

Research Article

Trajectory Simulation of Emergency Vehicles and Interactions with Surrounding Traffic

Cristián E. Cortés ¹ and Bruno Stefoni ²

¹Department of Civil Engineering, Universidad de Chile and Instituto Sistemas Complejos de Ingeniería (ISCI), Blanco Encalada 2002, Santiago, RM 8370449, Chile

²Department of Civil Engineering, Universidad de Chile, Blanco Encalada 2002, Santiago, RM 8370449, Chile

Correspondence should be addressed to Cristián E. Cortés; ccortes@ing.uchile.cl

Received 2 July 2022; Revised 19 December 2022; Accepted 24 December 2022; Published 22 February 2023

Academic Editor: Shamsunnahar Yasmin

Copyright © 2023 Cristián E. Cortés and Bruno Stefoni. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Emergency services play an important role in the life of a city and are subject to constant public scrutiny. The efficient dispatch of emergency vehicles (EMVs) requires realistic shortest-path algorithms involving the movement of EMVs within an urban network under emergency conditions. Trip-time estimates used in shortest-path algorithms would be much more precise if it were possible to model more realistically the interactions between EMVs and surrounding traffic, as well as the reactions of other vehicles in the presence of an EMV. Therefore, EMV trajectories should be studied at the microscopic level to accurately model the impact of EMV travel along a path shared with other vehicles. In this research, we develop three models to incorporate specific non-EMV reactions associated with changing lanes, mounting the sidewalk, and approaching an intersection, plus two algorithms to actuate traffic lights at signalized intersections. These models and algorithms were coded in commercial microscopic traffic simulation software through the implementation of an application programming interface (API) designed to overcome the limitations of the software to realistically simulate disturbed traffic conditions and anomalous nonemergency vehicle driver behaviour observed in the presence of an EMV. Basic information about these real-world effects was gleaned from video footage recorded in Santiago, Chile, by traffic cameras, fire truck-mounted cameras, and truck-originated GPS pulses. To validate the design, a real EMV trip captured by the footage was simulated by the API. The simulation considerably reduced the degree of error in delineating the path followed by the EMV compared to the default simulations generated by most commercially available software, thereby demonstrating that the API can provide highly accurate estimates of EMV trip times in an emergency context.

1. Introduction

Emergency services play a fundamental role in the life of a city. By their very nature, they are always in the eye of the media and thus are subject to constant public scrutiny. This reality and the impact sluggish emergency response times would have on the public explain why entities responsible for these services must ensure they are operated as efficiently as possible. In the Chilean capital of Santiago, emergency services are the responsibility of a number of different organisations, including *Carabineros de Chile*, the uniformed police force; *Socorro Andino*, the mountain rescue service; and the *Cuerpo de Bomberos de Santiago* (CBS), Chile's largest

company of firefighters that is responsible for fire extinguishment and debris cleanup in nine of the city's districts.

Technical experts at the CBS have been engaged in research to determine the possible effects on emergency response times of changes currently being implemented in the street network of Santiago's historic centre. An initial study involved building street network models to identify the shortest route to emergency sites using real-time traffic data derived from GPS signals generated by the city's buses [1]. A more refined analysis revealed that trip-time estimates used in shortest-route algorithms would be much more precise if it were possible to model more realistically the interactions between emergency vehicles (EMVs) and surrounding

traffic, as well as the reactions of other vehicles that find themselves on an EMV's route.

Both of these phenomena occur in a disturbance context relative to a normal situation, suggesting that their impact on EMV trip times may be considerable. This might be the case, for example, in a situation where traffic is highly congested and there is insufficient room for non-EMVs to drive up onto the sidewalk to let an EMV pass. Moreover, non-EMV drivers' reactions to the presence of an EMV differ from person to person and are highly stochastic, thus constituting another variable that influences EMV trip times. A good EMV dispatch system should therefore incorporate a correction factor into its route algorithms to take into account phenomena such as congestion, infrastructure, driver diversity, and traffic signal behaviour.

In light of the foregoing discussion, EMV operations must be studied at the microscopic level to accurately model the impact of EMV travel along a path that is shared with other vehicles.

The main contribution of the present article is the development of a series of driver behaviour models and algorithms for capturing specific traffic situations that arise when EMVs are on their way to an emergency, implemented at a microsimulation level. Specifically, we develop a lane-changing model, a sidewalk model arising when traffic congestion is observed, and an intersection approach model. In addition, for reasons explained later in this paper, we develop special algorithms for the treatment of EMVs approaching traffic lights.

The proposed implementation innovates with respect to the state-of-the-art literature, which is reviewed in some detail in next Section 2. A major achievement of our approach is the simulation of EMVs by incorporating lane, sidewalk, intersection, and traffic light models for EMV interactions simultaneously within a simulation platform built at a microscopic level. Furthermore, all these interactions are modeled with clearly defined parameters that can be calibrated to better fit the emergency scenario with all the commonly seen particular influences an incoming EMV has over normal traffic. The resulting comprehensive simulation at the microscopic level of normal traffic interactions with an incoming EMV allows for richer emergency event simulations.

The models were implemented in commercial traffic simulation software called Paramics, which is described in Kachroo and Ozbay [2] chapter 12 and in Barceló [3] chapter 4, through a flexible application programming interface (API) that calls specific developed functions when EMVs either travel along a multilane street, travel along a congested street with sidewalks, or approach an intersection. In each of these situations, the corresponding part of the API takes control of the simulator. The API is executed simultaneously with the simulation and may be implemented concurrently at different points in the simulated network or at a single point with the same EMV. The latter situation might occur if, for example, an EMV approaches a signalized intersection, activating the intersection model as well as special traffic signal algorithms.

The implementation of these models in the simulator via the API overcomes certain limitations of traditional microsimulation software, such as AIMSUN, VISSIM, SUMO, CORSIM, and Paramics, to generate a more realistic representation of the traffic conditions and anomalous non-EMV driver behaviour surrounding the movement of an EMV.

Since the proposed tool had to be calibrated with real information, we used GPS trajectory data together with audiovisual material recorded by CBS fire trucks along their emergency trips' routes, to capture the surrounding conditions as experienced by an EMV driver.

Loading this new API into the microsimulation software enabled us to run simulations and evaluate variations in fire trucks' response times for different traffic network configurations. The API parameters varied on the basis of the field data from the CBS material to compare and contrast the real-observed trajectories with the simulated ones and thus assess the quality of the proposed tool. In addition, a methodology was developed for estimating the basic parameters needed for the microsimulator from the real data, and a network was coded to simulate an EMV trip registered by traffic cameras and GPS data. Finally, network demand was calibrated with a view to achieving the best fit to reality.

The first step in designing the microscopic EMV interaction models was the analysis of the CBS's audiovisual material, captured by fire truck-mounted cameras recording emergency trips from the EMV driver's standpoint. The most frequent reactions observed were crossing intersections without the right of way, using restricted or bus-only lanes, forcing surrounding vehicles to change lanes or if traffic appeared to be stopped, pulling up onto the sidewalk, and forcing other vehicles to cross intersections, regardless of whether they were signalized, without the right of way.

Even though the API was coded in Paramics, the methodology is generic as it is based on behavioural models developed after observing the traffic behaviour from different sources, mainly from the CBS audiovisual material. Therefore, we strongly believe that adapting these models to another microsimulation tool should be feasible by properly choosing the functions and parameters to intervene in each specific case. The specific simulation tool used in this work was selected due to the flexibility of its API implementations. We understand that Paramics is currently migrating to a new platform called Paramics Discovery (<https://www.paramics.co.uk/en/>); thus, a natural continuation of this research would be the development of new APIs containing the same functions adapted to this new platform or to another state-of-the-art traffic microsimulation platform.

Our primary objective was thus to simulate EMVs in an urban street network with all of its impacts on ordinary traffic to evaluate the effects on EMV response times in transport projects involving simulations. This effort is described in the four sections that follow. Section 2 reviews the related literature, especially regarding the use of commercial microscopic traffic simulators such as Paramics. Section 3 details our proposed API and its constituent models and algorithms. Section 4 analyses the results for EMV response times and non-EMV trip times in a test network with

different levels of congestion and for non-EMV cooperation with EMVs in emergency situations. Finally, Section 5 presents our conclusions and some comments on future research.

2. Literature Review

Our examination of the relevant literature focuses on the reactions to EMVs of non-EMV drivers observed in the CBS videos noted above. Buchenscheit et al. [4] developed a communication system for privately owned vehicles (POVs), which alerts them to the approach of an EMV; in the present article, we conduct an exhaustive analysis of the behaviour of POV drivers based on an examination of many EMV videos. Weinert and Düring [5] used the Simulation of Urban MObility (SUMO) traffic simulation package to simulate POV flows, with automated formation of rescue lanes, when an EMV is present. The authors developed an API to give EMVs special capabilities and an algorithm for traffic signal pre-emption. Weinert et al. [6] developed this methodology further, designing a pre-emption system in a real-traffic environment that is evaluated in terms of trip times and traffic safety indicators. Behavioural models for both EMVs and POVs are built and integrated into their traffic simulators. Bieker-Walz [7] evaluated strategies for cooperation between EMVs and POVs using SUMO in a simulation of scenarios in Bologna, Italy. Lu and Kim [8] proposed an intersection control system based on vehicle-to-vehicle (V2V) communication to provide priority to EMV crossing based on a genetic algorithm to determine the optimal vehicle passing sequence in the intersection to give EMVs the highest priority. The scheme is tested using SUMO, using TraCI for the interaction of the control algorithms written in Python with the microsimulator. Bieker-Walz et al. [9] built models specifically for EMVs in a SUMO simulation setting using speed adjustment models, exclusive emergency lanes, and traffic light violations. Since the last technique, though possible in SUMO using the TraCI interface [10], is not supported by Paramics, we develop a special module that emulates red-light violations to capture what in real settings is a common EMV practice.

Zhang et al. [11] used CORSIM to simulate EMV behaviour, which includes driving on shoulders, crossing medians, driving through red lights, crossing intersections without the right of way, making prohibited turns, and driving against the direction of traffic. EMVs follow fixed routes, and the probability of a POV's awareness of an EMV's presence is a function of the distance between them. A POV that cooperates with an EMV is temporarily assigned the maximum aggressiveness level, following which the situation returns to normal. Cooperative POV actions include changing lanes, modifying speed, and moving to the right shoulder. If a POV is at a signalized intersection and an EMV is immediately behind, the POV crosses against red. The work by Zhang et al. [11] is quite complete and covers many of the relevant features we have added to our model, with the exception of a sidewalk model that we have incorporated as part of our simulation tool. Note that in Zhang et al. [11], EMVs can drive in the opposite direction of traffic,

unlike our model, which does not simulate these banned movements in the trajectory of EMVs.

Hannoun et al. [12] proposed an emergency assistance system with an EMV routing optimization model that is simulated in NetLogo. Behavioural parameters are adjusted to simulate EMV movement. Yoo et al. [13] conducted simulations using an IMAGES (intelligent multiagent system) to evaluate strategies for prioritizing EMVs, comparing different variations of POV pulling-over behaviour. Toy et al. [14] analysed a program for automated highway systems (with interconnected autonomous vehicles), which models three autonomous vehicle manoeuvres to permit priority EMV transit: vortex, part-and-go, and zigzag. The authors conducted simulations using SmartAHS and discussed the bubble-like effect of creating a moving low vehicle density area around an EMV. So et al. [15] presented an EMV control strategy that integrates automated driving technology and existing traffic signal pre-emption strategies, considering the impacts of automated driving on EMVs with the impact on mobility and safety using different aggressiveness scenarios. EMVs are equipped with on-board units able to perform V2X communication and sensors to detect adjacent vehicles. Moreover, traffic signals controllers are able to receive information from the on-board units installed in EMVs. They develop all the corresponding automated driving control models and signal pre-emption strategies, which are tested through microscopic simulation using VISSIM to properly control vehicles and manipulate traffic signals.

The literature also proposed models that incorporate variability of behaviour across the population of POV drivers given that their cooperation is not guaranteed. POV drivers are not generally homogeneous [16–18], and not all of them cooperate with EMV transit. Another reason for the behavioural uncertainty in POV drivers is that even those who are otherwise cooperative may inadvertently block the passage of an EMV if its siren is drowned out by traffic or street noise [19] or due to other distractions, such as the drivers' own cell phones [20].

