

## Controller Independent Software-in-the-Loop Approach to Evaluate Rule-Based Traffic Signal Retiming Strategy by Utilizing Floating Car Data

Sharma, Salil; Lussmann, Jonas; So, Jaehyun

**DOI**

[10.1109/TITS.2018.2877585](https://doi.org/10.1109/TITS.2018.2877585)

**Publication date**

2018

**Document Version**

Accepted author manuscript

**Published in**

IEEE Transactions on Intelligent Transportation Systems

**Citation (APA)**

Sharma, S., Lussmann, J., & So, J. (2018). Controller Independent Software-in-the-Loop Approach to Evaluate Rule-Based Traffic Signal Retiming Strategy by Utilizing Floating Car Data. *IEEE Transactions on Intelligent Transportation Systems*, 20(9), 3585-3594. <https://doi.org/10.1109/TITS.2018.2877585>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

# Controller Independent Software-in-the-Loop Approach to Evaluate Rule-Based Traffic Signal Retiming Strategy by Utilizing Floating Car Data

Salil Sharma<sup>1</sup>, Jonas Lüßmann, and Jaehyun So

**Abstract**—Floating car data present a cost-effective approach to observing the traffic state. This paper explores whether floating cars can substitute stationary detection devices (e.g., induction loops) for observers within traffic responsive control systems. A rule-based traffic control method at the local intersection level is proposed in this paper by utilizing the floating car data. The control method involves a three-fold approach: link-level speed forecasting, data-driven traffic flow estimation, and split optimization. To estimate traffic flow, a multivariable linear regression model is developed by utilizing forecasted link-level speed, signal control variables, and link length as predictors. The method is tested using a controller (hardware)-independent software-in-the-loop approach. Compared with the existing fixed-time control operating in Starnberg, Germany, the proposed method is able to improve the level of service of the signalized intersection when tested for different levels of market penetration of the floating cars. The findings underpin the use of floating car data in online traffic control applications; the benefits will increase with an increase in market penetration of floating cars. Overall, this paper presents a fully integrated technical system that is ready to be used in the field. The proposed system can be implemented at the tactical level of urban traffic-control hierarchy employed in Germany.

**Index Terms**—Signal re-timing, transport big data, floating car data, software-in-the-loop simulation, traffic flow estimation, urban traffic control, rule-based control.

## I. INTRODUCTION

**I**N GERMANY, urban traffic signal control is structured hierarchically [1]. The hierarchy has three levels of

Manuscript received July 5, 2017; revised February 12, 2018, May 20, 2018, and September 17, 2018; accepted October 11, 2018. This work was supported in part by the Bavaria State Ministry of Economic Affairs, Infrastructure, Transport and Technology, Germany (Vinstar project), and in part by the Institute for Information & communications Technology Promotion Grant funded by the Korea Government (MSIT) through the Development of 100Mbps grade V2X and LTE communication system for cooperative driving under Grant 2017-0-004000. The Associate Editor for this paper was C. S. Ioakimidis. (Corresponding author: Salil Sharma.)

S. Sharma was with the Technical University of Munich, 80333 Munich, Germany. He is now with the Department of Civil Engineering and Geosciences, TU Delft, 2628 CN Delft, The Netherlands (e-mail: s.sharma-4@tudelft.nl).

J. Lüßmann is with GEVAS Software GmbH, 80335 Munich, Germany (e-mail: jonas.luessmann@gevas.de).

J. So is with The Korea Transport Institute, Sejong City 30147, South Korea (e-mail: jso@koti.re.kr).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TITS.2018.2877585

decision with respect to the spatial and temporal scope. At the top, strategic level accounts for the long-term and large scale changes of the traffic flow, e.g., provisions of different signal programs for peak and off-peak period. In the middle, tactical level accounts for the mid-term changes, still sticking to the general strategy, e.g., coordination of several intersections by adjusting some of the control variables. At the bottom, operational level responds to the short-term changes in traffic demand, e.g., public transport prioritization. In recent years, control algorithms have been developed to comply with the control hierarchy. A rule based control is usually employed at the operational level and a model based control at the tactical level. Rule-based control runs through a logical flow chart every time step and decision is made based on the predefined logical rules. On the contrary, model-based control utilizes an inherent model and optimization algorithm [2], [3]. In lieu of transparency, the model based control algorithms are difficult to be interpreted by the traffic engineers who are the end-users of such traffic control products. Therefore, this paper focuses on product usability and ease with which system is supplied by the traffic engineers as the key criteria while developing a rule-based control method for tactical level of implementation in Germany. The proposed control algorithm can be integrated within the traffic engineer's workstation so that the product can be tested and deployed in the field through one system.

The traffic signal control can be categorized into two distinctive methods, fixed-time and traffic responsive. As for the fixed-time control, the signal parameters including cycle length and phase sequence are fixed beforehand [4], while in traffic responsive control the signal parameters are served based on the vehicular demand detected. The fixed-time control does not require the detectors thus making the installation, operation and maintenance cost-effective but it can be vulnerable to the unforeseen changes of traffic demand. The traffic responsive controls often rely on optimization methods and traffic data, which is gathered from the stationary observers such as induction loops or cameras [5]. In contrast to the stationary observers, floating car data (FCD) are emerging means to observe the traffic state where the car itself acts as a probe. The probes are fitted with the consumer GPS navigation devices, mobile phones, or Bluetooth devices which can transmit the location data or time stamp to the control center. The FCD is also available commercially through TomTom or INRIX in

the form of link-level speed and travel time data. Reference [6] notes that TomTom uses several data collection sources such as portable navigation devices (PND), smartphone navigation applications, automotive in-dash and on-dash devices, fleet management solutions, 3<sup>rd</sup> party smartphone navigation applications, and public traffic information. Through the large fleet of GPS devices, TomTom manages to collect the FCD on a continuous sampling basis. In addition to real-time FCD, TomTom also uses a historical traffic data base as a background source for the data fusion process to improve the confidence level of the link-level speed estimates. Therefore, the FCD in this context can be considered as *transport Big Data* that is composed of abundant traffic data from public traffic information and multiple nomadic devices such as PND and smartphones. With this capability of FCD, this study aims to develop a framework for utilizing the commercially available FCD as a possible alternative of the stationary detection systems (e.g., the induction loops) to act as an observer in the traffic control architecture.

