SimMobility Short-term: An Integrated Microscopic Mobility Simulator

Carlos **Lima Azevedo**
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA
Phone: +1 6174522482
Fax: +1 6172531130
email: cami@mit.edu

*Simon **Oh**
Singapore MIT-Alliance Research and Technology
1 Create way, 138602 Singapore
Phone: +65 66012828
Fax: +65 6684 2118
email: simon@smart.mit.edu

Neeraj Milind **Deshmukh**
Singapore MIT-Alliance Research and Technology
1 Create way, 138602 Singapore
Phone: +65 66011634
Fax: +65 6684 2118
email: neeraj@smart.mit.edu

Balakumar **Marimuthu**
Singapore MIT-Alliance Research and Technology
1 Create way, 138602 Singapore
Phone: +65 66011634
Fax: +65 6684 2118
email: balakumar@smart.mit.edu

Katarzyna **Marczuk**
Singapore MIT-Alliance Research and Technology
1 Create way, 138602 Singapore
Phone: +65 66011634
Fax: +65 6684 2118
email: katarzyna@u.nus.edu

Harold **Soh**
University of Toronto
27 King's College Cir, Toronto, ON M5S
Phone: -
Fax: -
email: harold.soh@utoronto.ca

Kakali **Basak**
Singapore MIT-Alliance Research and Technology
1 Create way, 138602 Singapore
Phone: +65 66011634
Fax: +65 6684 2118
email: kakali@smart.mit.edu

Tomer **Toledo**
Technion - Israel Institute of Technology
Haifa, Israel
Phone: +972 48293080
Fax: +972 48295708
email: toledo@technion.ac.il

Li-Shiuan **Peh**
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA
Phone: +1 6173248428
Fax: -
email: peh@csail.mit.edu

Moshe E. **Ben-Akiva**
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA
Phone: +1 6172535324
Fax: +1 6172531130
email: mba@mit.edu

* Corresponding Author: simon@smart.mit.edu

Word count: 6401 (Manuscript) + 5 Figures + 1 Tables = 7901words
Submitted the 1st August 2016

ABSTRACT

In this paper we present the development of an integrated microscopic mobility simulator, SimMobility Short-Term (ST). "Integrated" as its models, inputs and outputs, simulated components and code-base are integrated within a multi-scale agent- and activity-based simulation platform capable of simulating different spatial-temporal resolutions and account for different levels of travelers' decision making. "Microscopic" as both the demand – agents and its trips – and the supply –trip realization and movements on the network – are microscopic, i.e. modeled individually. Finally, "Mobility", as it copes with the multimodal nature of urban networks and the need for the flexible simulation of innovative transportation services such as on-demand and smart mobility solutions. This paper follows previous publications that describe SimMobility's overall framework and models. SimMobility is a multi-scale platform that considers land-use, transportation, and mobility-sensitive behavioral models. SimMobility ST aims at simulating the high-resolution movement of agents (traffic, transit, pedestrians and goods) and the operation of different mobility services and control and information systems.

This paper presents SimMobility ST modelling framework and system architecture, reporting its successful calibration for Singapore and its use in several scenarios of innovative mobility applications. We also show how detailed performance measures from SimMobility ST can be integrated with a daily activity and mobility patterns simulator. Such integration is crucial to accurately model the impact of different technologies and service operations at the urban level, as the identity and preferences of simulated agents are maintained across temporal decision-scales ensuring the consistency and accuracy of simulated accessibility and performance measures of each scenario.

KEYWORDS

Keywords: microscopic simulation, traffic simulation, activity-based models, mobility, communication networks

1 INTRODUCTION

Microscopic traffic simulation applications are now part of the daily transportation planning and operation realities. In order to explore and evaluate complicated future transportation scenarios, we need to conduct experiments. Physical experiments can be conducted in order to understand how the different options work together on a small scale. For the entire urban area, however, a simulation model is the only viable option. Microscopic traffic simulation models have been widely used to test different road network and ITS solutions. They aim at replicating detailed vehicle motions and interactions by modelling agent decisions such as route choice, accelerations, decelerations, and lane changes. These models are implemented as synchronous applications that update the kinematic parameters of each entity (driver-vehicle units, public transportation, management systems and even pedestrians) at every simulation time step. Similar to other transportation simulators, the design of microscopic models is based on a demand and supply equilibrium representation. Traffic demand input is formulated either by defining it in terms of input flows and turning proportions at intersections or, for larger networks, in terms of origin–destination (OD) matrices that will rely on route-choice models for network assignment [*1*]. An example of these simulators are AIMSUN [*4*], VISSIM [*5*], Q-Paramics [*6*], Tansmodeler [*7*], ARTEMIS [*8*], CORSIM [*9*], DRACULA [*10*], HUTSIM [*11*], Integration [*12*], MITSIMLab [*13*], SUMO [*14*], Cube DynaSIM *[15]*. The first four simulation tools belong to the short group of integrated platforms available for fast implementation and that have been successfully used in a variety of transportation projects, accounting for a share of more than 70% of the practitioners and researchers preference [16].

In microscopic traffic simulation, the supply implementation relies on the specification of the network configuration, the traffic management algorithms and the driving behavior model. From the initial models developed in the 50's for car-following behavior [*2*], traffic microscopic simulation models now include multiple detailed behaviors and have reached a high level of maturity not only among the research community but also at the practitioners regular [*1*]. For a comprehensive review on all driving behavior components used in simulation, the reader should refer to [*1*], [*3*].

Commercial simulation tools have devoted a large share of their new features to enhanced interfaces, visualizations and, sometimes, calibration frameworks. At the same time, they have managed to integrate dedicated traffic control modules, public transportation and pedestrian simulation [*4*], [*5*], into their core architecture. On the other hand, the simulation research stream has proposed several innovative driving behavior models [*11*],[*24*] integration of communication technologies or even emissions models [*20*]. Some of these features can also be found in case studies using commercial software, but typically by means of coupling external modules to the main simulation tool, eventually compromising computational performance and interactive behaviors. Furthermore, several recent efforts in the research community can be found in the development of sophisticated activity-based modelling frameworks and their integration in simulation platforms that focus on individual's entire day activity pattern. Comprehensive agent-based modelling structures developed so far can be listed as TRANSIMS [*17*], MATSim [*18*] and FEATHERS [*19*]. These applications model the multiple choices and behaviors of a single agent during a day and have shown to better represent the interactions and dependencies of individual mobility. However, computational efficiency of such platforms are always been a major concern as they usually deal with the entire population of an area which is synthetically generated.

The above development streams have raised several challenges in the integration of complex mobility and transportation models within microscopic simulation. The next generation of simulators should include activity-based frameworks, integrated formulations with higher level, consideration of alternative modes such as on-demand mobility and autonomous vehicles, advanced and flexible driving behavior models, the possibility to easily integrate innovative transportation services such as vehicle-to-vehicle communication or logistic services.

This paper presents SimMobility Short-term (SimMobilityST), a new open-source microscopic mobility simulator integrated in a multilevel simulation platform. Designed using an agent-based framework, SimMobilityST aims at simulating the movement of agents (traffic, transit, pedestrians and goods) and the decisions and operation of control centers, within one day. It considers individual travel behavior in detail using an activity-based formulation. In the next two sections both the modelling framework and the system architecture are presented. Its modular structure is described in more detail with a focus its multiple components along with the recent applications that showcase its flexibility in the simulation of different and innovative mobility scenarios. Finally, the successful calibration process of the demand-supply parameters of SimMobilityST for the city of Singapore is described.