In what follows, we classify in Table 1 the main features of relevant previous works related to simulations of EMVs and their interactions with the rest of the traffic when they are moving in a hurry towards an emergency; namely, the specific models the different authors put attention on (lane, sidewalk, intersection, and traffic light), the issue of calibration of such models, and finally the fact of considering either perfect cooperation or not in the simulation.

As shown in Table 1, most of the paper works include a lane model, referred to the change lane action of POVs to let EMVs pass properly, which is something we put particular attention by modifying the specific lane-changing models (LCMs) implemented in the microsimulator utilized (this issue is explained in detail later in this section). Apart from the lane model, we observe that the authors address either one or two out of the three discussed models in Table 1, namely, sidewalk, intersection, and traffic light. As mentioned above in Section 1, one innovative feature of the proposed approach in the present paper is to conceive and to implement the simultaneity of the three mentioned models

TABLE 1: Summary of modeling features found in relevant articles from the recent literature in this topic.

Article	Simulator	Lane model	Sidewalk model	Intersection model	Traffic light model	Calibration	Perfect cooperation
Weinert and Düring [5]	SUMO	Yes	No	Yes	Yes	No	Yes
Weinert et al. [6]	SUMO	Yes	No	Yes	Yes	Yes	Yes
Bieker-Walz [7]	SUMO	Yes	No	No	Yes	No	Yes
Lu and Kim [8]	SUMO	No	No	Yes	Yes	No	Yes
Bieker-Walz et al. [9]	SUMO	Yes	No	No	Yes	No	Yes
Hannoun et al. [12]	NetLogo	Yes	No	Yes	No	No	No
Zhang et al. [11]	CORSIM	Yes	No	Yes	Yes	Yes	No
Yoo et al. [13]	IMAGES	Yes	Yes	Yes	No	No	No
Toy et al. [14]	SmartAHS	Yes	No	No	No	No	Yes
So et al. [15]	VISSIM	Yes	No	Yes	Yes	No	Yes

into the same modeling platform. Previously in this section, we highlighted the work by Zhang et al. [11] as an important reference given the similarities between their approach and ours in some key aspects of the simulation. Another feature that is worth highlighting is that most of the articles do not develop a proper process of calibration in their works, and as explained in detail in Section 4, we develop several calibration procedures to make the simulation as real as possible. Specifically, we propose calibrating our EMV simulation parameters to fit the distance-time curve obtained from real GPS data of EMV expeditions. This helps obtain an objective goodness-of-fit measure from the simulations that can be optimized. In addition, reporting this measure makes possible benchmarking results with other EMV microsimulation research in the future.

With regard to some important features of the internal models that control the principles of traffic microsimulations, we highlight the car-following model (CFM), the lane-changing model (LCM), and the gap acceptance model (GAM). A comparative analysis of various pieces of microscopic traffic simulation software can be found in works by Kotusevski and Hawick [21].

As regards the CFM in the Paramics package, developed by Quadstone Ltd. [2, 3], it is classified by Brackstone and McDonald [22] as an action point model, while Saifuzzaman and Zheng [23] categorize it as a perceptual threshold model. These types of models dictate a level of acceleration for simulated vehicles until they arrive at a certain threshold defined on the basis of two variables, headway and speed difference, measured at each discrete instant of the simulation. Duncan [24] describes the psychophysical car-following model implemented in Paramics, based on the CFM proposed by Fritzsche [25], in which drivers try to maintain a certain target headway relative to the leading vehicle. If the leading vehicle is absent, drivers aim for their comfortable speed. In the former case, this is achieved by accelerating or decelerating, and the magnitude and direction of this action depend on the current headway and the speed difference (see Figure 1).

Another key aspect of Paramics is the use of the random variables' *aggressiveness* (Ag) and *awareness* (Aw) to model target headway variations in the driver population. Changes in the driving style and target headway have been linked to driver heterogeneity in factors such as age, gender, and

emotional state [16–18]. The mean target headway (MTH) parameter is set to 1 second by default, but for actual drivers, it varies depending on their individual values of Ag and Aw . These two variables assume integer values ranging from 0 to 8; the values are initially set randomly at the moment each vehicle is created and can be modified over the course of the simulation, depending on routing and traffic conditions. For example, if a driver with high Aw (and therefore relatively aware) is going to turn left at the upcoming intersection, the indirect effects of Ag and Aw through the LCM will be such that the driver will increase their target headway, facilitating the entry of vehicles seeking to change to an adjacent lane. By contrast, a driver with high Ag (i.e., relatively aggressive) will have a lower target headway, hindering other vehicles seeking to change into that driver's lane. Finally, Paramics also features the mean reaction time (MRT) parameter, which is used to simulate the mean temporary delay in vehicle queue dissipation that is observed in real-traffic situations. Both MTH and MRT are generally calibrated to reproduce the real data employed in the user's simulation study as accurately as possible [26–28].

3. API Emergency Vehicle Models

The API developed to incorporate the observed effects of EMVs on surrounding traffic into the Paramics microsimulator allows the reactions of POVs to be modified in the presence of an EMV. It consists primarily of three models: the EMV lane-changing model (ELCM), the EMV sidewalk model (ESM), and the EMV intersection model (EIM). To handle cases where there is more than one EMV on a network arc, an EMV-following rule is also programmed into the API. In addition, since Paramics never permits a vehicle to cross an intersection on a red light, algorithms are implemented to permit ad hoc traffic light actuation. Notice that the developed algorithms for traffic signals could be adapted and used in the context of any other simulator with the same constraint as the Paramics rule of not crossing on red to model the realistic observed behaviour of EMVs crossing many intersections in the red phase when travelling to an emergency.

Implementation of these models and algorithms was made possible by the flexibility of the library of functions for programming APIs in Paramics' Programmer module. According to Zhi et al. [29], the great variety of these

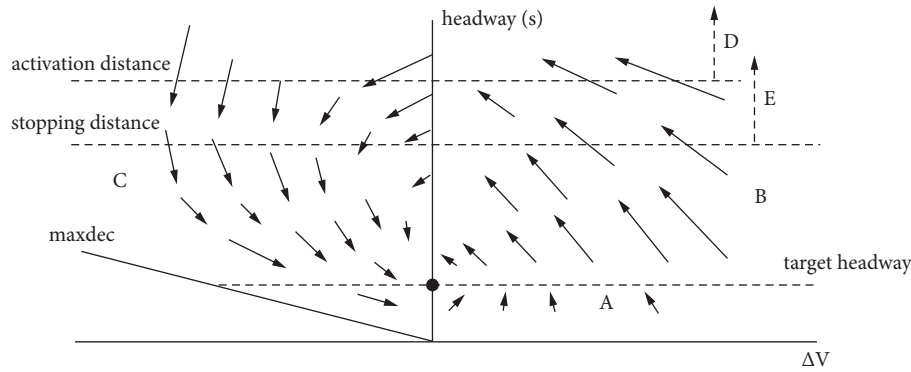


FIGURE 1: “Representation of the headway/velocity-difference phase space and the resulting acceleration values produced by the Paramics car-following algorithm. The magnitude of an arrow is a representation of the acceleration derived by the car-following algorithm for a given $(\Delta V, h)$ pair. Note that the arrows are not to scale” Duncan [24].

functions is one of the software package’s major advantages over other traffic microsimulators, allowing advanced users to reconfigure various basic aspects of a simulation. In the present case, thanks to the transparency and dynamic nature of the communication between the traffic microsimulator and our API at all times during simulations, the interrelationships between the various actions of the different agents triggered by the approach of an EMV could be effectively modeled.

3.1. CBS Audiovisual Material. To determine the behaviour of POV drivers surrounding an EMV that is en route to an emergency, we studied the videos recorded by the CBS fire truck cameras at various times of the day along different classes of streets. Most of the videos were recorded between April and September in 2013, 2015, and 2016 and were made available for the present study over the course of 2016.

Some of the recordings begin at a fire station, while some start at a point on the way to the emergency. Others captured the trip back to the station when the EMV was not in emergency mode and were therefore discarded. The great majority of the videos were recorded by cameras installed on truck windshields, and the remaining footage was filmed from cameras mounted on vehicle sides. In some cases, neither the truck nor its emergency trip trajectory could be identified. A list of the videos indicating their duration and the emergency trip trajectory recorded is given in Table 2.

3.2. Fundamental EMV Model Considerations. Examination of the CBS audiovisual material revealed the following behaviour patterns of POV drivers:

- (i) The distance from the EMV at which they begin to cooperate varies depending on the driver, the traffic congestion level, and the manoeuvre the POV must perform to facilitate the EMV’s transit.
- (ii) They do not change lanes if the EMV is in a different lane.
- (iii) Upon noticing an EMV trying to cross an intersection, they stop and yield even if the EMV does

TABLE 2: CBS videos indicating sequence of streets in the vehicle trajectory and duration of the recording.

Emergency trip trajectory	Duration
Abate Molina—Grajales	0:15:02
Eduardo Matte—Victoria	0:05:00
Domey—Carrera	0:08:01
Ventura Lavalle—Portugal	0:12:01
Maule—San Francisco	0:11:01
Pedro Montt—Fábrica	0:06:01
San Isidro—Cóndor	0:09:01
Pedro Montt—Luis Cousiño	0:15:31
Glaziau—Pedro Lagos	0:08:00
Arturo Prat—Arauco	0:06:00
San Diego—Eyzaguirre	0:04:00
San Diego—Victoria	0:05:01
Sierra Bella—Matta	0:03:00
Santa Elena—Ñuble	0:06:00
Dieciocho—Alameda	0:12:10
15 videos with no identification	1:35:20
Total	3:40:09

not have the right of way (i.e., the EMV’s approach has a yield sign, stop sign, or red light).

- (iv) They cross an intersection even though the intersection approach they are on does not have the right of way if an EMV behind them is trying to cross.
- (v) They change lanes to let an EMV pass, thus clearing the way for the EMV and minimizing the number of lane changes it must make.
- (vi) In situations of extreme congestion, upon noticing that traffic has stopped, some light vehicle drivers pull up onto the sidewalk to let an EMV pass.

Note that, due to certain limitations of the software, the first two of the above patterns (cooperation distance variation and not changing to a less congested lane) were not incorporated into the microsimulator. The EMV-related traffic changes induced in Paramics by our API are as follows:

- (i) Vehicles change lanes in advance upon perceiving the presence of an EMV upstream of them at a certain distance.

- (ii) Vehicles yield to an EMV at intersections even if the EMV does not have the right of way or, in the case of a signalized intersection, has a red light.
- (iii) Vehicles at an intersection stop line cross the intersection even though they do not have the right of way if one of the three vehicles immediately behind them is an EMV.
- (iv) EMVs can use exclusive buses and taxi lanes.
- (v) When traffic is congested, light vehicles pull up onto the sidewalk where possible to let an EMV pass.

These modifications are performed in the simulations through the API in accordance with the functioning of models, as described in the subsections below. The API itself works as follows: Upon the launch of a simulation, at each discrete moment (known as a time step), the software checks each arc in the network. In this sense, models are based on the individual arcs' properties, which include the vehicles travelling along the arcs at each time step. The speed and distance to the stop line of each vehicle can be checked. As regards the arcs' traffic direction, although a street can be coded graphically (in the user interface) for both directions, internally, the traffic simulation software interprets two-way streets as two one-way arcs, one for each direction. Thus, an arc in the API is always one way.

The schematic framework assumed in the explanations of the models that follow is set out in Figure 2. It represents an arc consisting of two lanes and three vehicles, with traffic moving from left to right. A key variable is d_i , the distance from the front bumper of the vehicle i to the stop line of the signalized intersection ahead. The variable $l(h)$ is the length of vehicle h .

3.3. EMV Lane-Changing Model. To develop the EMV lane-changing model (ELCM), we introduce the four additional variables:

d_{\min} is the minimum distance to the intersection stop line of EMVs present on the arc.

d_{\max} is the maximum distance to the intersection stop line of EMVs present on the arc.

α_a is the upstream distance over which an EMV influences POVs.

β_a is the downstream distance over which an EMV influences POVs.

Note that, if there is only one EMV on an arc, then $d_{\min} = d_{\max}$. If there are multiple EMVs, the one closest to the stop line is denoted the leading EMV.