To test the performance of traffic control within simulation environment, three approaches exist: emulation-in-the-loop simulation (EILS), software-in-the-loop simulation (SILS), and hardware-in-the-loop simulation (HILS) [7], [8]. EILS approach utilizes an application programming interface (API) to interact with the traffic simulator. Although EILS provides higher simulation speed, lower installation costs, and perfect coupling with the traffic simulator, it does not match the variety and sophistication provided by a field controller. Moreover, EILS does not have any latency in updating the control variables as the whole process is run within the simulation environment. Since latency occurs during the generation of data and communication of data between different components, HILS and SILS represent the systems performing operations in real-time. HILS uses actual hardware while SILS uses a virtual controller (hardware). The inability of HILS to run faster or slower than the real time makes SILS a preferred approach. However, there exist different types of controller hardware used in the field to implement a traffic signal control system. The SILS approaches developed previously do not take into account the controller's specifications which differ from manufacturer to manufacturer. There is a need to address this research gap. We have utilized the TRENDS kernel software [9] which is used in the core of many traffic controller units in the DACH area (Germany, Switzerland, and Austria). It is still the beginning phase for the FCD; therefore, SILS approach can test various scenarios of varying market penetrations which are not yet available in reality.

Consequently, this paper aims to develop a traffic signal retiming strategy by updating the splits at the local signalized intersections level to respond to the mid-term changes in the traffic demand. The decision interval for control action is set as 15 minutes. Instead of using raw floating car data, the method uses processed and aggregated link-level speed data. A modified form of single exponential smoothing, utilizing intermittent floating car data, is applied to forecast the link speed for next decision interval. Afterwards, a multivariable linear regression method is developed to estimate the traffic flow per lane using forecasted link-level speed, control

variables, and infrastructural characteristics in the form of link length. By utilizing the flow estimates from regression model, the splits are updated based on the control actions set by a rule-based algorithm. Optimization of traffic signal timing plans using transport Big Data is recently being highlighted by many researchers based on its effectiveness in terms of delay reduction at signalized intersections, thus this study is a timely approach and will provide expected benefits of the transport Big Data (i.e., pre-processed and aggregated FCD in this study) in terms of traffic signal operations. With this main objective, the study investigates the impact of penetration rate of FCD on the effectiveness of traffic control strategy and evaluates the performance of the proposed traffic control method using controller independent software-in-the-loop simulation (i.e., case study). The traffic signal retiming strategy is tested at the signalized intersections located in Starnberg, Germany, and the performance the proposed approach is assessed against the existing fixed-time signal control in Starnberg.

The key contributions of this paper are as follows. First, this paper utilizes pre-processed and aggregated FCD in developing a traffic signal retiming strategy for tactical level of control which can be useful for small-and medium level cities seeking to reduce the delay at the signalized intersections in a cost-effective way. Second, this paper presents a fully integrated technical system with which traffic engineers can design, test and deploy the traffic control product. Moreover, the control method itself is transparent since it is rule-based; hence, the end-users can understand the underlying mechanics of the system. Third, this paper presents a SILS approach to test the rule-based control method which uses the same software as in the controller (hardware) units. The SILS approach not only emphasizes the need to create open systems but also provides a robust testing solution.

The structure of this paper is as follows. Section 2 reviews the state-of-the-art. Section 3 describes the study area in Starnberg, Germany. The system architecture of the software-in-the-loop simulations is presented in section 4. The methodology is discussed in section 5 and the results are presented in section 6. Lastly, section 7 draws conclusions and provides suggestions for future research.

## II. LITERATURE REVIEW

The fundamental distinction between the FCD and fixed-location sensor data is that FCD represent the sample of overall traffic. Besides, the floating cars can be located anywhere in the network and are not necessarily distributed uniformly over spatial and temporal scale. Neumann [10], [11] mentions that substantial market penetration of probe vehicles is needed for traffic signal control. He notes that traffic signals are the ideal places to collect the data since vehicles are clustered there. Neumann's works show that the accuracy of traffic state estimation could be improved by aggregating over longer intervals and also using data fusion techniques to incorporate the ground-truth estimates of the traffic demand.

The FCD have been used in past for both urban and motorway traffic management schemes. Reference [12] presents the fusion of loop detector and FCD in traffic management center of Berlin, Germany. The taxi and bus FCD are used to

complement the loop detector so as to analyze the real-time level of service on the roads. In project COLOMBO [13], a decentralized traffic control based on FCD is proposed where every traffic light acted as an individual agent which can also coordinate with the other agents. The traffic control is shown to work with the low penetration of FCD as it uses the makes use of speed and acceleration instead of the number of cars on a particular approach. Afterwards, a swarm optimization based traffic control algorithm is devised; it has been shown that this algorithm can match the performance of the actuated traffic signal control at 25% of market penetration of FCD. This paper also utilizes the decentralized framework where the neighboring traffic lights can share the state of their control variables but develops a rule-based traffic signal retiming strategy. Reference [14] presents an adaptive traffic signal control based on FCD. They use a greedy algorithm which assigns more green to the approach which has more floating cars. In cases when the system does not detect any floating car, the algorithm falls back to the fixed time signal program. However, the spatial and temporal variability of FCD may influence the results of the greedy algorithm.

For motorways, [15] develops a FCD based variable speed limit (VSL) system. The performance of this system is compared with the existing induction loop based VSL system. The results indicate the FCD based VSL system, even at 6%–8% of market penetration, approximates the loop based VSL and underpin potential of FCD in designing traffic management strategies for motorways. Reference [16] notes that the FCD can be used to derive the travel times over a section of the motorway and this information can be utilized to design the better and cost-effective sectional control strategies for motorways. They made use of a fully deployed an operational fleet of 200,000 floating cars running on Belgian motorways. With such a high penetration of FCD, the findings suggest that the traditional camera based section control can be replaced by the fleet (FCD) based traffic management.

In contrast to above works, the commercially available FCD not only use real-time information but also enriches the historical database with this information. Such databases can contain the link-level traffic state information in the form of speed or travel times. The advantage of maintaining a historical database is that it can be used in online traffic management applications by avoiding the influence of spatio-temporal variability of FCD. Therefore, this paper emphasized the usage of a historical traffic database created by FCD and attempts to fill that gap by designing a transparent rule-based control at the tactical level for traffic engineers.

### III. SYSTEM DESCRIPTION

The network consists of three neighboring intersections located in Starnberg, Germany (see Fig. 1). The main corridor is German federal highway B2 which runs through the city. Therefore, Starnberg city, affected by the heavy through traffic, makes an ideal example case for this paper to test a tactical level of control strategy. Following intersections are considered: (1) Söckinger Straße-Bahnhofstr., (2) Hauptstr.-Tiefgarage, and (3) Hanfelderstr.-Wittelsbacherstr.-Münchenerstr.

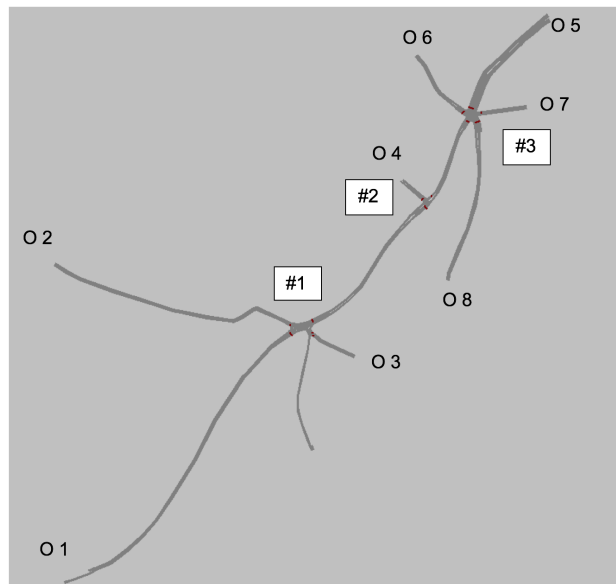


Fig. 1. VISSIM network for Starnberg, Germany.