2 SIMMOBILITY

2.1. Overall Framework

SimMobility is designed as three primary modules segmented according to timeframes [27]. The short-term model functions at the operational level; it simulates movement of agents at a microscopic granularity (within day) and is presented in more detail in the next section. The mid-term (day-to-day) simulator handles transportation demand for passengers and goods; it simulates agents' behavior including their activity and travel patterns; and it shares several mobility decisions with the short-term level (e.g., with the route-choice model) [38]. The long-term (year-to-year) model captures land-use and economic activity, with special emphasis on accessibility. It predicts the evolution of land use and property development and use, determines the associated life cycle decisions of agents, and accounts for interactions among individuals and firms. The high-level design of SimMobility is shown in **FIGURE 1**.

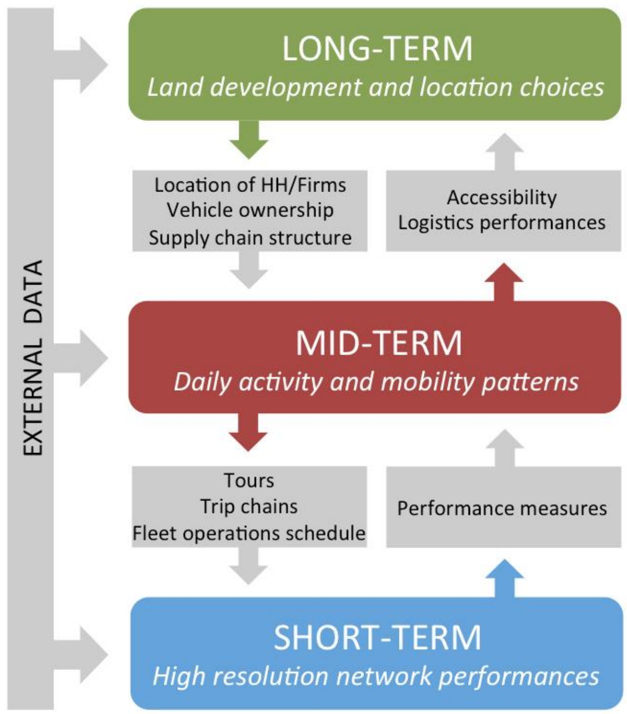
SimMobility must therefore include all the key mobility-related decisions that people make in their everyday life. These decisions may be personal decisions of households or the commercial decisions of firms [*26*]. To support this level of representation, SimMobility is based on the concept of agent-based or micro-simulation. Representation of individuals as agents in the model is necessary to simulate how people will react in the future to new infrastructures, new technologies, innovations in system management and policy changes.

The SimMobility framework is fully modular in the sense that each of the levels can run independently and only access the other levels when necessary. The key to multi-scale integration in SimMobility is a single database model that is shared across all levels. Every agent exists and is recognized at all levels simultaneously, and information is used according to each level's needs; in this way, behaviors will remain consistent and, even if run separately, the impacts from one level's model will be propagated to the others gracefully.

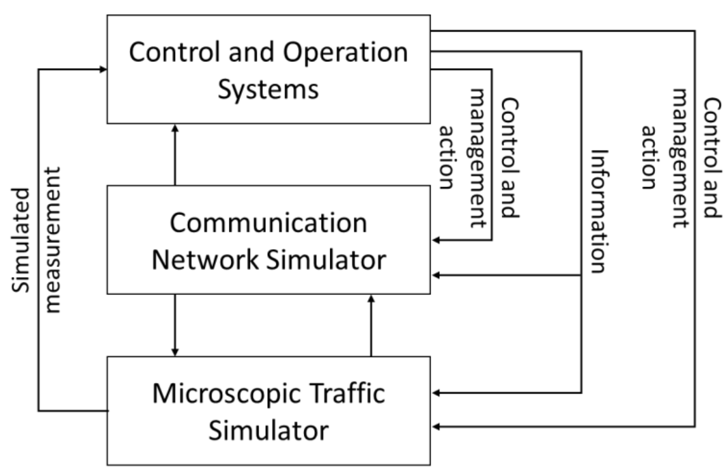
In previous work, these models have not been fully integrated. While there is limited interaction of outputs, there is no internal coherence. SimMobility is unique in that the same pool of agents is used across all timeframes. For further details on SimMobility's overall framework and Mid-Term and Long-Term Simulators, the reader is referred to [27],[38] and, [50] respectively.

2.2. SimMobility Short-term Framework

SimMobilityST is an agent-based, multimodal microscopic simulator where agents' movements are captured at a very fine resolution (up to 100 milliseconds). SimMobilityST comprises three main components. The Microscopic Movement module is responsible for advancing drivers, pedestrians and goods on the transportation network according to their respective behavioral and decision models. The Control and Management module simulates the control centers, such as traffic and parking control, bus control, rail control, autonomous fleet control, logistic control, etc. The outcomes of these control actions will influence an agent's movement decisions, path choices and other related decisions in the movement simulator. Within the Control and Management Module, different control centers may be considered and replicated. At the current state of the simulator, the service controller and the traffic management controller are operational as described below. On-going efforts are being made in the development of the freight controller using detailed freight and logistics data [47]. The third component is the Communication network simulator, which simulates agent-to-agent communications. The information can be passed from one agent to another via the mobile communication or via vehicle-to-vehicle communication or maybe via vehicle-to-infrastructure communication. The Communication network simulator is responsible for simulating the physical communication network (for example, a wireless network), and agents simulated within microscopic traffic network will use this simulated network to pass information between them. This will help the agents to get the realistic communication network, which will handle the message delivery delay or coverage.



(a) High-level SimMobility Framework



(b) Short-term Framework

FIGURE 1 Framework of SimMobility and SimMobility ST

3 MODULES

3.1 The Microscopic Traffic Simulator

The structure of the Microscopic Movement module is detailed in **FIGURE 2**. The virtual world is populated during the initialization phase, after which the simulation receives the control information/action plan at every time step. Two kinds of behaviors are simulated: high level (travel) decisions, such as route choices, are taken at some decision point (e.g., a bus stop). Lower level or movement decisions, such as car following and lane changing, occur while the agent is in movement. While the agent's position is updated at every time step, the movement-related decisions only takes place when specific events occur.

