusinghe, and Y. Azuta, "Car-Following experiments using RTK GPS and stability characteristics of followers in platoon," in *Applications of Advanced Technologies in Transportation*, American Society of Civil Engineers, 2002, pp. 608–615.
[49] J. Suzuki, T. Nakatsuji, Y. Azuta, and R. Prakash, "Experimental analysis of reaction time of car-following Model," *Infrastructure Planning Review*, vol. 19, pp. 861–868, 2002.
[50] E. Brockfeld and P. Wagner, "Calibration and validation of microscopic traffic flow models," in *Traffic and Granular Flow '03*, P. H. L. Bovy, S. P. Hoogendoorn, M. Schreckenberg, and D. E. Wolf, Eds., Delft (Netherland), 2003, pp. 67–74.
[51] P. Ranjitkar, T. Nakatsuji, Y. Azuta, and G. Gurusinghe, "Stability analysis based on instantaneous driving behavior using car-following data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1852, pp. 140–151, January 2003.
[52] P. Wagner and I. Lubashevsky, "Empirical basis for car-following theory development," arXiv:condmat/0311192, November 2003.
[53] P. Ranjitkar, T. Nakatsuji, and A. Kawamua, "Car-Following models: An experiment based benchmarking," *Journal of the Eastern Asia Society for Transportation Studies*, vol. 6, pp. 1582–1596, 2005.
[54] H. Suzuki, P. Ranjitkar, T. Nakatsuji, and Y. Takeichi, "An extended car-following model combined with a driver model," *Journal of the Eastern Asia Society for Transportation Studies*, vol. 6, pp. 1545–1556, 2005.
[55] P. Wagner, "Empirical description of car-following," in *Traffic and Granular Flow '03*. Springer, Berlin, Heidelberg, 2005, pp. 15–27.
[56] G. L. Shevlyakov and H. Oja, "Robust correlation: Theory and applications," in *Wiley Series in Probability and Statistics*, John Wiley & Sons, Ltd, Chichester, UK, September 2016.
[57] R. Herman and R. B. Potts, "Single lane traffic theory and experiment," in *Proc. Symposium on the Theory of Traffic Flow*, New York, 1961, p. 26.
[58] A. M. Powell, R. Saab, and O. Yılmaz, "Quantization and finite frames," in *Finite Frames, Applied and Numerical Harmonic Analysis*, P. G. Casazza and G. Kutyniok, Eds., Birkhaeuser Boston, 2013, pp. 267–302.
[59] M. Pelgrom, "Sampling," in *Analog-to-Digital Conversion*, Springer International Publishing, 2017, pp. 5–56.
[60] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," *Journal de Physique I*, vol. 2, no. 12, pp. 2221–2229, 1992.
[61] E. G. Walsh, "Role of the vestibular apparatus in the perception of motion on a parallel swing," *The Journal of Physiology*, vol. 155, no. 3, pp. 506–513, March 1961.
[62] A. J. Benson, "Sensory functions and limitations of the vestibular system," in *Perception and Control of Self-Motion*, Psychology Press, 1990, pp. 145–171.
[63] C. Gianna, S. Heimbrand, and M. Gresty, "Thresholds for detection of motion direction during passive lateral whole-body acceleration in normal subjects and patients with bilateral loss of labyrinthine function," *Brain Research Bulletin*, vol. 40, no. 5, pp. 443–447, January 1996.
[64] A. J. Gundry, "Thresholds to roll motion in a flight simulator," *Journal of Aircraft*, vol. 14, no. 7, pp. 624–631, 1977.
[65] E. Boerger and R. Stark, *Abstract State Machines: A Method for High-Level System Design and Analysis*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
[66] E. R. Gansner and S. C. North, "An open graph visualization system and its applications to software engineering," *Software - Practice and Experience*, vol. 30, no. 11, pp. 1203–1233, 2000.
[67] J. Ellson, E. Gansner, L. Koutsofios, S. C. North, and G. Woodhull, "Graphviz—Open source graph drawing tools," in *Graph Drawing, Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, September 2001, pp. 483–484.
[68] T. Kamada and S. Kawai, *An Algorithm for Drawing General Undirected Graphs*, Inf. Process. Lett., April 1989, pp. 7–15.
[69] M. Green, "How long does it take to stop?" *Methodological Analysis of Driver Perception-Brake Times, Transportation Human Factors*, vol. 2, no. 3, pp. 195–216, September 2000.
[70] K. Jeannotte, A. Chandra, V. Alexiadis, and A. Skabardonis, "Traffic analysis toolbox volume II: Decision support methodology for selecting traffic analysis tools," Technical Report, Federal Highway Administration, 1200 New Jersey Avenue, SE Washington DC, USA, June 2004.
[71] E. P. Todosiev, "The action point model of the driver-vehicle system," PhD thesis, The Ohio State University, 1963.
[72] P. Wagner, R. Nippold, S. Gabloner, and M. Margreiter, "Analyzing human driving data an approach motivated by data science methods," *Chaos, Solitons & Fractals*, vol. 90, pp. 37–45, September 2016.