Vehicles in this model are said to be on alert if they behave in a way that facilitates the transit of any EMVs that are present. A vehicle i travelling along an arc with one or more EMVs must simultaneously satisfy equations (1) and

(2) to change to a state of alert. These two geometric conditions determine that a POV is on alert if it is located (i) between two or more EMVs, (ii) no more than β_a metres downstream of an EMV, or (iii) no more than α_a metres upstream of an EMV.

$$d_i \geq d_{\min} - \beta_a, \quad (1)$$

$$d_i \leq d_{\max} + l(EMV) + \alpha_a. \quad (2)$$

An example of the state-of-alert concept for an arc with two EMVs is depicted in Figure 3. The two POVs m and n satisfy the above-stated conditions, while POVs h and k do not. Indeed, k would not be on alert even if part of it was within the "alert zone" because in the API, the distance d_k in the two conditions is measured from the front bumper.

Once a vehicle is on alert, it behaves as follows:

- (i) If it is not in the EMV's lane, it will not change to that lane to avoid interfering with the EMV's transit.
- (ii) If it is in the EMV's lane, it will attempt to change lanes to facilitate the EMV's transit.
- (iii) If it is in the EMV's lane but cannot change lanes due to congestion, it will continue advancing while evaluating the situation, that is, observing whether the leading EMV remains in the same lane and reacting in accordance with these three points.

If, in the presence of an EMV, a POV is able to change lanes either to right or left, the API, upon perceiving this, decides which lane it will order the POV to change to by generating a random binomial variable whose parameter is $p = 0.5$. In other words, the API makes a random decision by "flipping a coin." Such a situation can arise only with arcs of more than two lanes where the POV in question is not in a curb lane. In the case where a POV attempts to change lanes but fails because Paramics's LCM default conditions regarding the necessary space in the target lane are not satisfied, the API orders the vehicle to wait 3 seconds and then makes another attempt. If the default conditions are still not met, the API generates and issues a new random lane-change order. The LCM does not govern POVs on arcs with a single travel lane.

In the case of a public transport bus, once it is in a state of alert, the API will direct its behaviour as follows:

- (i) If it is in the curb lane and the EMV is in some other lane, the API orders the bus not to change lanes.
- (ii) If it is not in the curb lane and the leading EMV is in the same lane, the API orders the bus to change to, or towards, the curb lane.
- (iii) If it and the leading EMV are both in the curb lane, the API orders the bus not to change lanes.

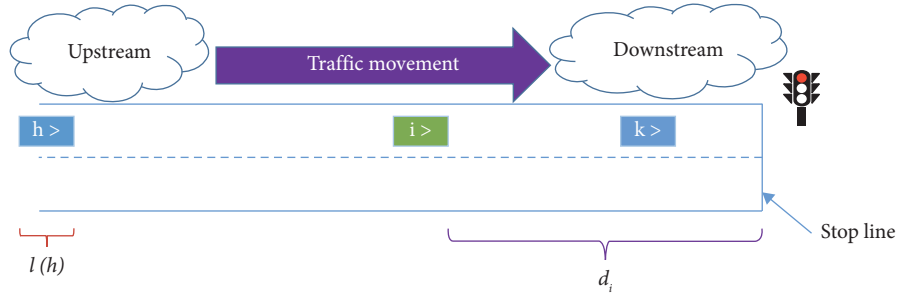
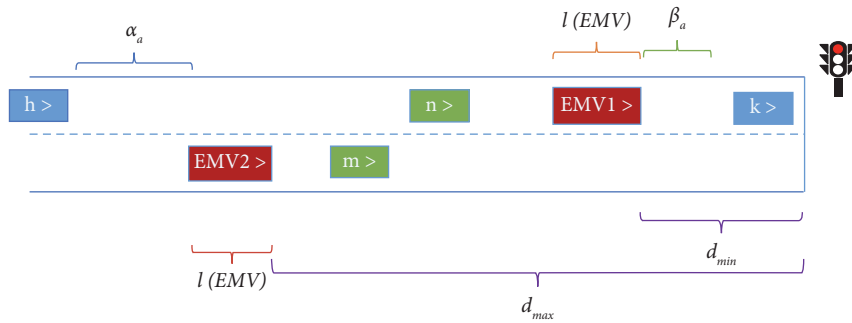


FIGURE 2: Schematic framework of an arc and its basic characteristics for the EMV models.


 FIGURE 3: Lane-changing model: POVs m and n are on alert, while h and k are not.

Finally, if there are multiple EMVs on an arc, nonleading ones, that is, those further from the stop line, are ordered by the API to “follow” the leading EMV by attempting to change to the latter’s lane (EMV-following) if they are not already in it. Such special orders are generated only for EMVs.

3.4. EMV Sidewalk Model (ESWM). To develop the sidewalk model, we introduce the five new parameters:

d_v^{\min} is the distance from the stop line at which vehicles are no longer subject to this model.

β_v is the downstream distance over which POVs perceive the presence of an EMV.

v_h is the speed of the vehicle h corresponding to the distance it covered in the last t_h seconds.

t_h is the time period (in seconds) ending at the current instant over which the speed v_h of the vehicle h is estimated.

s_a is the congestion speed, indicating the threshold below which an arc a is considered to be congested.

In this model, light vehicles (i.e., not trucks or buses) change their behaviour in the presence of an EMV if they are in the same lane and have not been able to change lanes either due to congestion in other lanes or because there is only one travel lane. If the arc has a sidewalk and the vehicle cannot advance due to congestion (determined by conditions to be explained in the next paragraph), it pulls up onto the sidewalk to let the EMV pass.

Before a light vehicle can pull up onto the sidewalk, it must first be on alert. Any such vehicle i has moved to the alert state in this model if it satisfies the conditions expressed by 3 and 4 below. If it also satisfies 5 and 6, it will attempt to pull up onto the sidewalk (Figure 4). The first two conditions reflect the presence of an EMV close behind the vehicle. The latter two indicate extreme congestion, which is considered to be present when the next three vehicles in the queue downstream are stopped and their speed in the moments before stopping is greatly reduced. This definition was adopted in light of the CBS videos, where it was observed that if drivers see few vehicles stopped in front of them, they will wait until the latter cross the intersection, sometimes without the right of way, and if there is a queue but it is advancing, they will not pull up onto the sidewalk. In the case where all four conditions are satisfied but the space where the vehicle could pull up onto the sidewalk is occupied by another vehicle, the former vehicle will abandon that option and continue advancing. After 5 seconds have passed, if the vehicle is still on alert, it checks whether it still satisfies all four conditions and, if so, makes another attempt to pull up onto the sidewalk.

$$\begin{aligned}
 d_i &\geq d_v^{\min}, \\
 d_{\min} &\geq d_i \geq d_{\min} - \beta_v, \\
 v_i &= v_j = v_k = v_l = 0, \\
 v_h &< s_a \quad \forall h \text{ present on arc } a.
 \end{aligned} \tag{3}$$

If a vehicle has pulled up onto the sidewalk, it will remain stationary there until the EMV has passed, and only then it

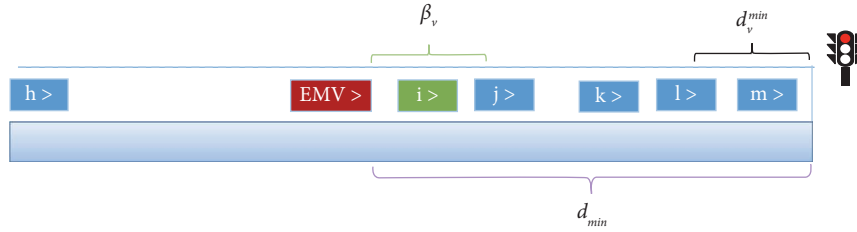


FIGURE 4: Sidewalk model: light vehicle i is on alert.

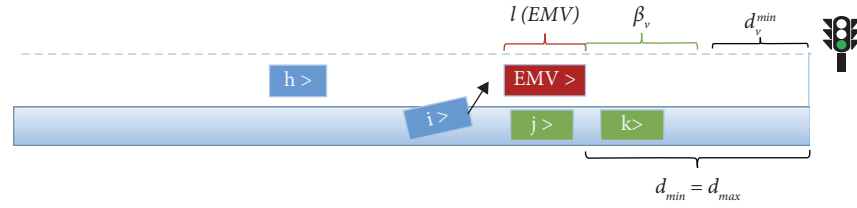


FIGURE 5: Sidewalk model: light vehicle i returns to traffic, while j and k do not.

will rejoin traffic. Mathematically, this means that it is static if it satisfies condition 4, as shown in Figure 5.

$$d_i \leq d_{\max} + l(EMV). \tag{4}$$

3.5. EMV Intersection Model (EIM). The intersection model applies in cases where an EMV is approaching an intersection stop line without having the right of way. The model incorporates only one additional parameter, δ_i , the distance between the point where the EMV is first perceived by POVs on other approaches to the intersection and those approaches' respective stop lines. POVs located less than δ_i metres from the stop line must yield to the EMV, even if they have the right of way.

A POV downstream in the same lane as an EMV approaching the intersection crosses it even if it does not have the right of way (see Figure 6); however, this is not the case if the vehicle is in a different lane. If there are multiple EMVs approaching the intersection, POVs are governed by the usual rules of the road in determining which EMV crosses first.

Once the EMV has crossed the intersection, the situation at the intersection returns to normal. The API then orders the vehicles that were forced to yield to the EMV to resume travel according to the usual rules of the road. Finally, to avoid internal conflicts between the intersection and sidewalk models, it is recommended that the latter's d_v^{\min} parameter be equal to or greater than δ_i . The default value for present purposes is $d_v^{\min} = \delta_i$.

3.6. Parameters and Predetermined Values. All three of the abovementioned models have adjustable parameters whose initial values are based on rough estimates derived from the CBS videos. The lane-changing model has two parameters, α_a and β_a ; the sidewalk model has three, β_v , t_h , and s_a ; and the intersection model has one, δ_i , which, as noted above, is assumed to equal d_v^{\min} in the sidewalk model. The reason for

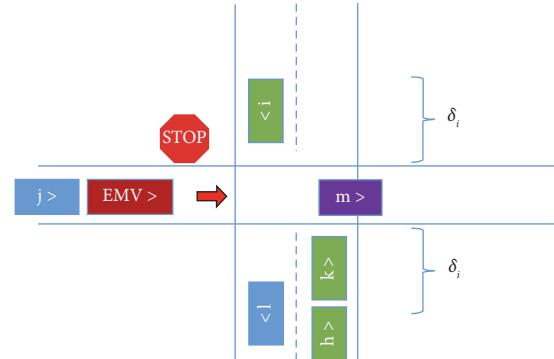


FIGURE 6: Intersection model: vehicle m crosses without the right of way.

this supposition stems from the fact that d_v^{\min} is the distance from the stop line within which a POV is no longer subject to the sidewalk model, while δ_i is the distance from the stop line within which it is subject to the intersection model if an EMV is present. In other words, δ_i metre from the stop line is the point at which a POV passes from the control of the sidewalk model to the control of the intersection model.

A cooperation factor parameter (CF) ranging from 0% to 100% is included in the API to represent the proportion of POVs that cooperate with EMVs. Specifically, it is the percentage of vehicles that obey API orders (i.e., cooperate) when an EMV is present. The idea is to capture the variation in driver behaviour given that the videos revealed that some drivers do not cooperate, either because they are unaware of the EMV's presence or out of a simple lack of willingness. To simulate this behaviour, a binomial random variable with a parameter $p = CF$ is generated for each POV upon its creation in the simulator. If the variable takes a value of 0, the vehicle will cooperate whenever an EMV is present; otherwise, the vehicle never cooperates. With this mechanism, the average cooperation factor percentage will be equal to CF . Note that to estimate it, the vehicles that do and do not

cooperate must be counted under ideal traffic conditions so that a lack of cooperation is not confused with impossibility to cooperate.

In addition to the above parameters, which are part of the API, the microsimulation software has a number of parameters, as mentioned in the previous section. As stated by Ratrouf et al. [27], the most important ones are the MTH (mean target headway) and MRT (mean reaction time). By default, these parameters are set to 1 second.