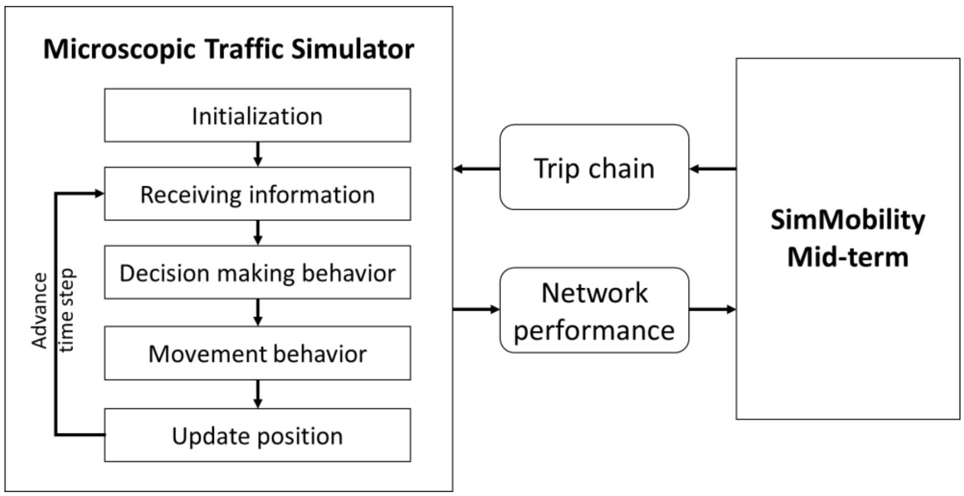


FIGURE 2 SimMobility Short Term Simulator

Demand input

Instead of the traditional Origin-Destination matrix definition used in the demand formulation of traffic microscopic simulation models, SimMobilityST (and its higher level counterparts) uses an activity-based demand formulation in the form of activity-schedules. In such approach, trip chains are generated by individual daily schedules instead of aggregated traffic specific matrices. Such data can be obtained directly using mobility and goods survey data or using a pre-day model, such as the one integrated in the SimMobility mid-term framework [38]. The pre-day model consists of an activity-based modelling system formulated as interconnected discrete choice models representing choices at distinct dimensions. This pre-day model development follows the Day Activity Schedule approach [*25*], [*26*] which focuses decisions related to daily activity and mobility. There are three different hierarchies in the system: day pattern level, tour level and intermediate stop level. Each level consists of several models, such as mode choice or departure tie choice. For the full specification of the pre-day in SimMobility Mid-term model, the reader is referred to [*27*]. The output is an activity schedule and the trip chains for each agent in the simulation.

Within SimMobilityST, agents are then moved as per the planned trip chain. However, the realized trips can be changed during the simulation if specific circumstances such as high congestion, incidents, public transportation interruptions or any control or information provision are observed.

For each agent's sub-trip (a multimodal trip can have several sub-trips) generated in the simulation its role is assigned (pedestrian, passenger and driver) and its role-specific characteristics are generated (e.g., aggressiveness, look-ahead distance, reaction times, etc). For each vehicle-based trip, an individual vehicle is generated. On-going work is being carried out to allow vehicle ownership and parking models to integrate a consistent vehicle generation model with unique identifiers. The generated vehicles are then assigned vehicle attributes (e.g., type and drive train) based on configurable distributions.

Static Supply input

The network in SimMobilityST is composed by a road network layer, a pedestrian network layer and a public transportation layer. The road network layer is composed by (1) Nodes, (2) Links, with (3) Segments, (4) Polylines, and (5) Lanes. Connectivity attributes is assured by (6) Lane Connectors, (7) Turning Groups, (8) Turning Paths and (9) Conflict Points.

Nodes represent intersections or source/sink for trip chains. They are used for Link definition in route-choice and for the detailed characterization of intersections. Links are directional roads that connect Nodes and are composed by Segments. The latter are road sections with uniform geometric characteristics (speed limit, design speed, grade). Each segment as a fixed number of Lanes, each with its specific lane rules (lane changing regulation and use privilege). Polylines determine the shape of the Segment and lane connectors define the connectivity between Segments. At the Node level, Turning Groups, Turning Paths and Conflict Points can be defined. Turning groups define the connectivity between Links, while the Turning Paths Additionally, (10) Road Items are point specific features that can be added to the network to represent items on the road to which drivers must respond (e.g.: traffic lights, bus stops, etc). On-going work is being carried out to extend this framework with parking infrastructure.

Driving behavior

The core traffic model of SimMobilityST is based on MITSIM, an open-source microscopic traffic simulation application developed by the Massachusetts Institute of Technology. MITSIM moves vehicles according to route choice, acceleration and lane changing models. The acceleration model captures drivers' response to neighboring conditions as a function of surrounding vehicles motion parameters. The lane-changing model integrates mandatory and discretionary lane-changes in single model. Merging, drivers' responses to traffic signals, speed limits, incidents, and tollbooths are also captured. The driving behavior models implemented in MITSIM are those estimated by [*22*] and [*21*]. The MITSIM lane-changing model was later enhanced by *[24]*, for the specific purpose of integrating latent plans in the lane selection process, namely in urban arterials and in freeways with a large number of lanes.

Several additional enhancements were made to the MITSIM original driving behavior: an enhanced reaction time formulation capable of explicitly model reaction time and perception delays for each person in a detailed and flexible manner as introduced (see *[28]* for further details); lateral movement during lane-change was also included. For the current implementation, the lateral speed is kept constant during the lane-change, but the implementation of a sine function for lateral acceleration similar to the one proposed in *[30]* has been initiated.

Finally, the design of a dedicated intersection behavior model, based on the conflicts technique has also been implemented. The intersection behavior starts once the intersection is visible to the vehicle. First, the driver identifies the intersection regime (no rules, priority, or controlled). If the intersection is not controlled, the subject vehicle identifies the neighboring vehicles and the conflicting vehicles, and proceeds with a gap-acceptance based model that accounts for intersection-specific priorities (if any).

Travel behavior

Within SimMobilityST, changes in planned trip chains have to be considered. As the simulation is running, the agents need to find the routes for their trips and transform the activity schedule into effective decisions and execution plans. Agents may get involved in a multitude of decisions, not constrained to the planned set of destination, mode, path and departure time depending upon the network and their state in the simulation cycle [38]. In the current implementation agents can re-route (as drivers or public transportation passenger) in the presence of congestion or the provision of control and information. Route choices are based on a probabilistic model that captures the impact of travel times and biases toward routes that use freeways over urban streets. The impact of real-time information on routing decisions is captured by a route-switching model in which informed drivers re-evaluate their pre-trip route choices based on the traffic conditions observed en-route. For perfect model integration with higher level simulators, the route-choice model used in SimMoblity ST is the same as the one used in the SimMobility Mid-Term framework and can its details be found [38], [27].

Pedestrian movements

The pedestrian behavior model focuses on the problem of how a pedestrian makes crossing-related decisions at different levels and at different points of times, when she/he walks along a given path to destination. Specifically, a crossing choice module is designed to determine where a pedestrian crosses the road, along a given path. A crossing timing decision module is designed to control when a pedestrian starts crossing the road, once she/he reaches a crossing point. At the time of writing this paper, pedestrian's microscopic movement has been implemented in a simplified way with constant speed and unified crossing time synchronized with pedestrian traffic light change.

Commodity movements

The movement of freight vehicles is typically considered in microscopic traffic simulation models by means of adapting driving behavior parameters for heavy vehicles [*31*] or by coupling dedicated external applications with the simulator [*32*]. Teo et al. [*33*] extended these traditional approaches by integrating the simulation of freight movements, logistics decisions and traffic within an agent-based simulation. SimMobilityST is the first microscopic simulator that integrates commodity specific movements with detailed traffic models. Similarly to individual trip-chain input, SimMobilityST allows commodity specific shipments. The commodity entity was specified for this purpose and freight drivers are assigned tours based on the commodities to deliver during the simulation period. A default tour generation model was developed but this will be relaxed and linked to a freight operator controller, which typically represents a carrier. Freight vehicles and drivers will then be assigned to a specific freight operator and a set of delivery stops specified as Road Items in the network. Decisions on the freight vehicle tours can be made by the freight operator controller or the driver himself. The design and integration of all these entities within the core models of SimMobilityST is still under development at the stage of writing this document.