[73] T. Akita, S. Inagaki, T. Suzuki, S. Hayakawa, and N. Tsuchida, "Analysis of vehicle following behavior of human driver based on hybrid dynamical system model," in *Proc. IEEE International Conference on Control Applications*, October 2007, pp. 1233–1238.
[74] R. Falconi, C. Longhin, A. Paoli, and C. Bonivento, "Hybrid driver-vehicle dynamics in microscopic modeling of traffic networks*," IFAC Proceedings Volumes, vol. 45, no. 9, pp. 356–363, January 2012.
[75] A. Schwarze, M. Buntins, J. S. Uffmann, U. Goltz, and F. Eggert, "Modelling driving behaviour using hybrid automata," *IET Intelligent Transport Systems*, vol. 7, no. 2, pp. 251–256, June 2013.
[76] J. W. Ro, P. S. Roop, A. Malik, and P. Ranjitkar, "A formal approach for modeling and simulation of human car-following behaviour," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–10, 2017.
[77] M. Colledanchise and P. Oegren, Behavior Trees in Robotics and AI: An Introduction, arXiv:1709.00084 [cs], August 2017.
[78] M. Colledanchise and P. Ögren, "How behavior trees modularize hybrid control systems and generalize sequential behavior compositions, the subsumption architecture, and decision trees," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 372–389, April 2017.

Frank Lehmann received the Masters degree in Civil Engineering with Management from the Brandenburg Technical University, Cottbus, Germany, in 2003. He then worked in automotive electronics R&D and currently pursues a Ph.D. under supervision of Prakash Ranjitkar and Partha Roop at the University of Auckland, New Zealand. His main research interests are large-scale microsimulations, fundamental modelling principles, Intelligent Transport Systems and Automata Theory.

Partha S. Roop received the Ph.D. degree in computer science from the University of New South Wales, Sydney, NSW, Australia, in 2001. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, The University of Auckland, New Zealand. His research interests include the design and verification of embedded systems and executable biology. In particular, he is developing techniques for the design of embedded applications in automotive, robotics, intelligent transportation, and medical devices that meet functional safety standards. He has received the Mercator Professorship from the German Research Council in 2016 and the Humboldt Fellowship for experienced researchers in 2009. He is an Associate Editor of the IEEE EMBEDDED SYS- TEMS LETTERS and the EURASIP Journal on Embedded Systems.

Prakash Ranjitkar received the B.E. degree in civil engineering from Tribhuvan University, Kathmandu, Nepal, in 1994, the M.E. degree in civil (transportation) engineering from the Asian Institute of Technology, Bangkok, Thailand, in 1999, and the Ph.D. degree in civil (transportation) engineering from Hokkaido University, Sapporo, Japan, in 2004. Since 2007, he has been a Senior Lecturer with the Department of Civil and Environmental Engineering, The University of Auckland, New Zealand. He was a Research Fellow with the Department of Civil and Environmental Engineering, University of Delaware, Newark, DE, USA, from 2006 to 2007, a JSPS Post-Doctoral Research Fellow with the Graduate School of Engineering, Hokkaido University, from 2004 to 2006. His research interests include intelligent transportation systems, traffic flow theory, optimisation of traffic control systems, human factors, traffic safety, modelling and simulation of transport facilities, public transport, and applications of emerging technologies in transportation.