As noted above, the API parameters were estimated based on a study of the CBS videos. The values assigned were as follows: $d_v^{\min} = \delta_i = 15\text{m}$, $t_h = 10\text{s}$, $s_a = 2\text{ (m/s)}$, $\alpha_a = 5\text{m}$, $\beta_a = 40\text{m}$, $\beta_v = 30\text{m}$, and $\text{CF} = 80\%$. Though not the result of a rigorous estimation or calibration, these values are used as the defaults in the example discussed in Section 4.8. A set of estimates derived from a rigorous calibration process based on recordings of an emergency trip are presented in Section 5.

3.7. Special Algorithms for Traffic Signals. Although the intersection model applies to all intersections including those with traffic lights, the behaviour of vehicles at a signalized intersection in the presence of an EMV was observed to display some special characteristics that are not accommodated in the three models described above. In such situations, POVs completely ignore the traffic signal phase, stopping even if the light is green to let an EMV pass or crossing the intersection even though the light is red so as not to block an EMV behind them. Paramics does allow vehicles to be ordered to yield (i.e., not cross the intersection) on a green light but unfortunately does not allow a vehicle to cross on red, even if it is an EMV. The latter situation thus had to be simulated indirectly.

To achieve a simulation for the signalized intersection case that closely mirrors reality, two special algorithms were implemented in the API to convert all traffic signals into actuated ones. The algorithms use real-time operating and demand data acquired by detectors at intersections to change one or more aspects of the current signal cycle, that is, alter the phase sequence, green time in each phase, or cycle length. An example of an actuated traffic signal is a change in the phase that allows a movement with very low demand, i.e., a movement desired by a very small proportion of the traffic flow. If a vehicle in the queue seeking to perform that a manoeuvre is detected, the corresponding phase is activated; otherwise, no change is actuated.

The API activates the traffic signals every time an EMV comes within δ_i of the stop line, and the approach it is on has either a red light or a green light with less than two seconds before the phase terminates. In either case, the control of the signals is taken over by Algorithms 1 and 2 set out below.

The state of a traffic signal at a given intersection is controlled in Paramics through a function in its library denoted as `qps_SIG_action` (intersection, phase, . . . , green time), where intersection is the number assigned to the intersection, phase is the number of the phase to be changed

at that intersection's traffic signal, and green time is a positive real number indicating the green time remaining or preprogrammed. To simplify the notation, the function is written hereafter as $T(\text{phase})$. It can both consult and assign the amount of the green time remaining in a given phase. In the latter case, if at the moment the function is called, the phase whose remaining green time is to be assigned is not the current one, and the change takes effect only once it is again activated. Thereafter, the phase returns to its predefined green time G_{phase} .

Algorithm 1 works as follows: The current phase f_A is defined as the phase encountered by the EMV as it approaches the stop line of an intersection it is seeking to cross. The phase closest to the current one in which the EMV is permitted to cross is identified. If the two are not the same, all phases between them are skipped by setting all the green times to zero and two seconds are assigned to the phase permitting the desired movement. If that is not sufficient for the EMV to cross, intervals of two additional seconds are assigned as needed. Once the EMV has cleared the intersection, Algorithm 2 is run to determine at what instant of the cycle the traffic signal should be returned to normal operation. This involves calculating how long the green time interval would have been had Algorithm 1 not intervened. To this end, the algorithm creates the variable Δ_G ("hypothetical green") to capture the hypothetical green time interval of the current phase f_A , while the EMV crosses the intersection without the right of way.

Note that, in Paramics, it is not possible to modify the all-red intervals (where no movement at all is permitted) between the phases that have to be skipped so that the phase allowing the EMV's desired movement can be activated. This limitation is circumvented by subtracting those intervals from Δ_G in line 12 of Algorithm 1 each time a phase is skipped, thereby offsetting the all-red "dead time" that the software inserts before a phase change.

If upon activation of Algorithm 2, $\Delta_G < 0$, the hypothetical green time terminated while the EMV was crossing the intersection. In such a case, the traffic signal is returned to normal by resetting it to the following phase $f_A + 1$ but with a green time less than the preprogrammed amount G_{f_A+1} .

The two algorithms aim to ensure that the API reproduces the real situation in the simulation software as closely as possible. In practice, the main difference between the simulation and the reality of a signalized intersection in the presence of an EMV is the all-red interval dead times that are lost by skipping to the f_{EMV} phase.

In the Chilean context, however, the all-red intervals range from zero to only two seconds. Furthermore, the traffic signals in Chile are programmed for two to four phases, the former number being the most common. So in almost every case, the dead time is insignificant.

Nevertheless, the dead time phenomenon may be quite realistic in a different sense; the CBS videos reveal that POV cooperation does not arise immediately upon the

```

(1) Calculate:
(2) number of phases  $N$ 
(3) current phase  $f_A$ 
(4) closest phase that permits the EMV's desired movement  $f_{EMV}$ 
(5) all-red interval between each phase  $f_i$  and following phase  $t_{RR}(f_i)$ 
(6) time remaining in the current phase  $T(f_A)$ 
(7) set  $\Delta_G = T(f_A) - 2$ ;
(8) set  $T(f_{EMV}) = 2$ ;
(9) if  $f_A \neq f_{EMV}$  then
(10)   for phases  $f_i$  between  $f_A$  and  $(f_{EMV} - 1)$ 
(11)     set  $T(f_i) = 0$ ;
(12)     set  $\Delta_G = \Delta_G - t_{RR}(f_i)$ ;
(13)   if current phase ==  $f_{EMV}$  and  $T(f_{EMV}) < 2$  then
(14)     set  $T(f_{EMV}) = T(f_{EMV}) + 2$ ;
(15)     set  $\Delta_G = \Delta_G - 2$ ;
(16) end

```

ALGORITHM 1: For signalized intersection with phases $1, 2, \dots, N$.

```

(1) Retrieve variables ( $N, f_A, f_{EMV}, \Delta_G$ ):
(2) Calculate all-red intervals between each phase  $f_i$  and the following phase  $t_{RR}(f_i)$ 
(3) if  $f_A \neq f_{EMV}$  then
(4)   for phases  $f_i$  between  $f_{EMV}$  and  $(f_A - 1)$ 
(5)     set  $T(f_i) = 0$ ;
(6)     set  $\Delta_G = \Delta_G - t_{RR}(f_i)$ ;
(7)   if  $\Delta_G \geq 0$  then
(8)     set  $T(f_A) = \Delta_G$ ;
(9)   else
(10)    set  $T(f_A) = G_{f_{A+1}} - |\Delta_G|$ ;
(11) end

```

ALGORITHM 2: For actuation of signalized intersections.

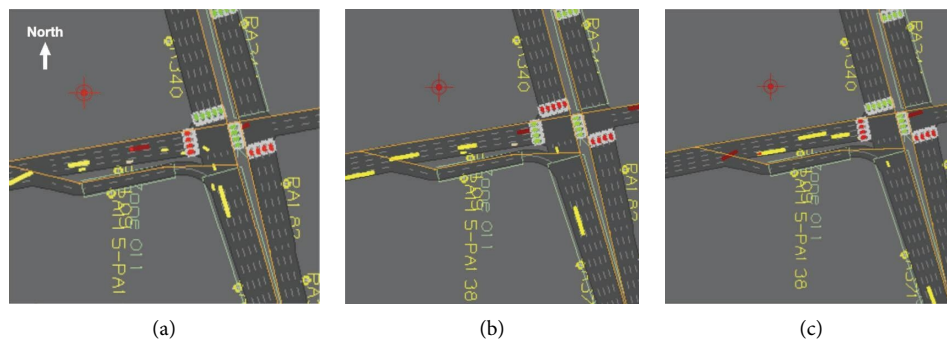


FIGURE 7: A sequence of snapshots in Paramics showing the traffic signal algorithms activated by the presence of an EMV.

arrival of an EMV at a signalized intersection. In other words, there is, in practice, a dead time where neither POVs nor the EMV attempts to cross the intersection, as it is not yet clear that all the POVs are cooperating with the EMV by giving way.

In Figure 7, we show a sequence of snapshots of the simulation in Paramics, corresponding to the activation of

the previous two algorithms in the API. The EMV painted in red in the figure is approaching the intersection from west to east, and as shown in Figure 7(a), the signal is red when the EMV arrives at the intersection; then, Algorithm 1 is activated to change the actuated signal to green during the required time interval for the EMV to cross the intersection (Figure 7(b)), returning to the phase

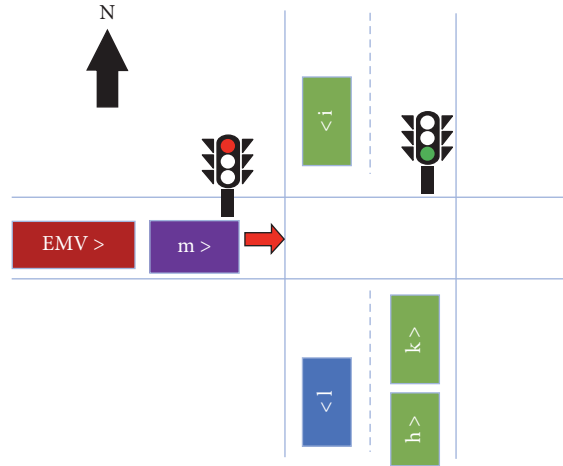


FIGURE 8: Schematic for the numerical example of the traffic signal algorithms.

TABLE 3: Traffic signal program for the example in Figure 8.

Phase	Time	Movements	Duration (s)
1	G_1	NS, SN	25
	Y_1	NS, SN	3
	RR_{12}	—	1
2	G_2	SN, SE	7
	Y_2	SN, SE	3
	RR_{23}	—	1
3	G_3	WE	20
	Y_3	WE	3
	RR_{31}	—	1

determined by Algorithm 2 after the EMV leaves the intersection (Figure 7(c)).

3.8. Numerical Example: Simulation of a Signalized Intersection. We now present a numerical example of a signalized intersection simulation based on the case depicted in Figure 8, in which an EMV approaches the intersection seeking to cross it travelling west-east (WE). The current phase of the traffic signal gives the right of way exclusively to traffic movements heading north-south (NS) and south-north (SN). Two additional traffic signal phases are detailed in Table 3.

In the simulation, the API carries out orders indicated by the EMV intersection model as soon as the EMV arrives within a distance δ_i of the intersection. Thus, POVs are on the cross-street stop (vehicles i , k , and h in the figure), while the POV downstream of the EMV (vehicle m) attempts to cross the intersection without having the right of way.

Simultaneously, the API activates Algorithm 1 to actuate traffic signals; otherwise, vehicle m and the EMV would have to wait for a green light to cross the intersection, while the vehicles on the cross street (except l) are stopped on red. The algorithm obtains the predetermined traffic signal program, as shown in Table 3, where G_i is the green time of the i th phase, Y_i is its yellow time, and RR_{ij} is the all-red interval that precedes the following phase. As shown in the last

column of the table, the traffic signal cycle is 64 seconds. Note that, for the API, the yellow time is the equivalent of an extension of the green time. In other words, for the API, only the green time and all-red times exist. Therefore, for the purposes of the algorithm, the green times in each phase are 28, 10, and 23 seconds, respectively.

Once Algorithm 1 is called, it calculates the number of phases, determines the current phase (phase 1 in this example), and identifies the closest phase that will permit the EMV's desired WE movement, denoted as the emergency phase. It first checks the phase following the current one, which in this case is phase 2. If the desired movement is not possible in that phase, it checks the next one, iterating this procedure as many times as necessary. In this example, the WE movement is possible in phase 3, so $f_{EMV} = f_3$. If we assume that at the moment the algorithm was activated, there were 10 seconds of the green time remaining in phase 1, and the variable $\Delta_G = T(f_A) - 2$ is created and $\Delta_G = 8$.

The algorithm then assigns green times of zero to all phases previous to f_{EMV} , including the current one. Thus, in phases 1 and 2 of the example, the green and yellow times are both set to zero. Additionally, for each phase so skipped, the algorithm subtracts the all-red time from Δ_G . Thus, $\Delta_G = 10 - RR_{12} - RR_{23} = 6$. The algorithm then waits until the traffic signal switches to f_{EMV} , whose green time was initialized at 2 seconds. If this is not sufficient time for the EMV

to clear the intersection, further 2 seconds are assigned and subtracted from Δ_G . If this were the case in the present example, Δ_G would be 4 seconds.