### IV. SYSTEM ARCHITECTURE

This section describes the system architecture to test our rule-based control method. We have developed an SILS approach, which is independent of traffic light signal system control equipment (i.e., hardware) to test the control method.

#### A. Traffic Simulator: VISSIM

VISSIM 7.00 is used in this paper [17]. It is a microscopic, time step and behavior-based simulation model, which uses the psycho-physical car-following model for longitudinal movements and a rule-based algorithm for lateral movements. The VISSIM network of three intersections located in Starnberg is shown in Fig. 1.

1) *Network Data*: This paper utilizes the VISSIM settings from the work of Lüßmann [3]. In his dissertation, VISSIM parameters for desired speed, acceleration and deceleration distributions were calibrated by conducted field measurements for a corridor in Munich. It is assumed that the urban driving behavior for Starnberg, which is close to Munich, can be sufficiently captured by using those parameters. Wiedemann 74 model [18] is chosen as the car following model. 2% of heavy goods vehicles are included in vehicle composition which is the default setting in VISSIM.

2) *Floating Car Data in VISSIM*: Instance of the vehicle type (private car) is created. Afterwards, a new vehicle class is created which included this instance. The percentage of the vehicle types can be changed under the header of vehicle composition. In this menu, the desired market penetration can be provided to the instances of the vehicle types which represent probe vehicles in this simulation study. The creation of vehicle class helps in collecting the desired output from the simulation. Link level data is collected every cycle of the signal program [17].

3) *Variable Demand*: Traffic demand scenario is constructed for two hours (see Fig. 2). First 30 minutes-time

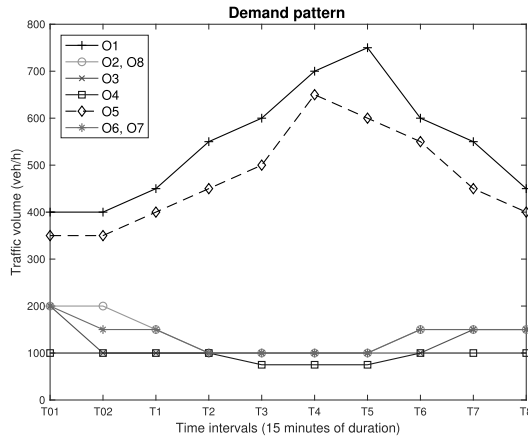


Fig. 2. Variable demand.

TABLE I  
BASIC INTERSECTION DATA FOR EXISTING FIXED  
TIME CONTROL IN STARNBERG

	Intersections		
	#1	#2	#3
Signal groups	4	3	5
Stages	3	2	4
Interstages	3	2	4
Switch-on program duration (s)	26	17	21
Switch-off program duration (s)	27	18	25
Cycle time (s)	90	90	90
Green time for main direction (s) (for fixed time controls)	41	69	31
Instant to switch signal plans (s)	10	50	55

periods, T01 and T02, are chosen as the warm up period so that the network gets filled. Afterward, eight flow intervals of 15 minutes of duration, numbered from T1 to T8, are created. The demand for main corridor, O1 and O5, forms a peak. The routing decisions are defined with low turning ratios to facilitate heavy through traffic.

### B. Creation of Basic Intersection Supply With CROSSIG

CROSSIG [19] is a state-of-the-art workstation for traffic engineers to plan and test control mechanisms for light signal systems and supply them with data in a quality-assured way. In addition, CROSSIG is the basic platform for creating traffic-technological data, which is required for further applications such as TRELAN/TRENDS [9], [20] and VISSIM [17]. At first, the data about the signal groups, conflict points and intergreen matrix is supplied in CROSSIG. Afterwards, the signal programs, switch-on and switch-off programs, interstages and stage sequence plans are generated for the respective intersection. Table I shows the basic intersection data for the fixed time control which is operating in Starnberg.

### C. User Specific Control With TRELAN

TRELAN stands for TRaffic Engineering LANguage with which user-defined control programs can be created. TRELAN is based on the German guidelines for traffic signals which use flow charts to depict the complete process

of traffic-actuated control. The description language TRELAN and the graphical editor openTRELAN make traffic control transparent and conveniently transferable into the controllers [20]. openTRELAN is used to develop the control method in this paper.

openTRELAN also has an inbuilt compiler which makes it easier for the end user to compile the source text by using graphical interface. Successful compilation generates two files: \*.vxe and \*.vad file. The vxe file contains a flow diagram, created using openTRELAN and produced by TRELAN compiler. The vad file includes the source code for the flow diagram which is required for debugger in the TRENDS system [20].

1) *VL Detector Variables*: The relevant link speed data can be retrieved from VISSIM and supplied to the virtual traffic control to process this data. New variables in TRELAN are created to retrieve the information about the link speed. These variables are denoted as VL1, VL2, ..., VL64, with value ranges from  $-3200$  to  $+3200$ . These detectors are controlled via the BALANCE telegram [21].

### D. TRENDS Control System

TRENDS stands for TRaffic ENgineering Development System. It is a computer-based system which can test and visualize the functionality and performance of traffic control methods, in a cost-effective manner, before deploying in the field. TRENDS kernel is the intelligent part of TRENDS, and it forms the core around which the controller is layered.

Since TRENDS kernel processes the compiled TRELAN file, a traffic light signal system can be planned and tested independent of the controller's manufacturer. The vxe and vad file become the input for TRENDS. TRENDS is a comprehensive concept which puts full control in the hands of the user since the traffic workflow being used for running the actual controller is the same workflow running on test station [9].

### E. Creating Controller Supply

The traffic engineering data, created with TRELAN/TRENDS, describe the signal control for an intersection. The data are then transmitted to the microcomputer of the controller and are used for the control actions generated by TRENDS kernel. The controller supply data is generated from TRELAN/TRENDS test station in the form of controller supply file (\*.stg) and TRELAN supply file (\*.vxb). The stg file contains all the information about the technical layout of the signal and the relevant traffic engineering data; while, the vxb file includes all the traffic engineering data necessary for realizing the signal control in the controller. The vxb file resembles a container with data in a binary format and it is embedded in the storage system of the controller's operating system so that TRENDS kernel can process this data to realize traffic signal control [9].

### F. Software-in-the-Loop Simulation

Fig. 3 shows the system architecture for the SILS used in this paper. VISSIM provides application programming interface (API) as add-ons to automate user-specific workflow.