3.2 Control and Operation Systems

Traffic Management Controller

The Traffic Management Controller mimics the traffic and information control system in the network under consideration. A wide range of traffic control and route guidance systems can be simulated. These include intersection controls, ramp control, freeway mainline control, lane control signs, variable speed limit signs, portal signals, variable message signs and in-vehicle route guidance. The Traffic Management Controller can represent different designs of such systems with logic at varying levels of sophistication (pre-timed, actuated or adaptive) by means of a flexible configuration input.

Control devices can be either link-wide (such as Variable Speed Limits) or lane specific (e.g.; lane use regulation). They are represented by Road Items and are characterized by their location, type and visibility distance. Their logic is implemented directly by the Traffic Management Controller and the analyst will need to code its logic through external scripting files (in Lua language). An example of tested implementation in the current state of SimMobilityST development is the SCATS-like algorithm for traffic signals [*29*] and an innovative time slot-based algorithm for the coordinated management of intersections for autonomous vehicles [45].

Service Controllers

Service controllers are the central control point responsible for operation a specific mobility service. It holds relies on static information (e.g.: vehicle fleet composition, routes, etc) as well as real time information (e.g. traffic sensor measurement of vehicle location data) and communicates with the simulated vehicle operators/drivers to send them instruction at various situations. In the current development of SimMobilityST, the bus transportation framework and controller are already implemented. We currently have two setting of the controller: the Bus Controller and the Smart Mobility Controller.

The Bus Controller is responsible for the scheduling and dispatching of buses, for keeping track of individual arrival times, and for deciding on transit control strategy. Along with the Bus controller, other features were implemented in SimMobilityST: the Bus Driver Agent will be responsible for routing the bus on fixed route, the bus movement near the bus stop, real time passenger count and dwell time calculation. The bus movement for example, includes mandatory lane-changing maneuvers in order to reach the lane where the stop is, depending on a distance-to-stop threshold. When a bus is far than bus-to-stop visibility the driver agent may make discretionary lane changes according to the same logic that applies to other drivers with a preference toward the lane that contains the bus stops; finally, the Passenger/Pedestrian Agents will also interact in the bus framework as they are responsible for realizing their individual bus route choice, boarding choice and alighting choice.

To provide reliable service the bus controller may implement different strategies, and holding strategy is one of them. Different holding strategies have already been implemented and tested in SimMobilityST: schedule-based, headway-based and even headway-based strategies.

SimmobilityST also provides mechanisms for simulating emerging technologies that are yet to be widely available, for example, mobility-on-demand services. Such feature was achieved with the development of uniform interfaces between a Smart Mobility Controllers and the different models in SimMobilityST. The Smart Mobility Controller relies on third-party code that can run separately from SimMobilityST and mimics the operation of fleets, from regular taxi to Uber-like systems. This code interacts with agents and entities in SimMobility in runtime, and a few features have been made accessible to it.

Two application of the proposed Smart Mobility Controller can be found already in the literature: an Autonomous Mobility on Demand (AMOD) service, which provides one-way car-sharing with self-driving electric vehicles and has emerged as a promising solution for autonomous urban transportation [48]; and Flexible Mobility on Demand (FMOD), which provides personalized and optimized services to travelers in real-time with flexibilities both on the operator and traveler side [49]. These technologies are designed to deal with the recent trends that emphasize more flexibility through the use of shared-ride services and integration of multimodal mobility options.

3.3 The Communication Network Simulator

Many simulators broaden their applicability by allowing customized interactions with third-party components, and SimMobility is no exception. In addition to traditional library-based extension, SimMobilityST provides a TCP socket integration layer that allows other software systems to interact with a running SimMobilityST simulation. This is primarily used in [*34*] to overlay Android emulators running transit-related applications or “apps” onto existing SimMobility agents, thus providing accurate location information to the apps. In return, the apps provide more realistic within-day re-routing, creating a feedback loop which should optimize the system.

The choice to communicate over TCP sockets has several advantages. First, it requires minimal changes to existing third-party software systems –usually only a small communication module is needed. Second, it facilitates interactions between a larger number of simulators. In the example just discussed, SimMobilityST connects to a running instance of a network simulator (ns-3), which it uses to provide accurate timing and packet loss information for messages sent between Android clients. Finally, the use of TCP sockets provides a stable, cross-platform means of interaction with clearly defined boundaries.

4 SYSTEM ARCHITECTURE

4.1 Overall framework

SimMobilityST applies several design heuristics to make modelling and development easier for a heterogeneous user base. First, entities are isolated from each other, and can only interact through properties that are shared among them. This isolation is achieved through the use of agent-based simulation techniques. Second, the simulator is location-agnostic regarding agents. In other words, an agent's interface does not change depending on where it is in relation to other networks (except when MPI is enabled). Third, SimMobilityST's time-step is indivisible –agents are assumed to all tick forward at once. Finally, SimMobility is hierarchical and provides sensible defaults. A good example of this behavior is the use of trip-chains, which can be "filled in" with more information as the agent's trip progresses. If an agent does not have a route for a given segment of the trip chain, one can be estimated for it.

SimMobilityST is designed as a hybrid software framework, including both event-driven messaging and discrete time-step simulation. Heterogeneous time-steps are supported, allowing coarse-grained software agents such as traffic signals to interact efficiently with fine-grained agents like drivers. Ultimately, SimMobilityST was designed with accuracy and performance as its two primary goals. To this end, it includes a parallel and a distributed component, which will now be described in turn.

4.2 Parallel computing

SimMobilityST features a robust, straightforward approach to massive parallel scalability that was designed to take advantage of the processing power of modern hardware. The majority of computations performed by SimMobilityST are done by entities as they update their internal state. This process is performed once per entity per time tick. Entities with similar time step resolutions are grouped together into Workers that manage the update process for a given thread.

Entities can generally ignore the Worker that they assigned to, as all communication through other agents is done using buffer-backed variables. These variables use an internal buffer to allow lock-less communication with any entity on any Worker. In addition, Entities will be automatically added to a Worker when their start time of their first trip arrives, and will be removed once the final trip's destination has been reached. Typically, entities only interact with their Workers when requesting a manual migration. This can occur when the agent crosses an MPI boundary (described in Section 4.3), or if some kind of spatial optimization such as the mid-term's "conflux" structure is desired.

This use of buffers runs counter to traditional logic of using mutex-based locking for parallel communication. Ultimately, the conventional approach exhibits several systemic and non-trivial problems. Primarily, it limits the repeatability of the simulation by introducing an unacceptable amount of non-determinism. Entities are constantly reading and reacting to the internal state of each other, and performing these reads in parallel leads to different orderings for each simulation run. Attempting to solve this by ordering entity updates will remove some of the benefits of parallelism, which is particularly detrimental to entities that are not sensitive to update order. A secondary issue with lock-based synchronization is the heavy toll it places on many-cored systems, especially in traffic simulation with its high degree of inter-agent data dependency. As the number of discrete processing units on a system increases, more and more performance is lost to overhead.