As soon as the EMV clears the intersection, Algorithm 2 is activated to return the traffic signal to the state it would have been in if Algorithm 1 had not intervened. It identifies this phase and the green time remaining in it. In practice, this means that identifying the phase the traffic signal was in when the EMV arrived at the intersection and the value of Δ_G . If $\Delta_G > 0$ as in this example, the green time is set to zero for all the phases between f_{EMV} and f_A (in the present case, 3, and 1, respectively). A green time of Δ_G is then assigned to the phase f_A , having already reduced Δ_G by the amount of all-red time between phases 3 and 1 since it, too, is the dead time. Finally, Δ_G is redefined to equal $4 - RR_{31} = 2$, thus returning the traffic signal to phase 1 with 2 seconds of the green time remaining.

4. Results in a Real Network

To achieve a realistic simulation of an actual EMV scenario, real data from various sources must be gathered so that the various parameters in the models can be accurately calibrated. With this in mind, we first checked the CBS logbook to find entries for fire truck emergency trips with GPS data records in June 2017. We then requested the audiovisual records of these trips to confirm that they were, indeed, emergencies.

Next, we used GeoPy package (<https://github.com/geopy/geopy>) together with a Google Map API to identify, in each case, the latitude and longitude of the emergency's precise location, in a process known as geocoding. An algorithm was employed to match the coordinates for each case with the last GPS pulses for each trip in June 2017 from the GPS database. This process enabled us to retrieve those trips whose destination was close to the emergency site, where "close" was defined as within 500 metres and a time difference of less than two hours. Of these trips, we retained the ones that were made mainly in central Santiago on a weekday to attend an emergency in that same zone.

Four of the trips were found to have been recorded on video, and these were viewed to determine which would be of use for our purposes. The required characteristics were that (i) the recording was of an EMV's trip to the emergency, not (just) its return to the fire station; (ii) the recording captured the EMV's interactions with surrounding vehicles; and (iii) the EMV's trip was sufficiently long to include situations of interest. On this basis, it was concluded that only the material recorded by ladder truck no. Q8 of the fire service's 8th Company would be useful. The recorded material consisted of four videos, each taken from a different point on the truck: front, rear, driver's side (left), and co-driver's side (right). The emergency event selected for analysis occurred at the corner of Bellavista and Capellán Abarzúa streets in Santiago's Bellavista district at coordinates WGS84 (-33.4319 , -70.6291), on June 19, 2017. The trip to the event covered a distance of 2,400 metres over 4 minutes and 10 seconds. The GPS pulses, recorded every 10 seconds, are plotted on a map in Figure 9.

Upon identifying this trip as the most appropriate, the network consisting of the route the truck followed and the cross streets at each intersection along the route was coded into Paramics. In addition, we coded in the microsimulator the traffic signals, with their predetermined programs, and the geometric characteristics of the streets. The complete network is shown in Figure 10, where the numbers in yellow correspond to the codes of the nodes in Paramics, the left blue arrow indicates the beginning of the trip at the fire station, and the right blue arrow indicates its end at the emergency site. The red circles identify the intersections of Santa María Avenue with Recoleta Street and Pio Nono Street, which are discussed below.

In the settings of the microsimulator, ten different types of vehicles were defined with their own features, as shown in Table 4. There are differences in top and conformable (crawl) speed, horsepower, length, familiarity, and also the percentage of each type vehicle that is loaded in the network according to the origin-destination matrix coded in Paramics (see details of the specific matrix used in this study in Subsection 4.4 later in this section). In fact, the sum of the percentages (except the special case of EMV and fixed route vehicles 15 and 16) is equal to 100 as expected. Note that the EMV vehicle type has a 100% share of the origin-destination matrix because a separate matrix is coded in Paramics to define the EMV's origin, destination, and the moment it appears in the simulated network.

In the previous table, LGV stands for light goods vehicle, which is a commercial carrier vehicle with a gross weight of no more than 3.5 tonnes. Notice that we have created a special classification for our EMVs (vehicle type 10) with technical features similar to an average fire truck used by the CBS. In addition, we investigate the acceleration and deceleration rates in $[m/s^2]$ used internally by the microsimulation software by plotting both the acceleration and the deceleration profiles with respect to speed, obtaining decreasing behaviour and values that are quite consistent with those reported in the specialized literature (for example, Tables 2 and 3 and Figures 6 and 7 from Bokare and Maurya [30]).

4.1. Estimation of the Traffic Microsimulator Parameters.

Once the network was coded, the intersections along the truck's route that had traffic cameras were identified and three 30-minute videos were obtained. With a view to emulating a period of demand similar to the one experienced on the EMV's trip, the footage chosen was recorded on different weekdays at the same time of day. One of the videos was taken by a camera fixed upon a single approach to the intersection of Santa María and Pio Nono, while the other two were taken by a rotating camera recording different approaches to the intersection of Santa María and Recoleta (see Table 5). In the latter case, the camera in the first video (27 December 2017) pointed east for 14 minutes and then rotated to point south for 5 minutes before finally aiming north for the remaining 11 minutes. In the other video of the intersection (3 January 2018), the camera began by pointing north for 3 minutes, then turned east for 7 minutes, and

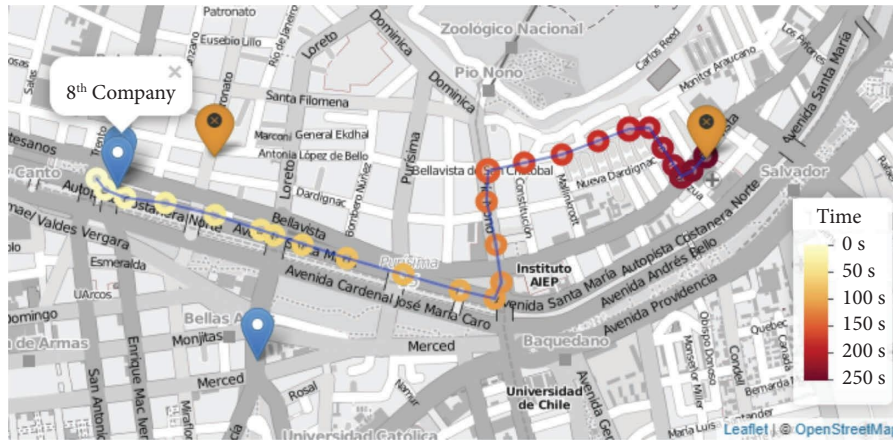


FIGURE 9: Truck Q8 trip route: 3:18 pm, Monday, June 19, 2017.

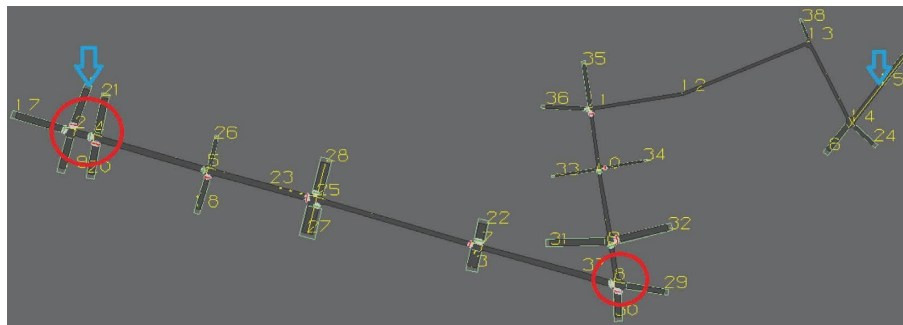


FIGURE 10: Truck Q8 trip route as coded in Paramics.

TABLE 4: Description of vehicles' features coded in Paramics.

Car type	Description	Top speed (kph)	Crawl speed (kph)	Horsepower (HP)	Length (m)	Matrix (%)	Familiarity (%)
1	Car	160	80.5	130	5	70	85
2	Van	160	80.5	250	5	6	70
3	Taxi	160	80.5	130	5	20.5	90
10	EMV	100	50	300	11.5	100	100
11	LGV	130	64.4	220	5	1.5	85
13	Truck	120	32.2	400	14	0.5	85
14	Coach	110	48.3	350	12	1.5	85
15	Minibus	100	64.4	200	8	Fixed route	100
16	Bus	100	48.3	220	12	Fixed route	100

TABLE 5: Intersection traffic camera video footage.

Intersection	Date	Start time	End time	Duration
Pio Nono-Santa María	24 January 2018	14:59:56	15:29:59	0:30:03
Santa María-Recoleta	27 December 2017	14:59:58	15:29:59	0:30:01
Santa María-Recoleta	3 January 2018	14:59:59	15:29:59	0:30:00

concluded by pointing south for the remaining 20 minutes. Images from the footage of the two intersections taken at various angles are shown in Figure 11.

4.2. *Estimated Mean Reaction Time and Mean Target Headway.* The videos were used to estimate the mean

reaction time (MRT) and mean target headway (MTH), the two most important parameters for calibrating the micro-simulation software [26, 27]. The MRT was measured by observing the dissipation of the queue at the intersection following the changing of the traffic signal for the approach targeted by the camera (see Figure 11(a)) from red to green.



FIGURE 11: Traffic camera views: (a) Pio Nono bridge approach to Santa María looking south, (b) Pio Nono bridge approach to Santa María looking north-east, (c) Recoleta at Santa María looking south, and (d) Recoleta at Santa María looking north.

The parameter value was identified as the time elapsed between the moment the signal changed and the moment the first vehicle in the queue began to move. If there was a vehicle behind the first one that did not start to move at the same time, the moment at which it did move was also determined. This step was repeated for the third and fourth vehicles. For vehicles further back in the queue, camera resolution was normally not sufficient to allow accurate measurement.

As regards the MTH, the procedure was as follows: The videos were examined to find pairs of POVs in ideal conditions, defined as both vehicles in the same lane, both travelling at the same constant speed, and neither immediately seeking to make a turn. Furthermore, they had to be relatively closely spaced so that the one behind would be influenced by the one in front. Finally, the camera angle had to be such that these conditions could be clearly confirmed. If all these criteria held, the time spacing for the pair was measured. This was performed by pausing the video, marking a reference point on the roadway, then advancing the video in discrete steps, and noting the instant at which the reference point was reached by each vehicle's front bumper. The time spacing thus obtained was considered to be a reliable estimate of the target headway desired by the vehicle behind.

The results obtained for the MRT are set out in Table 6. Note that it was not possible to derive an estimate of this parameter for every vehicle in the videos. In some cases, for example, the moment at which a vehicle started to move with a change to green could not be clearly observed, typically because the camera angle was blocked by a motorcycle,

bicycle, or pedestrian. If a vehicle not first in the queue started to move before the vehicle in front of it, the reaction could not be associated with the light change. Cases where the change to green could not be observed or where the camera view was on the back of the vehicle rather than the front also had to be rejected. Thus, no estimates could be derived from the two Santa María-Recoleta videos. In the end, 104 useable vehicle reactions were identified, all from the Pio Nono-Santa María video, and 40 of them were cases where the vehicle was first in the queue.

The MTH results for the two Santa María-Recoleta videos are collected in Table 7. To estimate time spacing or target headways, views of vehicle profiles moving in situations of low congestion rather than queues in the foreground (recorded from the front) had to be found. However, low congestion and a good angle for observing the front bumper of each vehicle were not sufficient; the situation itself had to be stable in the sense that vehicles were moving at the same constant speed; that is, headways were neither increasing nor decreasing. Furthermore, vehicles could not be braking due to congestion visible further ahead. Moreover, these conditions had to hold for any pair of vehicles after they had passed the reference point. In the case of the Pio Nono-Santa María intersection, there were few cases where these criteria were satisfied given that once the queues dissipated from the bridge, the few vehicles that were still visible were all either relatively distant from each other or in different lanes.

Box plots for the MTH and MRT estimates are shown in Figure 12. In each case, the 75th and 25th percentiles are indicated by the upper and lower bounds of each box; the

TABLE 6: MRT results for Pio Nono.

Intersection	Queue position	Data items	MRT (s)	Std dev (s)
Pio Nono-Santa María	First	40	1.49	0.77
Pio Nono-Santa María	Other	64	1.34	0.66
Total	—	104	1.4	0.71

TABLE 7: MTH results for Recoleta.