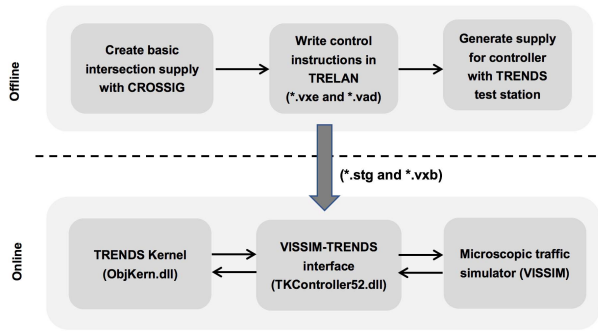


Fig. 3. System architecture for software-in-the-loop simulation.

External signal control is one such application which allows user to link external controller with the traffic simulator [17]. The source code of signal controller DLL for VISSIM comprises a frame which facilitates the communication between VISSIM and TRENDS kernel which inherits actual control strategy. For the purpose of this paper, TKController52.dll provides an interface between kernel (ObjKern.dll) and microscopic traffic simulation model (VISSIM). Moreover, TKController52.dll facilitates the exchange of information between VISSIM data format and TRENDS data format.

## V. METHODOLOGY

This section provides a three-fold methodology for rule-based signal re-timing strategy: link-level speed forecasting, traffic flow estimation, and split optimization. First, the link-level speed data, processed and aggregated by VISSIM, is retrieved every cycle length of the signal program. Then, a modified form of exponential smoothing technique is used to forecast link-level speed which could also make use of intermittent data. Second, a multivariable regression model is developed to estimate traffic flow per lane for an approach. Lastly, split optimization uses the flow estimates of various approaches and generates new splits in the ratio of traffic flow to saturation flow. The strategy is of reactive in nature and it updates splits after every 15 minutes. The proposed strategy qualifies as the rule-based traffic control method [2] since it can be coded with openTRELAN and it is suitable for the tactical level of traffic control [1].

### A. Link-Level Speed Forecasting

The method aims at tactical level of decision making. Therefore, the time-frame for the control actions is set to ten times the cycle time. From now on, the time frame will be referred as decision interval in this paper. The control action is of reactive type. Based on the data from the last decision interval, control variables are set for the next decision interval.

1) *Processing Link-Level Speed Data From VISSIM:* VISSIM processes and aggregates the speed data coming from the probe vehicles. Then VISSIM sends the processed and aggregated link-level speed profile every cycle, i.e., the speed observations for the time period of one cycle length of the signal program to the TRENDS kernel. In this way, the speed forecasting method utilizes the observed speed data per cycle for ten cycles in the decision interval. If the link segments

- 1: **Input:** processed and aggregated link-level speed by VISSIM for  $t$  time period ( $v_t$ ), Number of cycles for aggregation ( $n$ ), smoothing factor ( $\alpha$ )
- 2: **Output:** forecasted link-level speed ( $S_t$ ) for  $t$  time period
- 3: Initialize the forecast for first period with the first speed observation
- 4:  $S_1 \leftarrow v_1$
- 5: **for**  $t \leftarrow 2, n$  **do**
- 6:     **if**  $v_t \neq 0$  **then**
- 7:          $S_t = \alpha \cdot v_t + (1 - \alpha) \cdot S_{t-1}$
- 8:     **else**
- 9:          $S_t = S_{t-1}$
- 10:    **end if**
- 11: **end for**
- 12: **if**  $S_n = 0$  **then**
- 13:      $S =$  Progressive speed of the link or link speed retrieved from the historical database, if available
- 14: **else**
- 15:      $S = S_n$
- 16: **end if**
- 17: **return**  $S$

Fig. 4. Algorithm for link-level speed forecasting.

in VISSIM are created by joining a number of links together, the link-level speed data is computed via harmonic mean of individual link speeds.

2) *Forecasting Method:* Single exponential smoothing technique is used to forecast the average link speed for the next decision interval [22]. With spatial and temporal variability of floating cars in the overall traffic, the traditional single exponential smoothing algorithm is modified so that a suitable forecast can be produced in case of missing data in a time interval. The value of smoothing constant ( $\alpha$ ) is selected as 0.2 suggested by [23] and [24]. The proposed algorithm (see Fig. 4), is able to handle the intermittent data. A further modification is also involved in case of low traffic flowing on the roadway. Since this paper utilizes the floating car data, low penetration of probe vehicles in overall low traffic may not be sufficient for this algorithm to produce a reliable forecast. In that case, the forecasted average link speed for next decision period is set to progressive speed (with a lower bound of 0.85 times the design road speed) at which platoons of vehicles can pass a sequence of intersections without stopping [2]. Moreover, the use of progressive speed aligns towards one of the contribution of this work, i.e., design of a tactical level traffic control method. For this paper, the design speed for the roads is assumed as 50 km/h and the progressive speed is computed as 0.9 times the design road speed, i.e., 45 km/h. If a historical database consisting of link speed profiles, e.g., TomTom HD FLOW, is available to the end user, the appropriate speed value should be used. The forecasted speed serves as the input to estimate the traffic flow.

### B. Data-Driven Traffic Flow Estimation

1) *Hypotheses:* This paper utilizes multivariable linear regression approach to estimate traffic flow to incorporate

variables other than speed. When the traffic state changes from uncongested to congested regime, traffic flow starts behaving differently with link speed. In uncongested conditions, the traffic flow can be described as the multivalued function of speed, i.e., the speed is not varied much with a change in traffic flow [25]. When the traffic state approaches congested phase, link speed decreases with an increase in traffic flow. From modeling point of view, speed to flow conversion is easier in congested phase. In this paper, regression based relationship between the speed and flow is developed, which is compatible with the rule based control framework [2] as it can generate approximate estimates of the response variable and provide a substantial nature of evidence about the changes in response when input is changed.

In an urban scenario, traffic state is governed by the presence of traffic lights. Not only link speed, but also factors such as cycle length, ratio of green to cycle time, link length, and the manner in which vehicles arrive at the intersection affect the traffic state. The link length is the geometric property of the road way. Cycle time and green time for various approaches can be retrieved from the controller. However, it is difficult to get the precise value about the vehicle's arrival pattern in lieu of any detection device. Therefore, next section describes an approach to quantify vehicle's arrival pattern by enabling the communication between the neighboring controllers.

2) *Arrival Factor*: A novel parameter, called as *arrival factor*,  $A$ , is introduced in this paper to explain the variability of link speed with respect to the control parameters such as green time, cycle time, and offset between the neighboring traffic signals. Our preliminary experiments show that the link speed increases with the degree of coordination between the neighboring signals and with  $t_g/t_u$  as more vehicles can pass through the coordinated intersection without making a stop.