An additional channel for communication is event-driven messaging. Here, entities can register for messages to be delivered for a series of given events, such as node arrivals or agent deactivations. These messages will be triggered during a time tick, gathered and sent at the end of that time tick, and received at the beginning of the next time tick. Thus, messages inherently incur a one-tick delay, although it is possible to reduce this to zero if certain conditions hold (e.g., passengers on a bus can receive zero-delay messages). Although event processing is inherently single-threaded, it can lead to large performance gains by allowing entities to deactivate their update phase until a given message arrives. Furthermore, entities can register for events and still maintain their update phase, thus allowing for a hybrid of event-driven and discrete time-stepped simulation.

4.3 Distributed computing

Once simulations encompass a large enough number of entities, it is inevitable that parallel simulation will reach a point of diminishing returns. At this point, the simulation must be split and run on different machines ("nodes") through SimMobilityST's MPI-based distributed computing platform. Although the same parallel setup just discussed is still run on each node, the inter-node communication is inherently less flexible and requires modelers to abandon the location-agnostic property of the single-node setup. In particular, entities must be located in a particular geometric "space", and various spatial decomposition techniques are used to assign different spaces to different physical machines. The implications of this will now be discussed.

The global state of the simulation, including the road network and the trip chains, is split and distributed to each node, whereupon it is loaded on an as-needed basis (to reduce memory usage). Agents are distributed to the node which contains their starting trip, and will be automatically transferred to other nodes as they cross the relevant node's boundary. A node boundary is defined as a line perpendicular to a given road segment which divides the upstream and downstream halves of that segment between two nodes.

Entities on different nodes can sense each other via mirroring, but they cannot otherwise interact. Furthermore, entities that are not in a mirrored region cannot send each other messages unless they are on the same node. This is an unfortunate necessity, as it puts a lower bound on the number of nodes that a given node must interact with, in turn allowing SimMobilityST to scale efficiently up to arbitrarily complex road networks and simulation workloads. In order to work around these limitations, a delayed-hopping message protocol is in development which allows messages to be sent to agents on other nodes at a time cost of N time ticks, where N is the number of nodes between the originating agent and the receiving agent. This system is flexible, and permits the time delay to be further reduced (to at least 1) through the use of software-defined relays.

4.4 Visualizer

To represent the simulator output graphically an interface is developed in C++ using QT libraries. This GUI is used for debugging purpose and to demonstrate of traffic impact through vehicle animation (**FIGURE 3**). This application accepts the simulator output file produced in plain text format and then it display's vehicle trajectory at each frame tick and also saves the simulator output in the database for efficient subsequent runs. It supports zoom-in and zoom-out operation handle large or small area visualization, road network entity search functionality to locate object easily within the GUI. It also provides user to edit the road network and convert them in SimMobility XML format. The Visuliazer is capable to read large simulator output file and more optimization techniques are in progress to make it more efficient.

4.5 Data management

SimMobilityST supports multiple data interfaces to exchange data with the other level or the data those are exogenous to the simulator. It is able to read or write data from/to XML files, PostgreSQL database and the CSV files. This will give user greater flexibility to run SimMobilityST with variety of data sources depending on the experiment requirement. SimMobilityST data requirement can be grouped into: Configuration, In/Output data, Model parameters.

SimMobilityST stores the configuration data and the model parameters in XML format. It provides greater readability for the user to configure SimMobilityST (see Section 6). Simulation input data can be either exogenous to the system for example road network, traffic light phases etc. or received from other level of SimMobilityST for example trip chains received from mid-term to short term. SimMobilityST supports both XML and Database interfaces for input data and depending on the simulation need user can specify the format in the configuration file. Usually if the simulation is required to run for a large area with massive number of agents database is more appropriate option. Simulation outputs are generated in plain text format that can be used for further processing and also the required portion is written in the database for passing to the next level.

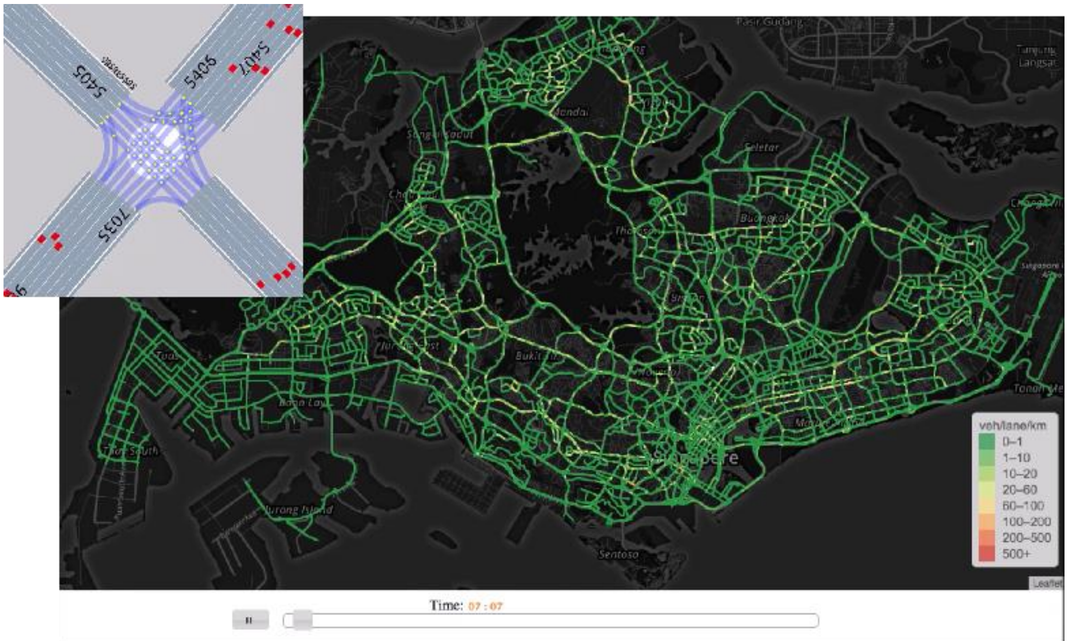


FIGURE 3 SimMobility Visualizer (Singapore simulation)

5 CALIBRATION

5.1 Demand and supply parameters

Demand parameters are typically calibrated through tuning of the OD flows. As SimMobility use activity schedules and trip chains instead of OD matrix, we aggregate the trip chain to generate OD parameter which to be calibrated, then, convert the updated OD parameter ($\hat{\theta}_{k+1}$) into trip chains by disaggregating through so-called 'killing-cloning process' for each iteration (k). Thus, each activity schedule will be "parameter" to be calibrated (killed or cloned). This means that each individual activity schedule, and therefore each agent, will be calibrated. Each activity schedule is considered as fixed, but replicates can be generated in the calibration process. The integrated calibration of Mid- and Short-Term simulators would relax this assumption by calibrating directly activity based model parameters as the demand.

Other parameters in demand side are route choice parameters. Route choice model for private transport corresponds to a Path-Size Logit Model *[39]*. Accordingly, drivers' route choice decisions are captured in a probabilistic manner, and highly likely to choose the route that maximizes his/her utility which is characterized by influencing factors including travel-time and distance. On the supply side, all driving behavior parameters are considered *[21]*, *[40]*.