Intersection	Date	Data items	MTH (s)	Std dev (s)
Santa María-Recoleta	27/12/17	20	1.43	0.24
Santa María-Recoleta	03/01/18	18	1.45	0.25
Total	—	38	1.44	0.24

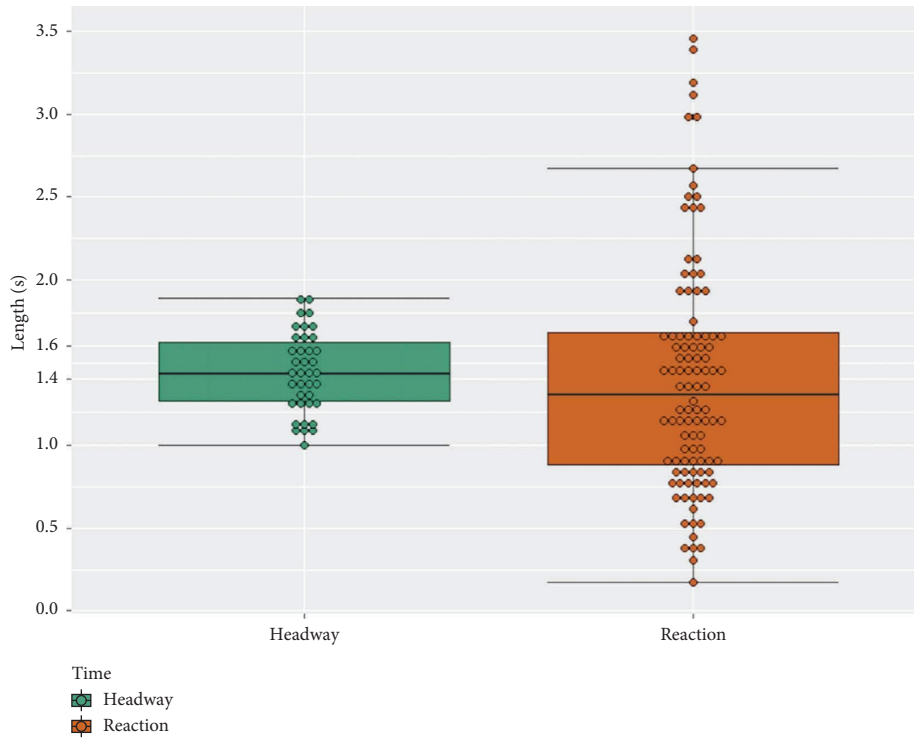


FIGURE 12: Box plots for headways and reaction times.

horizontal line in each box marks the median or 50th percentile. The data are graphed as points ordered in intervals of 0.1 seconds. The plots reveal that the reaction times display greater dispersion than do headways.

4.3. *API Parameter Estimation.* Once the MRT and MTH estimates (1.40 and 1.44 seconds, respectively) were derived, the relevant API EMV model parameters were calibrated. Since no vehicles in the videos were observed pulling up onto the sidewalk or failing to cooperate in the presence of fire trucks (i.e., the cooperation factor was $CF = 100\%$), the only parameters that were estimated were α_a , β_a , and δ_i (the last one was assumed to equal d_v^{\min}).

The estimates were derived from the interactions between the EMV and the POVs observed in the videos that were recorded by the fire truck-mounted cameras from the four angles noted in the previous section during the truck’s four-minute trip to the emergency site. The events identified for each parameter are detailed in Table 8; images for two of the events are shown in Figures 13 and 14.

The procedure for accurately estimating the reaction distances involved measuring the lane widths and lengths and spacing of the broken lane-separation lines. This was performed for the streets travelled by the fire truck (Santa Maria and Pio Nono), where POV reactions were identified. The measurements were the same for both streets, as indicated in the diagram in Figure 15. The resulting distances

TABLE 8: Events identified in fire truck Q8 videos en route to emergency sites.

Angle	Time	POV reaction	Distance	Value (m)	Rel. parameter
Front	15:16:54	Changed lanes	In front of EMV	10	β_a
Codriver	15:17:11	Stopped although light green	To intersection (POV, EMV)	(5, 5)	δ_i
Rear	15:17:15	Continued normally	Behind EMV	15	α_a
Front	15:17:27	Changed lanes	In front of EMV	27	β_a
Front	15:17:44	Changed lanes	In front of EMV	40	β_a
Front	15:17:45	Changed lanes	In front of EMV	30	β_a
Codriver	15:17:46	Stopped although light green	To intersection (POV, EMV)	(10, 10)	δ_i
Front	15:18:24	Changed lanes	In front of EMV	30	β_a
Front	15:18:40	Changed lanes	In front of EMV	30	β_a
Codriver	15:18:42	Stopped although light green	To intersection (POV, EMV)	(10, 10)	δ_i
Front	15:18:44	Crossed although light red	In front of EMV	10	δ_i
Codriver	15:18:47	Changed lanes	In front of EMV	40	β_a
Rear	15:18:46	Returned to original lane	Behind EMV	15	α_a
Front	15:19:23	Stopped although light green	To intersection (POV, EMV)	(10, 20)	δ_i
Front	15:20:34	Crossed w/out the right of way	In front of EMV	15	δ_i



FIGURE 13: Vehicles stopped at green light.



FIGURE 14: Vehicle crossing intersections on a red light.

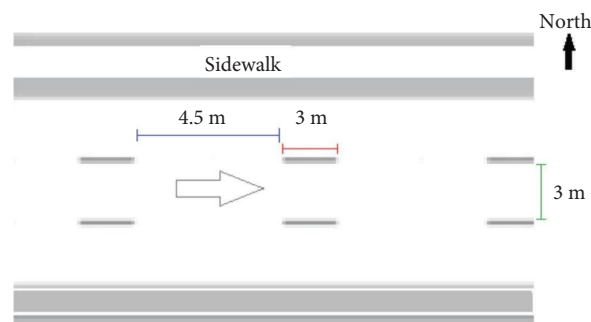


FIGURE 15: Santa María street measurements.

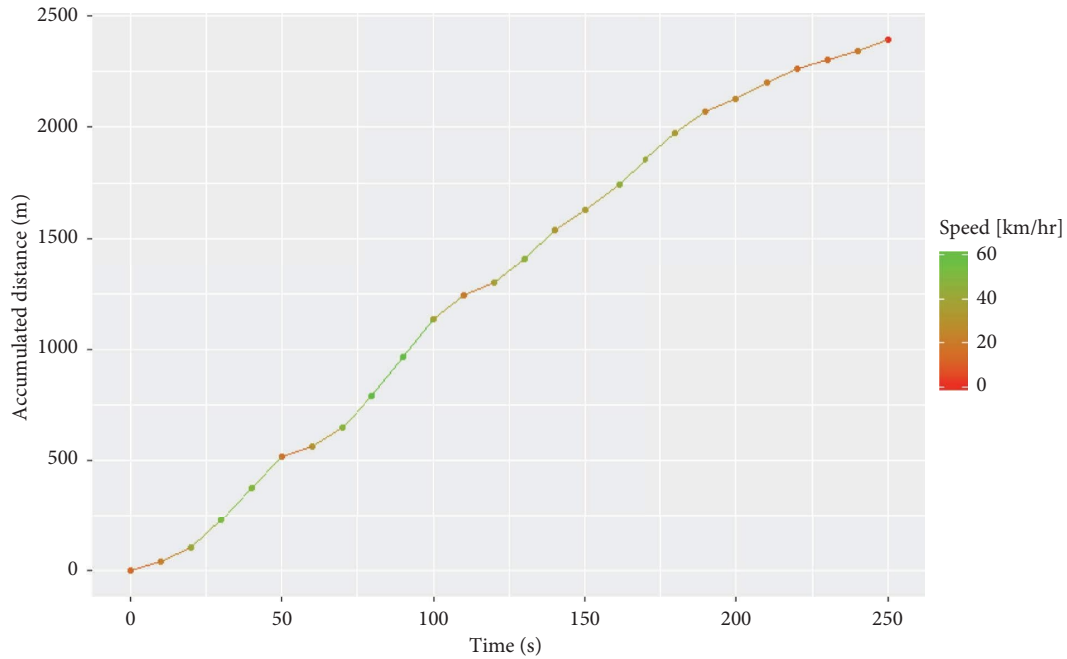


FIGURE 16: Accumulated distance vs. time profile from real truck GPS pulses.

for each type of reaction and the related parameter are presented in Table 8. The averages for the three parameters were $\alpha_a = 15m$, $\beta_a = 30m$, and $\delta_i = 10$.

4.4. Real-Network Calibration. Having coded the network and estimated the parameters using real data, the next step was to calibrate the network flow. The flow or demand along streets impacts EMVs via the effects on congestion, slowing progress and thereby influencing the distance covered versus time profile. The profile for the real truck was derived from its GPS pulses (see Figure 16) and then compared to the profiles constructed from simulated GPS pulses, that is, distance measurements taken every 10 seconds in each simulator run, to determine the simulations' goodness of fit.

The simulated profiles were generated for three special scenarios with no demand: one without the API, another without traffic signals, and a third with both the API and traffic signals. The truck was always created at the fifth minute of the simulation, and the simulated GPS pulses were counted starting from the same point along the route as the first GPS pulse from the real truck. The simulated profiles together with the real one are shown in Figure 17. Arrival at the emergency site is indicated by the flattening out of the curves. The top profile represents the simulation without traffic signals, allowing the truck to arrive at the emergency site more rapidly than in other simulations. The simulation at the other extreme is the one without the API but with traffic signals, in which the truck stopped at red lights and arrived at the site at some point beyond 250 seconds, not visible off the right edge of the graph.

Once the special cases were simulated, the traffic network was then loaded to obtain traffic queue lengths that were visually similar to those observed in the truck-mounted camera footage. This describes the base scenario used for our

analyses. In order to replicate queue lengths, and also to model flows that are similar to those obtained by counting the vehicles observed in the videos recorded by the cameras mounted on the fire truck, we construct the trip demand or origin-destination matrix shown in Table 9, in $[veh/15mins]$, based on the origin-destination zones defined in Paramics, as shown in Figure 18.

The numbers in the matrix (Table 9) as well as the proportion of vehicles in each category (Table 4) were properly chosen to generate a base scenario behaving similarly to the real scenario recorded by the cameras, on both the fire trucks and also the fixed cameras installed at key intersections along the EMV path. We observe two numbers in the table that are significantly larger than the rest of the numbers in the table. These correspond to the following pairs: (16 – 15) and (2 – 21), both corresponding to flows crossing the EMV path: one intersection located in the middle of the path, while the other very close to the emergency site. In these cases, the queue lengths observed in the accesses to those intersections were significant. Therefore, in order to replicate those queues in the model, it was necessary to intensify the flows on those origin-destination pairs. This detail could become relevant at the moment of triggering the traffic light model in which EMVs could eventually cross the junction in red.

With the latter as the base scenario, the total network load was varied to obtain 8 additional simulated scenarios. The variations were generated by dividing the base trip demand matrix and trip structure by demand factors ranging between 0.5 and 2. The precise factor values employed were 0.50, 0.57, 0.66, 0.80, 1.25, 1.50, 1.75, and 2.00, the latter four being just the reciprocals of the first four to simplify the process of manually changing the scenario in Paramics.