The value of  $A$  is set as  $t_g/t_u$  when there is no upstream intersection in the network.  $t_g$  and  $t_u$  refer to the green time and cycle time for an intersection respectively. In the cases of neighboring intersections, the arrivals on a particular intersection are dictated by the presence of an upstream signal. Then  $A$  is set as the ratio of shared green time between downstream and upstream intersection to the green time of the upstream intersection. The shared green time is determined by using the progression speed and the offset between the intersections. The progression speed is chosen as 45 km/h [2].

The inclusion of  $A$  in the rule-based control method was challenging as the traffic signal program works in a cyclic way. The signal program starts from zero and then ends at the cycle time, and then the clock returns to the zero again for the next cycle. It is therefore important to know the green time shared between the two intersections when computing the arrival factor. The problem occurs when the green time for a phase starts in one cycle and continues to the next cycle. Therefore, the logical rules were developed to divide the green band in to two: from the start of the green up to the absolute time where cycle ends and then from zero to the end of the green time. In this way, algorithm *arrival\_factor* (Fig. 5) makes use of continuous timeline. Then it calls the function *sharedG* (Fig. 6) to compute the respective amount of the shared green time between the two intersections.

```

1: Algorithm: arrival_factor
2: Input: start and end of green time:  $[a, b]$  and  $[c, d]$  where  $a$  and  $b$  are start and end of green for intersection 1;  $c$  and  $d$  respectively for intersection 2.
3: Output: arrival factor  $A$  such that  $0 \leq A \leq 1$ 
4: Variables:  $S$  is used to store the shared offset between two intersections.  $G$  denotes the green time for upstream intersection. The value of  $G$  would be  $a \sim b$  or  $c \sim d$  depending on which intersection is positioned upstream.  $C$  denotes absolute time when cycle ends.
5: if  $a < b$  &&  $c < d$  then
6:    $S = \text{sharedG}(a, b, c, d)$ 
7: else if  $a > b$  &&  $c < d$  then
8:    $S = \text{sharedG}(a, C, b, d) + \text{sharedG}(0, b, c, d)$ 
9: else if  $a < b$  &&  $c > d$  then
10:   $S = \text{sharedG}(a, b, c, C) + \text{sharedG}(a, b, 0, d)$ 
11: else if  $a > b$  &&  $c > d$  then
12:   $S = \text{sharedG}(0, b, 0, d) + \text{sharedG}(a, C, c, C)$ 
13: end if
14: return  $A = S/G$ 

```

Fig. 5. Algorithm to compute arrival factor  $A$ .

```

1: Algorithm: shared_green (sharedG)
2: Input: Two green time intervals for which shared offset is to be determined. The two intervals are noted as:  $[a, b]$  and  $[c, d]$ .
3: Output: shared green band  $S$  between two neighboring intersections
4: if  $(c \leq a \leq d) || (c \leq b \leq d)$  then
5:   if  $a \geq c$  &&  $b \leq d$  then
6:      $S = b - a$ 
7:   else if  $a \leq c$  &&  $b \geq d$  then
8:      $S = d - c$ 
9:   else if  $a \geq c$  &&  $b \geq d$  then
10:     $S = d - a$ 
11:   else if  $a \leq c$  &&  $b \leq d$  then
12:     $S = b - c$ 
13:   end if
14: else
15:    $S = 0$ 
16: end if
17: return  $S$ 

```

Fig. 6. Algorithm to compute the green time shared between the neighboring controllers.

The *control marks* in CROSSIG are used to enable communication between the neighboring controllers so that they can exchange variables for above algorithms [20].

3) *Preliminary Variables for Regression Modeling*: Based on the hypotheses, it can be inferred that the link speed gets affected by cycle length, ratio of green to cycle time, link length, and the manner in which vehicles arrive at the intersection. In our preliminary experiments, link speed is found to be positively correlated with respect to these variables. An increase in green time within one cycle would increase the link speed. Longer the links, more will be the speed

TABLE II  
DESCRIPTIVE STATISTICS OF THE DATA COLLECTED  
FOR REGRESSION MODEL

Variables		Intersections		
		#1	#2	#3
$v$ (km/h)	mean	26.83	27.02	18.57
	Standard deviation	14.25	11.30	9.61
$Q$ (veh/h)		449.91	436.91	266.13
		306.01	238.67	214.05
$L$ (m)		396.02	222.80	207.48
		256.93	106.35	85.30
$\frac{t_g}{t_u}$		0.42	0.46	0.25
		0.18	0.16	0.15
$A$		0.50	0.67	0.36
		0.26	0.26	0.29
Sample size		1210	1050	1495

as vehicles begin to decelerate near to the intersection and the upstream of the link would be free from these effects. If vehicles are passing the intersection in groups or platoons, less intersection delay would contribute to increased speed. Therefore, multivariable linear regression is employed to deal with the joint distribution of the predictor variables.

Traffic flow per lane is used as the response variable instead of traffic flow. The relationship of independent variables with that of dependent variable, i.e., traffic flow per lane is drawn roughly. If the speed is lower, more will be the traffic flow due to the friction among the vehicles prevailing on the link. Since capacity of an approach at an intersection is governed by the ratio of green to cycle time, higher the capacity more vehicles will pass through the intersection. At this stage, it is however difficult to predict the possible signs of other predictor variables as the joint distribution of link speed with the rest of the predictors would affect the relation. The hypotheses thus formed bring us to the next section to test the hypotheses by setting up a controlled experiment to collect data and build the regression model.

4) *Synthetic Data*: Data have been collected from VISSIM simulation runs. Since the three intersections located in Starnberg have distinct geometry in terms of link lengths, number of approaches and different stages for the signal plan, intersection specific regression models are developed. The link lengths are assumed to be fixed. Since three intersections have same cycle length of 90 s, the same cycle length is considered for the data collection. The rest of the variables are varied in simulation environment to generate diverse data for performing regression analysis, e.g., the green time of various approaches, the adjust factor by altering the level of coordination between the neighboring intersections. The volume to capacity ratio is varied from very low (0.2) to close to saturation (1.0) in order to produce free flow as well as close to capacity conditions. Moreover, the simulation is run for ten times with different random seed in order to incorporate the possible random variations existing due to vehicle arrival and departure patterns. Data collection is done for an interval of 15 minutes. Table II presents the descriptive statistics.

5) *Regression Model Estimation and Validation*: Four independent variables are considered as explanatory or predictor variables: link speed ( $v$ ), ratio of green to cycle time ( $\frac{t_g}{t_u}$ ),

link length ( $L$ ) and arrival factor ( $A$ ). Further interaction terms-product of two predictors-or power terms can be added to help explain as much variance as possible in the response variable. Cycle time is not considered explicitly as independent variable as it has been shown in [25] that cycle time does not influence speed significantly. Further, the remaining effects are covered by the inclusion of ratio of cycle to green time. As explained earlier in this paper, average link speed  $v$  responds to the changes in the ratio of green to cycle time  $t_g/t_u$ , link length  $L$ , and arrival factor  $A$ . Therefore, it is important to include these interactions in the model. The fundamental diagram shows that speed-flow relation is non-linear in nature; hence, a square term is added to explain the same. Thus, the initial variable selection has provided us with four independent ( $v$ ,  $\frac{t_g}{t_u}$ ,  $L$  and  $A$ ), four derived variables ( $v^2$ ,  $v\frac{t_g}{t_u}$ ,  $vL$ , and  $vA$ ), flow per lane ( $\frac{Q}{N}$ ) as the response variable.