5.2 Calibration framework

The simultaneous calibration of demand and supply parameters generate a large set. To deal with such complexity, we used the Weighted Simultaneous Perturbation Stochastic Approximation, W-SPSA (*[36]*, *[37]*). The algorithm finds the best parameter set by iteratively updating the parameter set to the decreasing direction in goodness-of-fit (GoF), which may reply on existing measurements and on prior knowledge on demand-supply parameters from previous experiments. Through the 'killing-cloning process', we run the SimMobilityST to get initial simulation output and assignment matrix. The assignment matrix as weight matrix for the measurements in W-SPSA. After initial setting, the calibration loop runs until reach to the convergence condition and objective value is within an acceptable level of performance. The optimization problem over parameter space during the period of $\mathrm{H} = \{1, 2, \dots, H\}$ can be formulated as:

$$\min_{x,\beta} z(\theta) = \sum_{h=1}^{H}\left[w_C z_C\left(C_h^O, C_h^S\right) + w_p z_x(x_h, x_h^a)\right] + w_p z_\beta(\beta, \beta^a) + w_p z_\gamma(\gamma, \gamma^a) \quad (1)$$

$$\text{s.t.}\ ,\ C_h^S = \text{f}_1(x_1, \dots, x_h; \beta; \gamma),\ \text{lb}_{x_h} \le x_h \le \text{ub}_{x_h},\ \text{lb}_\beta \le \beta \le \text{ub}_\beta,\ \text{lb}_\gamma \le \gamma \le \text{ub}_\gamma$$

where, z_C measures GoF between externally observed count (C_h^O) and simulated count (C_h^S); z_x compares estimated time-dependent OD parameter (x_h) with the seed OD (x_h^a) from MT; z_β and z_γ respectively evaluates the estimated parameter set in driving behavior (β) and route choice (γ) against a priori values (β^a, γ^a); and θ is decision vector ($\theta = [x_1, \dots, x_h, \beta, \gamma]$). The parameters are bounded upper/lower limits. Note that each evaluation term includes weighting coefficient (w_C;w_T;w_p), which determined by the reliability on the external information.

W-SPSA selectively perturbs relevant parameters based on a weight matrix (w) which represents spatiotemporal correlations between each parameter and measurements. Readers can refer the full structure of W matrix from *[43]*. To increase applicability in ST calibration, which deals with large number of agents for large spatial ranges, a sparse matrix has been generated in this phase. Then, the gradient approximation can be formulated as:

$$\hat{g}_{ki}\left(\hat{\theta}_k\right) = \frac{\sum_{j=1}^{D} w_{ji}\left[\left(\epsilon_{kj}^+\right)^2 - \left(\epsilon_{kj}^-\right)^2\right]}{2c_k\Delta_{ki}} \quad (2)$$

where, c_k, Δ_{ki} indicate the perturbation amplitude and random perturbation vector (following Bernoulli process) respectively. Also, note that $\left(\epsilon_j^+\right)^2$ and $\left(\epsilon_j^-\right)^2$ represent deviation vectors measuring distance between the observed and the simulated measurement with plus ($\hat{\theta}_k + c_k\Delta_k$) and minus perturbed parameter ($\hat{\theta}_k - c_k\Delta_k$) respectively. $\hat{g}_{ki}$ is the i^{th} element of the approximation of the gradient vector. This gradient provides the amount of movement from current k^{th} state ($\hat{\theta}_k$) to the next iteration ($\hat{\theta}_{k+1}$).

Using Eq. 2, we update $\hat{\theta}_{k+1}$ which yields less cost on evaluation function:

$$\hat{\theta}_{k+1} = \underset{\hat{\theta}_{k+1}}{\text{arg.}\min}\left(z\left(\hat{\theta}_{k+1} = \hat{\theta}_k - \alpha_k\hat{g}_k(\hat{\theta}_k)\right), z\left(\hat{\theta}_{k+1} = \hat{\theta}_k + \alpha_k\hat{g}_k(\hat{\theta}_k)\right)\right) \quad (3)$$

where, α_k is a step size in a gain sequence. Therefore, SimMobilityST calibration includes a backward decision process that gives additional chance to consider opposite direction in the decision vector update. Also, note that the two different function evaluation on two parts (Eq. 2) conducted on a parallelized way its independent each other and the algorithmic parameters are selected as in *[41]*. Once objective value satisfies the internal convergence term, we terminate full calibration process.

5.3 Experimental setting

SimMobilityST is calibrated by running the traffic for the extended Central Business District (CBD) in Singapore (**FIGURE 4**). This area contains more than 1200 intersections, which covered by more than 2000 loop detectors. A smaller sub-network called Bugis, located inside CBD (10 intersections), was also tested for assessing the impact of daily variability in the calibration process. The aggregated demand generated by SimMobility MT has 1497 observed ODs pairs and a total of 48,988 trips. These trips (demand), 11 route-choice parameters for demand and 112 driving behavior parameters (supply) are the set of parameters to calibrate.

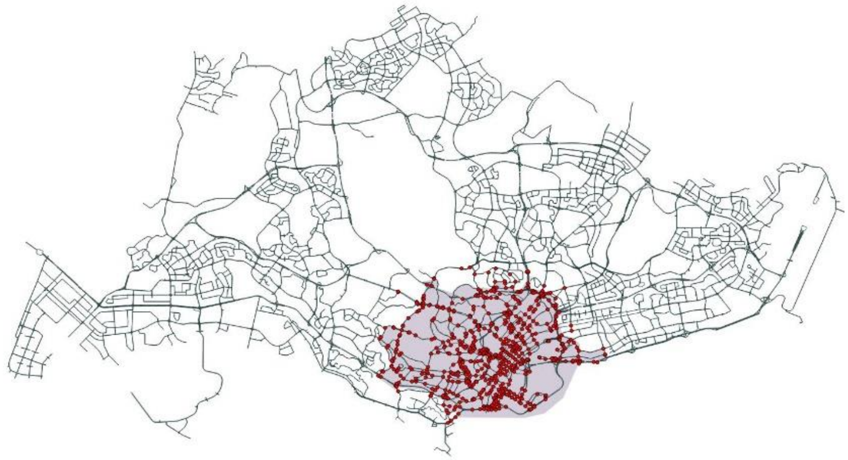


FIGURE 4 Extended Central Business District (CBD) in Singapore (shaded area; red dots are measurement locations)

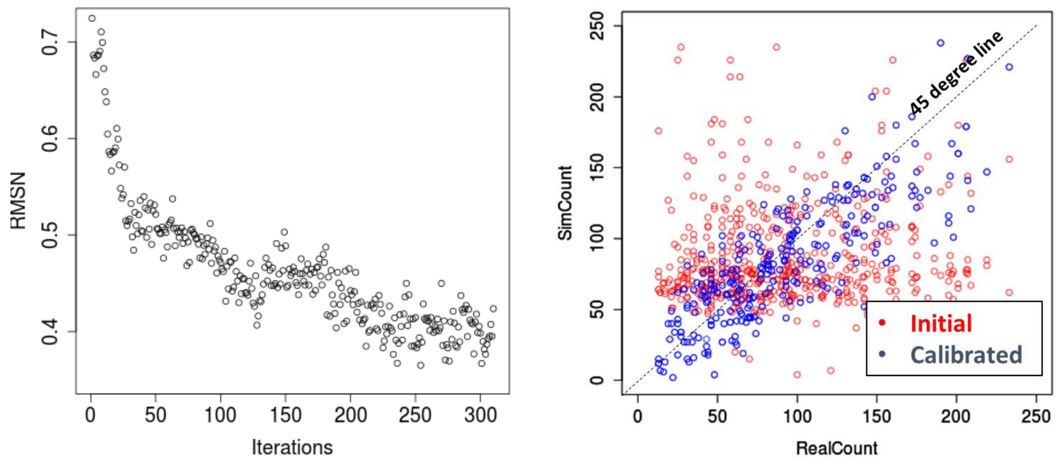
For the calibration, two types of data were available: loop count and GPS travel-time data from probe vehicles, collected in August 2013 by the Land Transport Authority of Singapore and Taxi data location respectively. Counts had a resolution of 5min while the GPS data was structured in OD a travel-time tables for each 30min interval of the day. Counts were also preprocessed for outlier detection *[44]*. A total of 360 sensors were used in the calibration. The following data was also used in network settings: SCATS signal phases, GIS network configuration, Google transit network data for the buses routes and schedules, as well as freight (background) traffic data.

