UNIVERSITY OF ZAGREB

FACULTY OF TRANSPORT AND TRAFFIC SCIENCES

DEPARTMENT OF INTELLIGENT TRANSPORTATION SYSTEMS

Variable speed limit control on motorways with a high share of truck traffic

Internee

Helena Tanoue Vizioli

Supervisors

Edouard Ivanjko

André Cunha

Final Report – FPZ-ZITS-01-2020

Zagreb - Croatia

July/2020

Contents

1. Introduction

This technical report resulted from the graduate student Helena Vizioli's traineeship in the Department of Intelligent Transportation Systems (ITS), Faculty of Transport and Traffic Sciences, University of Zagreb. This research aims to implement Variable Speed Limit Control (VSLC) on highways with a high share of truck traffic. For this purpose, the traffic conditions of one Brazilian highway segment will be simulated using the microscopic simulator VISSIM.

VISSIM is the traffic simulation software that is going to be used for this research. It simulates modes of traffic and their interaction, and it will be used in conjunction with Python and MATLAB. Python is a straightforward object-oriented and interpreter-based programming language that presents high portability. In turn, MATLAB is a high-performance framework for technical computing that integrates computation, visualization, and programming. In this report, the basis of the mentioned tools and their connection principles will be presented.

To implement the VSLC in VISSIM, it is necessary to first evaluate the congested values for traffic flow and speed, using flow, speed, and density data. Once these values are determined, it is possible to define the threshold values for change of speed limit values and implement a first simple rule-based speed limit controller.

Another important step to simulate a highway segment in VISSIM is to calibrate the simulation model's parameters. VISSIM has default parameters that do not always fit the highway segment studied; thus, it is necessary to alter them. To do this, one must understand the VISSIM COM interface to integrate Python or MATLAB with VISSIM and input traffic data or obtain the results. Based on the output data, it is possible to change parameters to get a simulation as close to reality (measured traffic data) as possible.

To obtain an adequate model in VISSIM, i.e., a model similar to reality, it is necessary to change some default parameters and calibrate de model. In the case of this research, it is also possible to change the vehicle flow on the on- and off-ramps since these values were not collected. Considering the mentioned aspects, this report presents the vehicle volume estimation on the on- and off-ramps, the new model of trucks created to represent the Brazilian vehicle fleet, and the new parameters for the driver behavior, i.e., car-following and lane change. The model's adequacy is then analyzed using the GEH statistics, comparing the results of the simulation with the collected real-world data.

The VSLC is a possible solution for reducing traffic congestion, creating a more homogenous traffic flow and speed distribution. To implement it to the model, two methods for VSLC will be studied: the rule-based and the Fuzzy-Logic-based VSLC. A rule-based system consists of a system that involves curated rules, i.e., does not use an automatic rule inference. Fuzzy Logic, in turn, is a computational paradigm that is based on human reasoning, i.e., evaluating the world in imprecise terms and responding with precise actions.

This report is organized as follows. After the first introductory section, the second section describes the applied methodology. In the following third section, traffic problems present on Brazilian highways are elaborated. The fourth section presents the available traffic data. The fifth section describes the calibration of the simulation model used in the sixth section to test the implemented two VSLC methods. The last seventh section ends this report with a conclusion and future work proposals.

2. Methodology

VISSIM is the traffic simulation software that is being used for this research. It simulates different modes of traffic and their interaction, and it will be used in conjunction with MATLAB. MATLAB is a high-performance framework for technical computing that integrates computation, visualization, and programming. In this report, the basis of the mentioned tools and their connection principles will be presented.

2.1. VISSIM

VISSIM is a microscopic multimodal flow simulation software released in 1992 by PTV Planung Transport Verkehr AG in Karlsruhe, Germany [13]. This program provides realistic modeling for multimodal transport operations, enabling evaluation and planning of urban and extra-urban transport infrastructure ahead of its implementation. Since experimenting in the real world is expensive due to a range of potential costs, the use of VISSIM is increasing worldwide, especially in the public sector, consulting firms, and universities.