To arrive at the optimal model, stepwise approach is used. The predictors, being insignificant ( $p$ -value $>0.05$ ), are removed in this iterative procedure. The models thus obtained are checked for their performance on the test data. Besides, Cook's distance is used to identify the outliers in the predictors and assess their impact on the response variable [26].

Since, the model contains the interaction and power terms, Belsley collinearity diagnostics is used [26]. The cases where multicollinearity is detected, the predictors are centered on the mean [27]. The mean values ( $\mu$ ) used for centering are provided in Table III for four independent variable.

The regression model presented in Table III is calibrated for the 70% of the data and is validated on the remaining 30% of the data. The test and training error for the model are also reported in Table III.

### C. Updating Splits

1) *Assumptions*: The method to update the splits is based on the following assumptions.

- 1) Saturation flow of a lane is assumed as 2000 veh/h. For future studies, it is advised to use a field-calibrated saturation flow.
- 2) Capacity of a lane is defined as:  $C_i = \frac{t_{g,i}}{t_u} \cdot q_{s,i}$ . Here,  $C_i$  refers to the capacity of lane  $i$  in veh/h.  $t_{g,i}$  is the green time allotted to lane  $i$  in s.  $t_u$  is the cycle time in s.  $q_{s,i}$  is the saturation flow for lane  $i$  in veh/h.
- 3) Link length is fixed.
- 4) Regression coefficients are estimated beforehand.
- 5) Cycle length, offset, and stage sequence are fixed.
- 6) Traffic flow is assumed to be uniformly distributed across all the lanes.
- 7) Left turning movements are not considered explicitly. Even if there exists exclusive left-turning lane, single lane approach is applied to calculate traffic flow for that approach. In single lane approach, the traffic flow for all the movements is summed up and presented as a whole for that approach.
- 8) If regression model estimates negative flow, then the value of traffic flow for that approach is set as 50 veh/h thus signifying low traffic flow conditions. The negative flow is possible only in the cases when speed is high.



TABLE III  
PARAMETERS' ESTIMATES FOR THE INTERSECTION

Independent variables		Intersections		
		#1	#2	#3
Intercept	coefficient	534.49	345.42	121.66
	Standard error	13.187	9.2684	6.6987
	p-value	<0.05	<0.05	<0.05
$v$ (km/h)		-17.141	-26.588	-31.114
		0.90477	1.0297	1.0461
		<0.05	<0.05	<0.05
$\frac{t_g}{t_u}$		1183.9	1614.3	1487.2
		57.732	48.1	38.841
		<0.05	<0.05	<0.05
$A$		470.8	360.08	451.66
		56.401	34.696	30.964
		<0.05	<0.05	<0.05
$L$ (km)		403.18	559.57	1635.6
		35.569	60.603	71.065
		<0.05	<0.05	<0.05
$vA$		23.925	41.162	40.355
		3.4855	3.8966	3.6399
		<0.05	<0.05	<0.05
$vL$		-8.4738	-13.412	47.244
		3.244	6.009	9.9346
		<0.05	<0.05	<0.05
$v^2$		-0.82017	-1.0364	-0.78316
		0.0874	0.08329	0.10098
		<0.05	<0.05	<0.05
$\mu(v)$		25	27	18
$\mu(\frac{t_g}{t_u})$		0.4	0.4	0.2
$\mu(A)$		0.5	0.6	0.3
$\mu(L)$		0.3	0.2	0.2
Sample size		830	735	1050
Training error		166	136	101
Test error		159.9	139.59	119.51
Adj. R-squared		0.711	0.679	0.777
F-statistics		292	223	523
p-value		0	0	0
Belsley condition index		7.44	8.05	13.2

2) *Method*: The split optimization method is based on [28] and it utilizes the traffic flow estimates from the regression model. The steps are described below:

- 1) Find critical flow per lane in the stage which would be the maximum of the flow for various approaches in the same stage. The flow per lane depends on the level of coordination between the signals and this dependence is expressed by arrival factor in previous section.
- 2) Compute the required green time for a stage by dividing the critical flow per lane by the saturation flow.
- 3) Check if minimum green time conditions are met. If not then increase the required green time computed in step 2 to the minimum green time value.
- 4) Compute the available green time by subtracting the sum of intergreen times from the cycle time.
- 5) Allocate the available green time, calculated in step 4, in proportion to the ratio of critical flow to the saturation flow for a stage, and given the conditions for minimum green time are met.

## VI. IMPLEMENTING TRAFFIC SIGNAL RE-TIMING STRATEGY

### A. Scenarios

The base case is the fixed-time control running a morning time plan (see Table I). Three different values of market

TABLE IV  
STATISTICAL RESULTS OF AVERAGE DELAY PER VEHICLE (S)

Intersections	Base case Fixed time	FCD Market penetration			
		20%	50%	100%	
#1	Mean	32.42	29.49	36.11	35.36
	Standard deviation	1.08	2.74	2.73	3.29
	p-value (T≤t) two-tail Significance		0.34	0.23	0.41
#2		17.31	12.89	10.69	8.28
		2.57	4.78	2.25	0.12
			0.43	0.06	<0.05
#3		52.89	37.99	34.70	25.45
		2.02	11.85	8.17	0.45
			0.24	0.055	<0.05
			*	**	

\*\* indicates statistically significant difference of means at 95% CI.

\* indicates statistically significant difference of means at 90% CI.

Note: CI refers to the confidence interval.

penetration of floating cars, to cover low to high penetration, are considered: 20%, 50%, and 100%.

### B. Performance Measure

Average delay per vehicle for the specific intersection is used as the measure of performance. Average delay per vehicle is an indication of the time that a vehicle spends in waiting before it could cross the intersection. The data is retrieved from VISSIM by utilizing its *node evaluation* feature. The evaluation for three intersections is carried out separately in two ways. First, the vehicle delay is averaged for two hours of simulation run, i.e., the periods T1 to T8. Second, the vehicle delay is averaged individually for the eight time intervals, T1 to T8 and presented as a trend graph.

### C. Results for Two Hours of Simulation Run

For two hours of simulation run, average delay per vehicle is computed at the intersection level for all the scenarios. Table IV provides a comparison of the values of average delay per vehicle between the base case and three other cases representing different market penetration of floating cars for the demand pattern presented in Fig. 2. The student t-test (assuming unequal variance) is conducted to test if there exists a significant difference between the values of performance measure for two cases, i.e., between base case and one other market penetration scenario at 95% and 90% of confidence level.