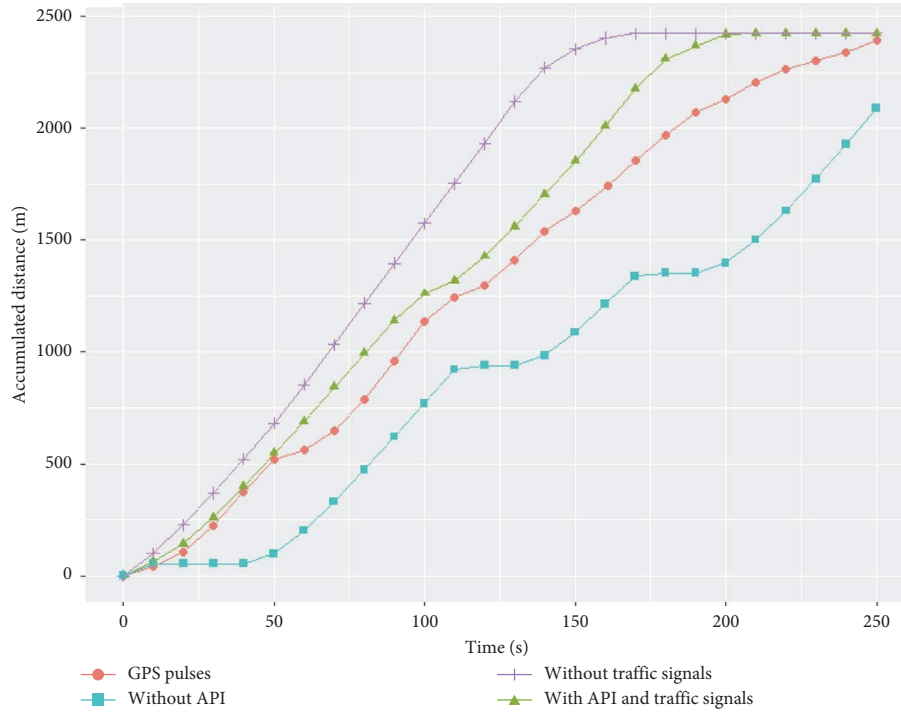


FIGURE 17: Accumulated distance vs. time profiles from real and special simulation GPS pulses.

TABLE 9: Origin-destination matrix built for the base scenario [veh/15mins].

OVD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1				8		4																	
2																							215
3						8		10	6	8	8			8	8								
4																							
5						35		50	10	4	4			4	4								
6																							
7								40			4			6	6								
8																							
9										20	4			20	4								
10									30		4			30	6								
11											10												
12												10		35	6								
13														8	15							8	
14																							
15																							
16															150								4
17																							
18																		8					
19																							50
20																							
21																							
22																							
23																							

Each of the simulated scenarios was run 200 times, varying the seed value used to generate the pseudorandom numbers used by using the microsimulation software. The accumulated distance profile for the base scenario is shown in Figure 19: the black curves represent each of the scenario's

200 runs and the blue curve represents their average. The average RMSP (root mean square percentage error) for the differences between the 200 scenario runs and the real GPS profile was calculated for each of the 9 scenarios and the three special no-demand simulations using the formula in

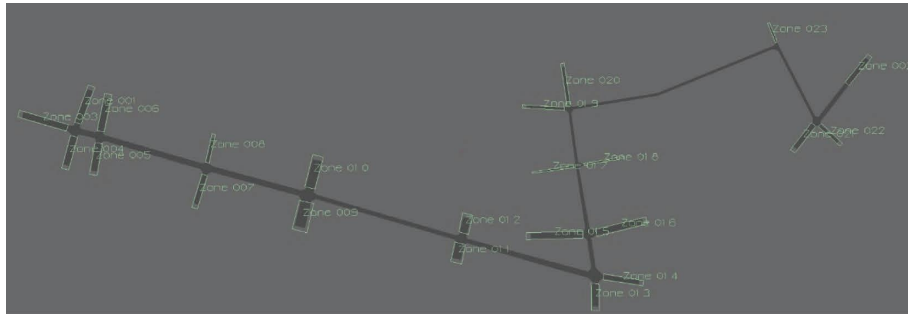


FIGURE 18: Network zones coded in Paramics.

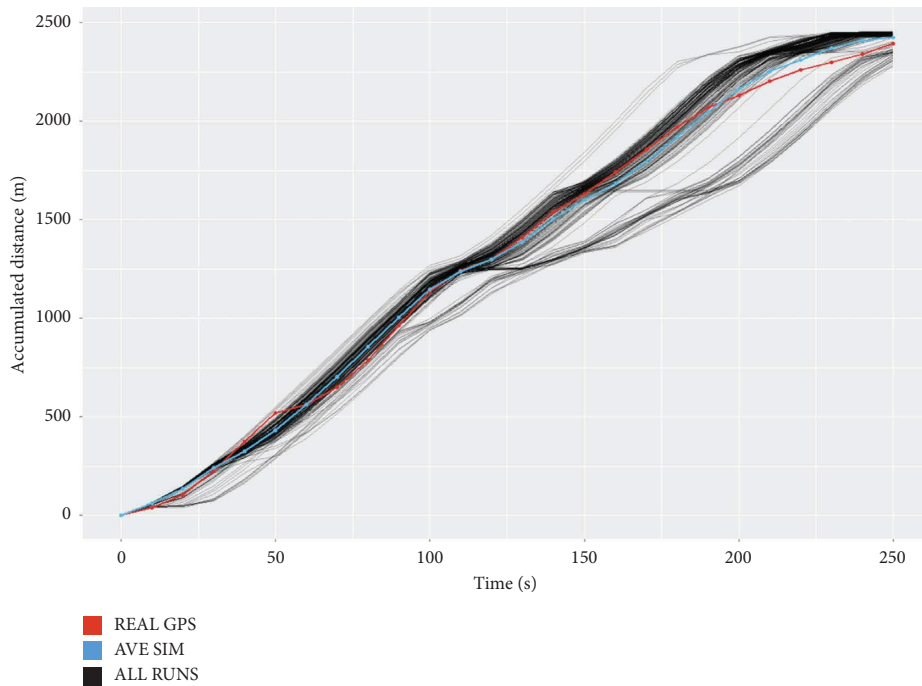


FIGURE 19: Accumulated distance vs. time profile from real GPS pulses and base scenario simulations (average and each run).

TABLE 10: RMSP and correlation coefficients for simulated scenarios.

Scenario	Average RMSP	σ_{RMSP}	Correlation coeff
No demand w/traffic signals and API	0.1901	—	0.9953
No demand w/out traffic signals	0.5324	—	0.9659
No demand w/out API	0.4320	—	0.9827
Base scenario w/factor of 0.50	0.1851	0.0462	0.9951
Base scenario w/factor of 0.57	0.1800	0.0552	0.9972
Base scenario w/factor of 0.66	0.1715	0.0392	0.9979
Base scenario w/factor of 0.80	0.1651	0.0373	0.9981
Base scenario w/out factor	0.1635	0.0311	0.9983
Base scenario w/factor of 1.25	0.1630	0.0369	0.9988
Base scenario w/factor of 1.50	0.1676	0.0291	0.9989
Base scenario w/factor of 1.75	0.1729	0.0261	0.9988
Base scenario w/factor of 2.00	0.1711	0.0271	0.9988

equation (5). The mean and standard deviation (σ_{RMSP}) of the RMSP were then computed to determine which scenario exhibited the least error. The correlation coefficient r of the

average scenario profile was also calculated via equation (6). For calculation purposes, the first simulated pulse was excluded given that it was set at the same level as the first real

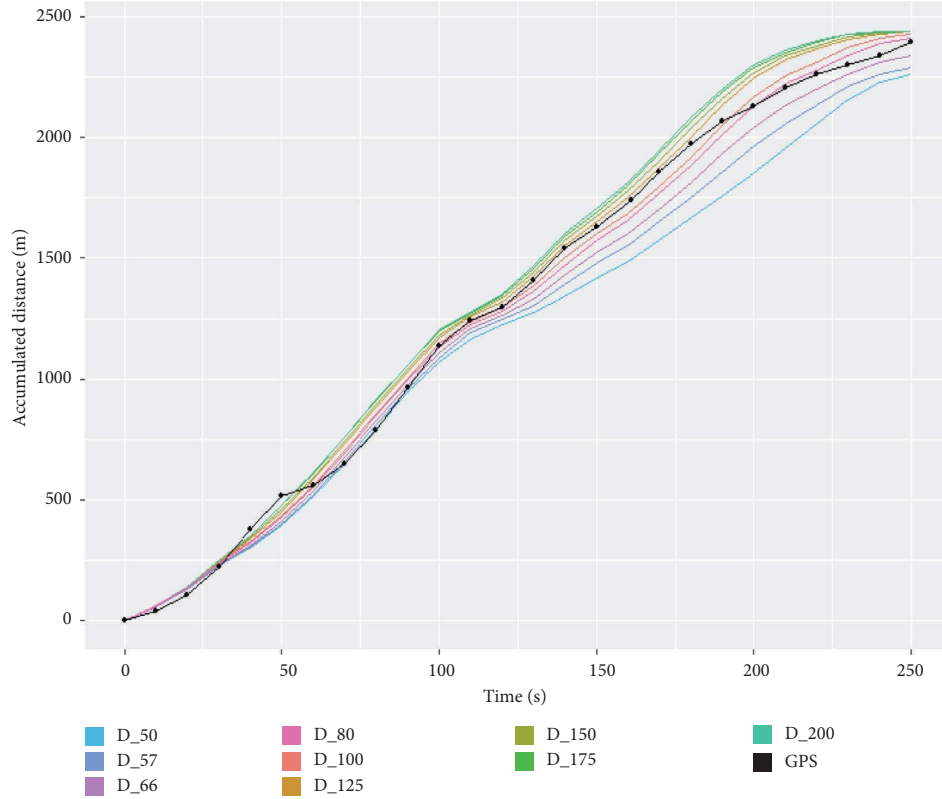


FIGURE 20: Accumulated distance profile for real GPS pulses and all simulated scenarios.

GPS pulse. The results of all three sets of computations are presented in Table 10.

$$RMSP = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{x_i - y_i}{y_i} \right)^2}, \quad (5)$$

$$r = \frac{1}{N-1} \sum_{i=1}^N \frac{(x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y}. \quad (6)$$

The correlation coefficient quantifies the correlation between two data series, with a value of 1 indicating a perfect positive correlation. Statistically, this would mean that there was a linear relationship between the two such that if one of the series increases, so would the other one. The RMSP indicator, on the other hand, quantifies the mean square error in percentage terms. The quadratic aspect implies that small errors are proportionally less influential on the indicator's mean value than large ones. Expressing the value as a percentage normalizes the errors for a given variable where the magnitude of the latter varies at different moments in the sample.

The results show that all the scenarios had a strong positive linear correlation with the real data given that the correlation coefficient was very close to 1 in every case. The scenario generated with a demand factor of 1.25 had the lowest RMSP, and therefore, the best goodness of fit was obtained. Thus, the scenario that the best fit reality

was the one with 80% of base scenario demand. However, the order of magnitude of the standard deviation of all the average RMSP values for the different scenarios was at the second decimal place. The same was true for the differences between their standard deviations. Furthermore, the scenario with no demand, with the API and traffic signals, had an RMSP value slightly higher than that of the best demand scenario, which may indicate that congestion is not a major factor for improving the goodness of fit. However, the RMSP improved considerably when the API and traffic signals were incorporated into the simulations, declining from 43.2% to 19.0%. In addition, although the RMSP value rose as the scenario diverged from the one with the lowest RMSP, it fell again in the least congested scenario $D_{2.00}$. The literature reports that this phenomenon may lead to calibration problems depending on the choice of the goodness-of-fit indicator and the parameters that are being calibrated, in the sense that there may be local minima and variable behaviour [31].

To get a better idea of the average goodness of fit for each of the scenarios, their average profile curves (i.e., the distance accumulated every 10 seconds averaged over the 200 runs) are plotted in Figure 20. The curves for the 200 simulations of the lowest-RMSP scenario ($D_{1.25}$) and the corresponding average curve (in black and blue, respectively) are exhibited together with the profile for the real GPS pulses in Figure 21.