Before starting using VISSIM, it is fundamental to understand the graphic interface (Fig. 1) and its most important tools [12, 19]. The two main tools are the network editor and network objects. The first one is used to set up the network, in which the highway segment of the study will be displayed. The second one, in turn, contains the list of the elements used to create the traffic scenario, such as links, traffic signals, and vehicle parameters. Also, the quick view and the lists are two other tools that can help control the simulation. The first allows the quick edition of attributes of a selected object, and the other is used to edit all data, including non-graphic network objects.

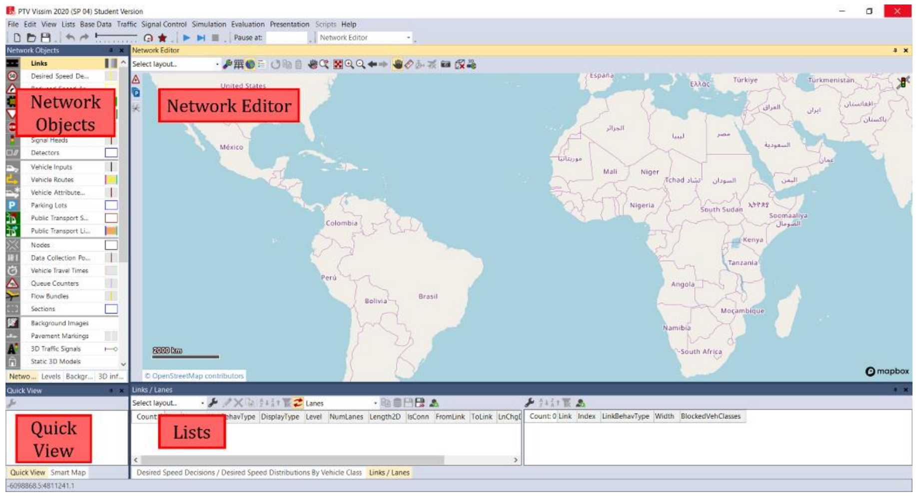


Figure 1 - VISSIM Graphical User Interface.

Using the mentioned and the other available tools, one can design a road network segment in VISSIM, input traffic data to the simulator, determine the routing decisions, and modify the model parameters. To design the segment of study in VISSIM, a background map was used in conjunction with the network objects, using the "Background Images" tool, and adding links and connectors. To replicate the traffic conditions of a highway segment, it is essential to import the collected traffic data related to the section into the simulator, which is done by creating as many time intervals as needed with the "Time Intervals" tool and relate each time interval to a volume of vehicles, with the tool "Vehicle Inputs." Regarding the vehicle routes, they are not automatically defined by VISSIM, meaning that some parameters must be input using the "Vehicle Routes" tool. Finally, to create an accurate simulation in VISSIM, it is necessary to configure particular parameters of the model using the "Parameters" section.

After the model creation and calibration, it is possible to run the simulation and collect the desired data. It is possible to run the simulation continuously or step by step to evaluate each frame, and the data that results from the simulation can be collected on the software or by using another program, such as MATLAB. For that, it is necessary to create data collection points with the "Data Collection Point" tool.

2.2. MATLAB

Matrix Laboratory (MATLAB) is a programming platform that uses a matrix-based language to solve engineering and scientific problems simply and efficiently. Its primary uses are oriented to analyze data, develop algorithms, and create models and applications. In this context, it can be a useful tool for this research, especially to implement codes or scripts to calibrate VISSIM models, create the VSLC system, and analyze the data generated by the VISSIM simulation. Moreover, regarding the data analysis, MATLAB enables the disposition of data into graphics objects, which allows a more straightforward general evaluation of results. For these reasons, MATLAB will be studied as a possible platform to be combined with VISSIM to achieve the research goals.

The first step to start using MATLAB is to understand its interface and the most important tools. The graphical user interface can be divided into four main sections, which are "File Directory," "Script Editor," "Command Window," and "Workspace," presented in Fig. 2. The "File Directory" window shows in which folder MATLAB is searching for files to run and functions to call. Fundamentally, the current script must be saved in this same folder to run the script's code with no problems. Using the "Script Editor" (Fig. 3), it is possible to write and edit codes using various functions and, on the "Command Window," the user can enter and run individual lines of the code to verify its operation. Finally, all variables and their respective values or sizes are displayed on the "Workplace." Once familiarized with the interface, one can start programming.