As evident from the Table IV, the intersection 1, 2 and 3 are operating at level of service (LOS) 'C', 'B' and 'D' respectively for the base case [28], [29]. For 20% market penetration of the floating cars, the average delay per vehicle is not significantly different from the base case for all three intersections. The intersections continue to operate at the LOS observed for the base case which signifies that the FCD can replace induction loops in the traffic control loop at low market penetration of 20%.

For 50% market penetration of the floating cars, intersection 1 does not report any significant changes in the average delay per vehicle. For intersection 2, significant savings are reported at 90% of confidence level. The intersection 3 improves its LOS to 'C' when estimated at 90% of confidence level.

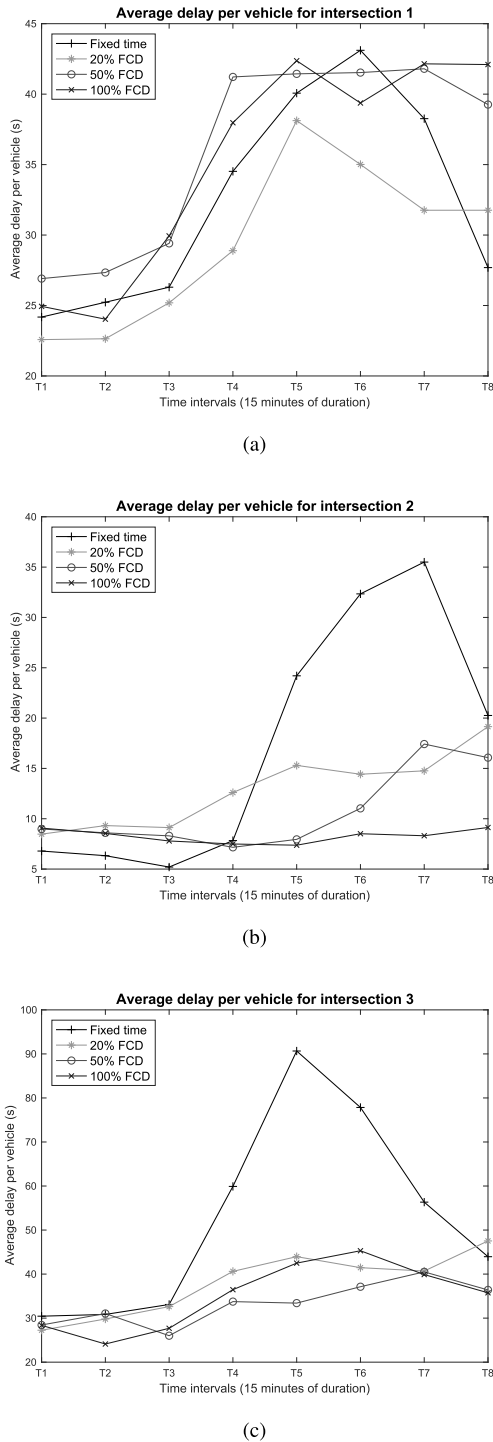


Fig. 7. (a), (b), (c) Average delay per vehicle trend over different time intervals.

The best outcomes are reported for 100% market penetration of the floating cars. Estimated at 95% of confidence level, the LOS for intersection 2 and 3 is improved to 'A' and 'C' respectively. Intersection 3 reported a saving of 51.87% in average delay per vehicle when compared with that of the base case and operated in the acceptable level of service. Intersection 1 does not show significant changes in the values of average delay per vehicle as this intersection has wide variation in the infrastructural characteristics, i.e., the link length which limited the predictive power of the regression model.

The impact of link length on link speed and traffic flow within the regression model can be an interesting research question for future.

#### D. Results for Distinct Time Intervals With 15 Minutes of Duration

This section presents a detailed graphical analysis for the performance measure, i.e., average delay per vehicle. Two hours of simulation run is divided into eight periods, each of 15 minutes of duration. The trend of average delay per vehicle is shown in Fig. 7 which follows the demand pattern for the base case at intersection 1 as it first increases and then decreases after the time period T6. The intersection 1 is found to be operating at the acceptable levels of service during the probe eight time intervals for varying market penetration of the probe vehicles.

Intersection 2 suffers increasingly larger delays for the time intervals T5, T6, and T7 for the base case. The part of the delay is contributed due to the intersection 3 (refer Fig. 1) as this intersection presents a bottleneck for the incoming demand given the amount of green time it can allot to the main direction in the base case scenario. At the intersection 2, the resulting queues due to the presence of upstream intersection 3 and the incoming traffic coming from intersection 1 add to the delay occurred to the vehicles.

The value of average delay per vehicle for intersection 3 reaches to 90 s per vehicle for the time interval T5 which also coincides with the traffic demand peak (refer Fig. 2). It reflects that the intersection 3 is operating at the worst level of service, i.e., 'F' for that particular time period. The proposed traffic control method is able to alleviate the situation for intersection 3 by giving more green time to the main directional phases and thus bringing down the average delay per vehicle. As a result, the traffic situation also gets improved at the intersection 2.

## VII. CONCLUSION

This paper has proposed a fully integrated technical system where floating cars are utilized to retune the traffic signals. It has been shown that the traffic signal retuning strategy based on FCD can perform better than the fixed time controls in reducing the delays at the intersection. The best outcomes are observed in the scenarios when the whole population could act as the probes. This technical system is highly useful for the small and medium-sized cities which often lack the detection facilities to determine the current traffic state. These cities, especially those which suffer the heavy through traffic, can utilize the potential of such a cost-effective solution proposed in this paper. Moreover, the proposed system is controller (hardware) independent and can be readily used by the traffic engineers as it brings transparency within the traffic responsive control framework at the tactical level.

The commercially available FCD use historical dataset to improve the reliability. Therefore, the data fusion of historical data could enhance the outcomes of the proposed strategy in this paper. These commercial products open up new possibilities to incorporate the FCD in devising traffic control strategies. The future research direction will be to test the

system with commercially available data from TomTom HD FLOW.

#### ACKNOWLEDGMENT

The authors would like to thank Prof. Fritz Busch and the researchers/colleagues with the Chair of Traffic Engineering and Control, Technical University of Munich, Germany, for their sincere support and advice. The first author would like to express his gratitude and appreciation to the colleagues at GEVAS software GmbH in Munich, Germany; he especially thanks Dr. Carsten Kemper, Bernd Thormann, Andreas Haftmann and Georgios German for their kind support. The authors also gratefully acknowledge the suggestions and comments provided by the Associate Editor and anonymous reviewers in improving the quality and clarity of the paper.