5.4 Calibration result

The result shows the calculated RMSN between the simulated and observed count over all segments and time intervals. **TABLE 1** shows RMSN for multiple days in small Bugis network. This shows that the calibration framework is able to calibrate simulator using different external data sets, with improvements 40~50% in RMSN. In the extended CBD area and after 250 iterations, the fit-to-counts has been improved from 0.72 to 0.37 of RMSN (**FIGURE 5** (a)) and the calibrated counts became close to the 45-degree line (**FIGURE 5** (b)). The calibrated RMSN seems to be satisfactory given the above small number of sensors and relatively large number of parameters.

TABLE 1 RMSN for multiple days in Bugis network

Date in Aug./2013	6^{th}	7^{th}	13^{th}	14^{th}	15^{th}	20^{th}	21^{st}	22^{nd}
Initial	0.58	0.53	0.53	0.54	0.49	0.47	0.51	0.46
Calibrated	0.29	0.29	0.32	0.26	0.25	0.29	0.27	0.25



(a) RMSN over iterations (b) Fit-to-counts

FIGURE 5 Calibration result for the Singapore extended CBD area

6 CONCLUSIONS

A microscopic mobility simulator, SimMobilityST, aiming at simulating the movement of agents (traffic, transit, pedestrians and goods) and the decisions and operation of control and logistic centers is presented. SimMobilityST allows for a modular integration of specific behaviors associated with new mobility services and transportation modes. It is integrated within the SimMoblity framework, a multi-scale simulation platform that considers land-use, transportation and communication interactions using various behavioral models. The new simulator particularly focuses on impacts of innovative transportation services on transportation and mobility networks, thereby enabling the simulation of a portfolio of technology, policy and investment options under alternative future scenarios. SimMobilityST has been successfully calibrated using external data in Singapore. Multiple days supports the replicability of calibration capability as well. This calibrated simulator would contribute to increase simulation reliability in evaluation of new scenarios in Singapore and elsewhere.

The main on-going development efforts have been focusing in integrating further (existing) advanced driving behavior models; designing and implementing the urban freight tour-based logic, along with its specific behaviors and logistic decisions; and implement further smart mobility services.

Finally, the integration of a dedicated framework for simulating electrical vehicles and both environmental and safety impacts assessment modules are also three short-term key milestones. The first, will allow SimMobilityST to be used in the decision process of the design of the electrical vehicles grid and to model the associated changes in the mobility patterns at the city level. The environmental and safety impacts assessment modules will allow a more comprehensive evaluation of the technologies and services being tested within SimMobility.

AKNOLEDGEMENTS

We thank Max Zhiyong and Seth Hetu for key contributions to the early developments of SimMobility Short-Term. Lastly, we thank Land Transport Authority (LTA) of Singapore for providing data for this research study. The research is supported by the National Research Foundation, Prime Minister’s Office, Singapore, under its CREATE programme, Singapore-MIT Alliance for Research and Technology (SMART) Future Urban Mobility (FM) IRG.

BIBLIOGRAPHY

[1] Barceló, J. (ed.), 2010. Fundamentals of Traffic Simulation, 1st Edition. Springer.

[2] Pipes, L. A., 1953. An operational analysis of traffic dynamics. Journal of applied Physics 24 (3), 274–287.

[3] Hranac, R., Gettman, D., Toledo, T., Kovvali, V., Vassili Alexiadis, 2004. NGSIM Task E.1-1: Core Algorithms Assessment. Tech. Rep. Federal Highway Administration, USA.

[4] TSS, 2011. Aimsun Dynamic Simulator Users' Manual. Version 7.0, Barcelona, Spain.

[5] PTV, 2009. VISSIM 5.20 User Manual. Tech. rep., Planung Transport Verkehr AG, Karlsruhe, Germany.

[6] Quadstone Paramics, 2009. The Paramics Manual. Version 6.6. 1. Edinburgh, Scotland, UK.

[7] Caliper, 2008. Traffic Simulation Software: TransModeller Users' Guide. MA, USA.

[8] Hidas, P., 1998. A car-following model for urban traffic simulation. Traffic engineering & control 39 (5), 300–305.

[9] Federal Highway Administration, 2006. CORSIM User's Guide Version 6. FHWA, USA.

[10] Liu, R., 2010. Traffic simulation with DRACULA. In Fundamentals of Traffic Simulation, pp. 295-322, Springer New York, USA.

[11] Koskinen, K., Kosonen, I., Luttinen, T., Schirokoff, A., Luoma, J., 2009. Development of a nanoscopic traffic simulation tool. Advances In Transportation Studies - An International Journal Section B 17.

[12] Rakha, H. 2014, INTEGRATION Rel. 2.40 for Windows - User's Guide, Volume I: Fundamental Model Features, V. Aerde and Associates, Ltd., Blacksburg, Virginia, USA.

[13] Ben-Akiva, M., Koutsopoulos, H. N., Toledo, T., Yang, Q., Choudhury, C. F., Antoniou, C., & Balakrishna, R., 2011. Traffic Simulation with MITSIMLab. In Fundamentals of Traffic Simulation, 145, 233.

[14] Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D., 2011, Sumo-simulation of urban mobility-an overview. In SIMUL 2011: The Third International Conference on Advances in System Simulation (pp. 55-60).

[15] CubeDynasim, 2013. http://www.citilabs.com/new-dynasim-4. CityLabs.

[16] TU0903-Cost Action, 2012. http://www.multitude-project.eu/.

[17] TRANSIMS, 2014: Transportation Analysis Simulation System. http://ndssl.vbi.vt.edu/transims-docs.html. Accessed online June to July 2014.

[18] MATSIM, 2014. Multi Agent Traffic SIMulation. http://www.matsim.org. Accessed online June to July 2014.

[19] Bellemans T, Bruno K, Janssens D, Wets G, Arentze T and Timmermans H, 2010. Implementation Framework and Development Trajectory of the Feathers Activity-Based Simulation Platform, Transportation Research Record-Journal of Transportation Research Board, No. 2175, pp: 111-119

[20] Krajzewicz, D., Erdmann, J., Behrisch, M. and Bieker, L., 2012. Recent Development and Applications of SUMO – Simul ation of Urban Mobility. International Journal on Advances in Systems and Measurements, Vol 5, N. 3 & 4, pp. 128-138.