The ultimate result of this calibration process is a simulation of an EMV trip in Santiago with an average error of only 16.3%. This suggests that the proposed API is a useful

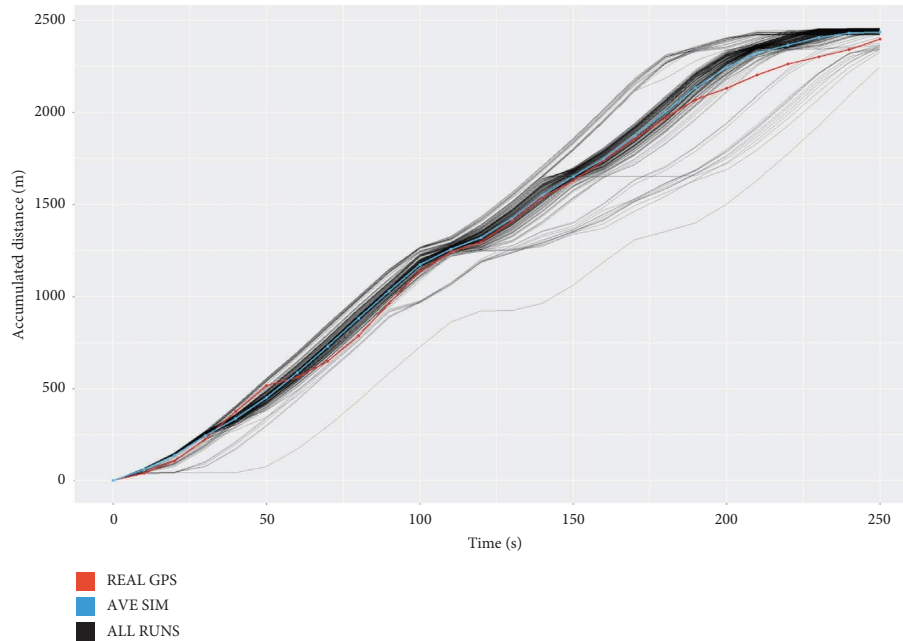


FIGURE 21: Accumulated distance profile for real GPS pulses and simulated scenario $D_{1,25}$ (average and each run).

tool for simulating EMVs. Since the process involved estimating the model parameters for intersections and arcs and was conducted for a specific urban zone and time of day, future EMV simulations for the same zone and time of day should use the parameters obtained here. The MTH and MRT parameter estimates could also be used for simulations of traffic on weekdays at off-peak hours in central Santiago if no more recent data for the zone and time of day are available.

5. Conclusions and Further Research

In this work, we present a collection of behavioural models that were developed to permit the simulation of emergency vehicles in normal traffic conditions with all the effects typically observed both in the literature and field. The models were based on data derived from audiovisual material on fire trucks recorded en route to emergency situations in Santiago, Chile. The models were incorporated in commercial microsimulation software (Paramics) through the implementation of an API, which was calibrated and used to simulate one of the truck's real emergency trips.

Thanks to the extensive library of API functions available in the chosen microsimulation platform, the proposed API was able to reproduce almost the entire range of behaviours of non-emergency vehicles in the presence of fire trucks observed in the Santiago fire service videos, as well as those reported in the literature. The API models included one for the interactions between non-EMVs and EMVs on single-lane streets, another for cases in which non-EMVs are able to pull up onto the sidewalk to let an EMV pass, and a third that describes vehicle behaviour at intersections. Also, two ad hoc algorithms were incorporated in the API that were conceived to circumvent limitations imposed by the software on

signalized intersections so that the real vehicle dynamics observed at these intersections could be reproduced.

As regards the simulation of the real trip, the specific parameters representing the reaction time and drivers' target headways were estimated, using traffic camera video footage recorded in the same urban zone and time of day, to be 1.4 and 1.44 seconds, respectively. The API parameters were derived from the videos captured by truck-mounted cameras, yielding estimates for the average distance at which a non-EMV perceived the presence of an EMV of 30 metres in general and 10 metres on the approaches to an intersection. The average distance at which a non-EMV's state of alert terminated once an EMV passed was estimated to average 15 metres. These estimates should be useful for other studies where real data or reliable estimates for the specific case under investigation are not available.

A simulation of the traffic network in which the real trip was made under typical congestion conditions was calibrated to determine the goodness of fit. Before loading the network with vehicle flows, however, it was found that using the API developed for this study reduced the error from 43% to 19%. This result demonstrated that the API met the objective of improving simulations by successfully incorporating real-observed behaviour. Upon calibrating the network, the error was reduced to 16.3%.

In conclusion, the proposed API tool proved to be able to microsimulate and evaluate the operation of emergency vehicles in an urban environment. In addition, a calibration framework was designed to estimate various parameters required to conduct the simulations in such a manner that they represent EMV and non-EMV behaviour in the real world as faithfully as possible. We are currently using the tool to model the entire downtown of Santiago to obtain better estimates of the real travel times of EMVs in that area of the city to improve the shortest-path predictions provided

by the dispatch system of CBS to reach emergency locations faster with available trucks.

We would like to emphasize that the API was coded in Paramics because the software has shown flexibility in the use of APIs in the past, in different and diverse applications, from detailed modeling of the interactions between passengers and transit vehicles in the context of public transport applications [32, 33] to modeling a baggage handling system at an international airport [34]. Nevertheless, the methodology is generic, and the adaptation of the APIs to another platform will depend on the capabilities of the software to manage the different models based on its own API's capabilities. We have explored the TraCI interface in open-source SUMO to create a module in the simulator to directly incorporate the functionalities of the plugins developed in Paramics. A task for future studies is to use the API for networks in other cities, calibrating them with real local data so that the tool can be used to generate better predictions or estimates of EMV travel times.

Data Availability

The simulated data, videos, and simulation API codes used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are grateful for the financial support provided by the projects ANID/FONDECYT/Regular 1191200 and Instituto Sistemas Complejos de Ingeniería (ANID PIA/PUENTE AFB220003).

References

- [1] A. Echeverría, *Evaluación De Tiempos De Respuesta Para El Cuerpo De Bomberos De Santiago: Nuevo Modelo De Ruteo, Validación Y Métricas De Desempeño Del Sistema Actual*, Universidad de Chile, Santiago, Chile, USA, 2013.
- [2] P. Kachroo and K. Ozbay, *Feedback Ramp Metering in Intelligent Transportation Systems*, Springer, Berlin, Germany, 2003.
- [3] J. Barceló, *Fundamentals Of Traffic Simulation*, vol. 145, Springer, Berlin, Germany, 2010.
- [4] A. Buchenscheit, F. Schaub, F. Kargl, and M. Weber, "A vanet-based emergency vehicle warning system," in *Proceedings of the 2009 IEEE Vehicular Networking Conference (VNC)*, pp. 1–8, IEEE, Tokyo, Japan, October 2009.
- [5] F. Weinert and M. Düring, "Development and assessment of cooperative V2X applications for emergency vehicles in an urban environment enabled by behavioral models," in *Modeling Mobility with Open Data*, pp. 125–153, Springer, Berlin, Germany, 2015.
- [6] F. Weinert, M. Düring, and K. Bogenberger, "Influence of emergency vehicle preemption on travelling time and traffic safety in urban environments enabled by innovative behavioral models and V2X communication - simulation and case study," in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 4043–4048, Auckland, New Zealand, October 2019.
- [7] L. Bieker-Walz, "Cooperative traffic management for emergency vehicles in the city of Bologna," *Proceedings of the SUMO User Conference 2017*, vol. 31, pp. 135–141, 2017.
- [8] Q. Lu and K.-D. Kim, "A genetic algorithm approach for expedited crossing of emergency vehicles in connected and autonomous intersection traffic," *Journal of Advanced Transportation*, vol. 2017, Article ID 7318917, 14 pages, 2017.
- [9] L. Bieker-Walz, M. Behrisch, and M. Junghans, "Analysis of the traffic behavior of emergency vehicles in a microscopic traffic simulation," in *SUMO 2018- Simulating Autonomous And Intermodal Transport Systems, Volume 2 of EPiC Series In Engineering*, E. Wießner, L. Lücken, R. Hilbrich et al., Eds., EasyChair, Wythenshawe, Manchester, United Kingdom, pp. 1–13, 2018.
- [10] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, and S. Fischer, "Traci: an interface for coupling road traffic and network simulators," in *Proceedings of the 11th Communications and Networking Simulation Symposium*, pp. 155–163, ACM, Ottawa, Canada, April 2008.
- [11] L. Zhang, J. Gou, K. Liu, G. McHale, R. Ghaman, and L. Li, "Simulation modeling and application with emergency vehicle presence in CORSIM," in *Proceedings of the Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th*, pp. 1–7, IEEE, Anchorage, AK, USA, September 2009.
- [12] G. J. Hannoun, P. Murray-Tuite, K. Heaslip, and T. Chantem, "Meso- and microscopic routing of an emergency response vehicle with connected vehicle technologies," *International Journal of Emerging Technology and Advanced Engineering*, vol. 10, no. 4, 2020.
- [13] J. B. Yoo, J. Kim, and C. Y. Park, "Road reservation for fast and safe emergency vehicle response using ubiquitous sensor network," in *Proceedings of the Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on*, pp. 353–358, IEEE, Newport Beach, CA, USA, June 2010.
- [14] C. Toy, K. Leung, L. Alvarez, and R. Horowitz, "Emergency vehicle maneuvers and control laws for automated highway systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 2, pp. 109–119, 2002.
- [15] J. J. So, J. Kang, S. Park, I. Park, and J. Lee, "Automated emergency vehicle control strategy based on automated driving controls," *Journal of Advanced Transportation*, vol. 2020, Article ID 3867921, 11 pages, 2020.
- [16] S. Panwai and H. Dia, "Comparative evaluation of microscopic car-following behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 3, pp. 314–325, 2005.
- [17] L. Evans and P. Wasieleski, "Risky driving related to driver and vehicle characteristics," *Accident Analysis and Prevention*, vol. 15, no. 2, pp. 121–136, 1983.
- [18] T. A. Ranney, "Psychological factors that influence car-following and car-following model development," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 2, no. 4, pp. 213–219, 1999.
- [19] R. A. De Lorenzo and M. A. Eilers, "Lights and siren: a review of emergency vehicle warning systems," *Annals of Emergency Medicine*, vol. 20, no. 12, pp. 1331–1335, 1991.
- [20] M. M. Haque and S. Washington, "A parametric duration model of the reaction times of drivers distracted by mobile phone conversations," *Accident Analysis and Prevention*, vol. 62, pp. 42–53, 2014.

- [21] G. Kotusevski and K. Hawick, "A review of traffic simulation software," 2009, https://www.researchgate.net/publication/228966705_A_Review_of_Traffic_Simulation_Software/link/00b4951f49b6359826000000/download.
- [22] 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, 1999.
- [23] M. Saifuzzaman and Z. Zheng, "Incorporating human-factors in car-following models: a review of recent developments and research needs," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 379–403, 2014.
- [24] G. Duncan, "Paramics technical report: car-following, lane-changing and junction modelling," *Quadstone, Ltd*, vol. 5, no. 1, pp. 5–2, Edinburgh, Scotland, 1997.
- [25] H. T. Fritzsche, "A model for traffic simulation," *Traffic Engineering and Control*, vol. 35, pp. 317–321, 1994.
- [26] I. Reza, N. T. Ratrou, and S. M. Rahman, "Calibration protocol for paramics microscopic traffic simulation model: application of neuro-fuzzy approach," *Canadian Journal of Civil Engineering*, vol. 43, no. 4, pp. 361–368, 2016.
- [27] N. T. Ratrou, S. M. Rahman, and I. Reza, "Calibration of Paramics model: application of artificial intelligence-based approach," *Arabian Journal for Science and Engineering*, vol. 40, no. 12, pp. 3459–3468, 2015.
- [28] R. Zhong, K. Fu, A. Sumalee, D. Ngoduy, and W. Lam, "A cross-entropy method and probabilistic sensitivity analysis framework for calibrating microscopic traffic models," *Transportation Research Part C: Emerging Technologies*, vol. 63, pp. 147–169, 2016.
- [29] T. Y. Zhi, C. R. Long, and L. Der Horng, "Comparison of paramics and getram/aimsun for its simulations," *Institution of Engineers, Singapore*, vol. 45, 2005.
- [30] P. Bokare and A. Maurya, "Acceleration-deceleration behaviour of various vehicle types," *Transportation Research Procedia*, vol. 25, pp. 4733–4749, 2017.
- [31] B. Ciuffo and V. Punzo, "No free lunch theorems applied to the calibration of traffic simulation models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 553–562, 2014.
- [32] R. Fernández, C. E. Cortés, and V. Burgos, "Microscopic simulation of transit operations: policy studies with the MISTRANSIT application programming interface," *Transportation Planning and Technology*, vol. 33, no. 2, pp. 157–176, 2010.
- [33] C. E. Cortés, V. Burgos, and R. Fernández, "Modelling passengers, buses and stops in traffic microsimulation. review and extensions," *Journal of Advanced Transportation*, vol. 44, no. 2, pp. 72–88, 2010.
- [34] J. P. Cavada, C. E. Cortés, and P. A. Rey, "A simulation approach to modelling baggage handling systems at an international airport," *Simulation Modelling Practice and Theory*, vol. 75, pp. 146–164, 2017.