#### REFERENCES

- [1] B. Friedrich and C. Schutte, "Strategic control in metropolitan areas [road traffic]," in *Proc. IEEE Intell. Transp. Syst. (ITSC)*, Dearborn, MI, USA, 2000, pp. 464–469, doi: [10.1109/ITSC.2000.881112](https://doi.org/10.1109/ITSC.2000.881112).
- [2] *Guidelines for Traffic Signals: RiLSA*, Road Transp. Res. Assoc. (FGSV) Std., Cologne, Germany, 2015.
- [3] J. Lüßmann, "Coordination of traffic signal control considering the progressive speed and platoon management to reduce CO<sub>2</sub> emissions," Ph.D. dissertation, Dept. Chair Traffic Eng. Control, Civil, Geo Environ. Eng., Tech. Univ. Munich, Munich, Germany, 2015.
- [4] F. Webster, "Traffic signal settings," Road Res. Lab., London, U.K., Tech. Rep. 39, 1958.
- [5] A. Stevanovic, *Adaptive Traffic Control Systems: Domestic and Foreign State of Practice*. Washington, DC, USA: National Research Council, 2010.
- [6] J. Brouwer, "Measuring real-time traffic data quality based on floating car data," in *Proc. ATEC ITS France Congr.*, Paris, France, 2014, pp. 1–8.
- [7] A. Stevanovic, A. Abdel-Rahim, M. Zlatkovic, and E. Amin, "Microscopic modeling of traffic signal operations: Comparative evaluation of hardware-in-the-loop and software-in-the-loop simulations," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2128, pp. 143–151, Dec. 2009.
- [8] M. Zlatkovic, P. T. Martin, and I. Tasic, "Implementation of transit signal priority and predictive priority strategies in ASC/3 software-in-the-loop simulation," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 2130–2135.
- [9] GEVAS Software GmbH, Munich, Germany. (2018). *TRENDS Test Station User Manual*. [Online]. Available: <https://gevas.eu/en/trends-kernel/>
- [10] T. Neumann, "A cost-effective method for the detection of queue lengths at traffic lights," in *Traffic Data Collection and its Standardization*. New York, NY, USA: Springer, 2010, pp. 151–160.
- [11] T. Neumann, "Efficient queue length detection at traffic signals using probe vehicle data and data fusion," in *Proc. 16th ITS World Congr. Exhib. Intell. Transport Syst. Services*, 2009, pp. 1–12.
- [12] P. Vortisch, "Integration of floating car data and stationary detector data in the traffic management center Berlin," in *Proc. 13th ITS World Congr.*, 2006. [Online]. Available: <https://ertico.assetbank-server.com/assetbank-ertico/action/viewAsset?id=5448>
- [13] D. Krajzewicz, R. Blokpoel, A. Bonfietti, J. Häri, S. Hausberger, and J. Dubois-Lacoste, "Traffic management based on vehicular communication at low equipment rates," in *Proc. 22nd ITS World Conf.*, 2015, pp. 5–9.
- [14] V. Astarita, V. P. Giofrè, G. Guido, and A. Vitale, "The use of adaptive traffic signal systems based on floating car data," *Wireless Commun. Mobile Comput.*, vol. 2017, May 2017, Art. no. 4617451.
- [15] M. Houbraeken, S. Logghe, P. Audenaert, D. Colle, and M. Pickavet, "Examining the potential of floating car data for dynamic traffic management," *IET Intell. Transport Syst.*, vol. 12, no. 5, pp. 335–344, May 2018.
- [16] M. Vanlommel, M. Houbraeken, P. Audenaert, S. Logghe, M. Pickavet, and P. De Maeyer, "An evaluation of section control based on floating car data," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 617–627, Sep. 2015.
- [17] PTV Planung Transport Verkehr AG, Karlsruhe, Germany. (2014). *VISSIM 7.00-06 User Manual*. [Online]. Available: <http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/>
- [18] R. Wiedemann, "Simulation des strassenverkehrsflusses," Schriftenreihe des Instituts fuer Verkehrswesen der, Univ. Karlsruhe, Karlsruhe, Germany, Tech. Rep. 8, 1974.
- [19] GEVAS Software GmbH, Munich, Germany. (2018). *CROSSIG 6.0 User Manual*. [Online]. Available: <https://gevas.eu/en/portfolio-items/crossig/?portfolioCats=135>
- [20] GEVAS Software GmbH, Munich, Germany. (2018). *Open TRELAN 5.0 User Manual*. [Online]. Available: <https://gevas.eu/en/portfolio-items/crossig/?portfolioCats=135>
- [21] GEVAS Software GmbH, Munich, Germany. (2018). *Adaptive Network Control BALANCE User Manual*. [Online]. Available: <http://vision-traffic.ptvgroup.com/en-us/products/ptv-balance/>
- [22] Q. Ye, W. Y. Szeto, and S. C. Wong, "Short-term traffic speed forecasting based on data recorded at irregular intervals," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1727–1737, Dec. 2012.
- [23] H. V. Ravinder, "Determining the optimal values of exponential smoothing constants—does solver really work?" *Amer. J. Bus. Educ.*, vol. 9, no. 1, p. 93, 2016.
- [24] S. Chopra and P. Meindl, *Supply Chain Management: Strategy, Planning, and Operation*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2007.
- [25] D. M. Levinson, "Speed and delay on signalized arterials," *J. Transp. Eng.*, vol. 124, no. 3, pp. 258–263, 1998.
- [26] *MATLAB Release 2018b*, MathWorks Inc., Natick, MA, USA, 2018.
- [27] D. Afshartous and R. A. Preston, "Key results of interaction models with centering," *J. Statist. Educ.*, vol. 19, no. 3, pp. 1–24, 2011.
- [28] *German Highway Capacity Manual (HBS)*, Road and Transportation Research Association (FGSV) Std., Cologne, Germany, 2009.
- [29] *Highway Capacity Manual*, Transportation Research Board, National Research Council Std., Washington, DC, USA, 2010.



**Salil Sharma** received the bachelor's degree in civil engineering from IIT Guwahati in 2010 and the master's degree in transportation systems from the Technical University of Munich, Germany, in 2016. His research interests include traffic management, transport big data, and intelligent transportation systems.



**Jonas Lüßmann** received the Diploma degree in civil engineering and the Ph.D. degree in traffic engineering from Technische Universität München. He is currently a Project Leader and the Product Manager of the traffic engineers workstation CROSSIG with GEVAS Software GmbH, Munich, Germany. His research interests include traffic signal control and cooperative systems.



**Jaehyun (Jason) So** received the B.S. degree and the M.S. degree in transportation engineering from Ajou University, South Korea, in 2006 and 2008, respectively, and the Ph.D. degree in civil and environmental engineering from the University of Virginia, VA, USA, in 2013. He is currently an Associate Research Fellow with The Korea Transport Institute (KOTI). Before joining KOTI, he was a Research Scientist and a Lecturer with Florida Atlantic University, FL, USA, from 2013 to 2014, and with the Technical University of Munich, Germany, from 2014 to 2016. His research interests include traffic operations, safety, and connected-automated vehicles. He is involved in several R&D programs, including construction of automated vehicles test-bed in South Korea (K-City), high-speed V2X technology and use cases for connected-automated driving, and smart mobility systems toward Smart City.