[21] Toledo, T., Koutsopoulos, H., Ben-Akiva, M. E., 2007. Integrated driving behavior modeling. Transportation Research Part C: Emerging Technologies 15 (2), 96-112.

[22] Ahmed, K., 1999. Modeling Drivers' Acceleration and Lane Changing Behavior. Ph.D. thesis, Massachusetts Institute of Technology

[23] Yang, Q., Koutsopoulos, H. N. and Ben-Akiva, M. E., 2000. "Simulation Laboratory for Evaluating Dynamic Traffic Management Systems". Transportation Research Record: Journal of the Transportation Research Board, 1710, 122-130.

[24] Choudhury, C. F., 2007. Modeling Driving Decisions with Latent Plans. Phd thesis, Massachusetts Institute of Technology.

[25] L. Bowman. 2001. Activity-based disaggregate travel demand model system with activity schedules. Transportation Research Part A 35: 1–28.

[26] Ben-Akiva, Moshe, John L. Bowman, and Dinesh Gopinath. 1996. Travel demand model system for the information era. Transportation 23: 241–266.

[27] Adnan, M., Carrion, C., Basak, K., Hetu, S., Zhiyong, W., Vahid, S., Loganathan, H., Yang, L., Pereira, F., Ben-Akiva, M. SimMobility Midterm Framework: A state of the art Integrated Agent Based Transport Demand and Supply Simulator. Submitted to the 94th Annual Meeting of the Transportation Research Board, January 2015, Washington D.C., USA.

[28] Basak, K., Hetu, S., Li, Z., Lima Azevedo C., Loganthan, H., Toledo, T., Xu, R., Peh, L. S., Ben-Akiva, M., 2013. Modeling reaction time within a traffic simulation model. Annual Meeting of the Transportation Research Board, Washington D.C., USA.

[29] Daizong, L., 2003. Comparative evaluation of dynamic TRANSYT and SCATS-based signal control systems using Paramics simulation. M.Sc. Thesis in Civil Engineering. National University of Singapore, Singapore.

[30] Chovan, J., Tijerina, L., Alexander, G., Hendricks, D., 1994. Examination of Lane Change Crashes and Potential IVHS Countermeasures. Tech. Rep. March, US Department of Transportation, NHTSA, Washington D.C., USA

[31] Sarvi, M., 2013. Heavy commercial vehicles-following behavior and interactions with different vehicle classes. Journal of Advanced Transportation 47 , pp. 572–580.

[32] Nourinejad, M., Wenneman, A., Nurul Habib, K., Roorda, M. J., 2014. Truck parking in urban areas: Application of choice modelling within traffic microsimulation. Transportation Research Part A 64 (2014) 54–64.

[33] Teo, J., Taniguchi, E. and Qureshi, A. G., 2012. Evaluation of Distance-Based and Cordon-Based Urban Freight Road Pricing in E-Commerce Environment with Multiagent Model. Transportation Research Record: Journal of the Transportation Research Board, No. 2269, Transportation Research Board of the National Academies, Washington, D.C., 2012, pp. 127–134.

[34] Hetu, S., Vahid, S. and Peh, L.S., 2014. Similitude: Interfacing a traffic simulator and network simulator with emulated Android clients. Accepted for presentation in IEEE 79th Vehicular Technology Conference, 18–21 May 2014, Seoul, Korea.

[35] Gao, J. and Peh, L.S., 2014. RoadRunner: Infrastructure-less Vehicular Congestion Control, Accepted for presentation in the 21st World Congress on Intelligent Transport Systems (ITSWC), Detriot, Michigan, September 2014.

[36] Lu, L., Xu, Y., Antoniou, C., & Ben-Akiva, M. 2015. An enhanced SPSA algorithm for the calibration of dynamic traffic assignment models. Transportation Research Part C: Emerging Technologies, 51,

149-166.

[37] Antoniou, C., Azevedo, C. L., Lu, L., Pereira, F., & Ben-Akiva, M. 2015. W-SPSA in practice: Approximation of weight matrices and calibration of traffic simulation models. Transportation Research Part C: Emerging Technologies, 59, 129-146.

[38] Lu, Y., Adnan, M., Basak, K., Pereira, F. C., Carrion, C., Saber, V. H., and Ben-Akiva, M. (2015). Simmobility mid-term simulator: A state of the art integrated agent based demand and supply model. In Transportation Research Board 94th Annual Meeting (15-3937).

[39] Ramming, M. S. (2001). Network knowledge and route choice (Doctoral dissertation, Massachusetts Institute of Technology).

[40] Ciuffo, B., & Azevedo, C. L. (2014). A sensitivity-analysis-based approach for the calibration of traffic simulation models. IEEE Transactions on Intelligent Transportation Systems, 15(3), 1298-1309.

[41] Vaze, V., Antoniou, C., Wen, Y., & Ben-Akiva, M. (2009). Calibration of dynamic traffic assignment models with point-to-point traffic surveillance. Transportation Research Record: Journal of the Transportation Research Board, (2090), 1-9.

[42] Antoniou, C., Azevedo, C. L., Lu, L., Pereira, F., & Ben-Akiva, M. (2015). W-SPSA in practice: Approximation of weight matrices and calibration of traffic simulation models. Transportation Research Part C: Emerging Technologies,59, 129-146.

[43] Lu, L., Xu, Y., Antoniou, C., & Ben-Akiva, M. (2015). An enhanced SPSA algorithm for the calibration of dynamic traffic assignment models. Transportation Research Part C: Emerging Technologies, 51, 149-166.

[44] Lu, X. Y., Varaiya, P., Horowitz, R., & Palen, J. (2008). Faulty loop data analysis/correction and loop fault detection. In 15th World Congress on Intelligent Transport Systems and ITS America's 2008 Annual Meeting.

[45] Tachet, R., Santi, P., Sobolevsky, S., Reyes-Castro, L. I., Frazzoli, E., Helbing, D., & Ratti, C. (2016). Revisiting street intersections using slot-based systems. PloS one, 11(3), e0149607.

[46] Zhu, Y. and J. Joesph Ferreira, Synthetic population generation at disaggregated spatial scales for land use and transportation microsimulation. Transportation Research Record: Journal of the Transportation Research Board, , No. 2429, 2014, pp. 168–177.

[47] Teo, J., Cheah, L., Lee, Y.J., Marzano, V., Santos, J., Lima Azevedo, C., Zhao, F., Ben-Akiva, M. (2015). An integrated sensing-based urban freight data collection framework: methodology and pilot projects in Singapore. URBE conference, Oct 1-2, 2015, Rome, Italy.

[48] Lima Azevedo, C., Marczuk, K., Raveau, S., Soh, H., Adnan, M., Basak, K., ... & Ben-Akiva, M. (2016). Microsimulation of Demand and Supply of Autonomous Mobility On-Demand. In Transportation Research Board 95th Annual Meeting (16-5455).

[49] Atasoy, B., Ikeda, T. and Ben-Akiva, M. (2015), Optimizing a Flexible Mobility on Demand System, Transportation Research Record (TRR), Vol. 2536, pp. 76-85.

[50] Le, D.T., Cernicchiaro, G., Zegras, C., Ferreira, J. (2016). Simulation of synthetic establishments for modeling firm behavior in SimMobility. International Scientific Conference on Mobility and Transport Transforming Urban Mobility, mobil.TUM 2016, 6-7 June 2016, Munich, Germany.