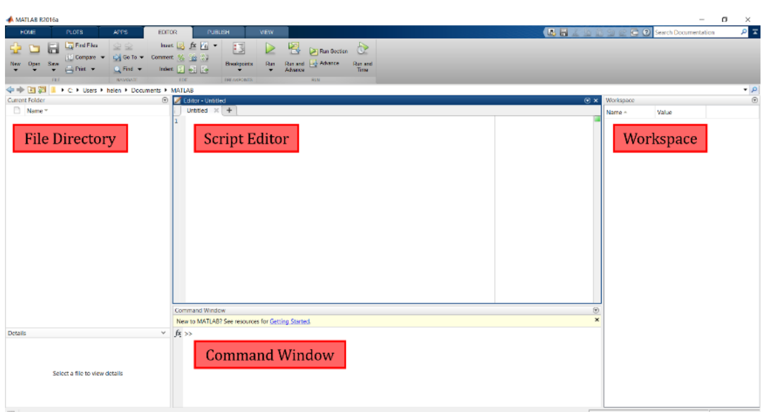


Figure 2 - MATLAB Graphical User Interface.

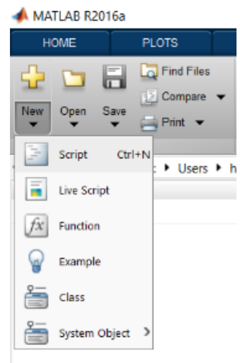


Figure 3 - Creating a new "Script Editor".

To effectively create a scrip or code, it is necessary to know MATLAB's basic operators and functions. Once getting acquainted with the basic operators and using MATLAB functions, which can be found in *"Introduction to MATLAB for Engineering Students"* [7], it is possible to create a code. In MATLAB, it is possible to load external data, create graphic elements, and communicate with other software, such as VISSIM, using the COM interface.

3. Traffic problems at Brazilian's highways

Brazil covers an area of 8.5 million square kilometers and has around 1.5 million kilometers of highways. In 2017, more than 75% of cargo transportation was performed through highways [14]. Trucks represent a high proportion of highway traffic in Brazil and are longer, heavier, and have smaller engines than the trucks used in other countries [23]. The mean presence of trucks in typical weekday traffic is 30%, with peaks of 70% early in the morning [23, 24].

This high share of trucks directly influences the traffic flow. This type of vehicle presents a lower speed limit than cars, causing a reduction in the average speed. The latter is evident in grades where trucks' performance regarding speed is slower than the one of cars [25, 26]. Other than that, the Brazilian fleet is composed of different types of trucks, being some of them very long, causing slower traffic dynamics. For example, the National Traffic Council (CONTRAN) [27] approved the 11-axle truck and up to 91 tonnes for traffic on Brazilian highways, through resolutions 640/2016 and 663/2017. However, the deliberation 172/2018 suspended the circulation of this super truck. Another point to be considered is the high level of pollution generated by trucks, especially on congested highways.

In that sense, the study of the truck fleet's impact on highway traffic is of significant importance. For that, available data from 23 measurements on four different Brazilian highways were analyzed, and one segment was selected to be simulated using microscopic simulator VISSIM.

4. Available traffic data

The data available for this research consists of volume (traffic flow) and average speed of vehicles in 23 different points of 4 highways, listed in Table 1, and was provided by ARTESP (Agência de Transporte do Estado de São Paulo). The highways encompassed (SP-280, SP-270, SP-021, and SP-348) are in the state of São Paulo, Brazil, and connect the city of São Paulo to other localities.

Table 1 - Available sensor data.

		Km	Lane
SP-280 (Rodovia Presidente Castelo Branco)	Sensor 1	15.90	W (side road)
	Sensor 2	16.08	E/W (expressway)
	Sensor 3	16.30	E (side road)
	Sensor 4	18.40	E (side road)
	Sensor 5	22.40	E/W (expressway)
	Sensor 6	22.40	E (side road)
	Sensor 7	22.40	W (side road)
	Sensor 8	26.90	E/W (expressway)
	Sensor 9	29.50	E/W (expressway)
	Sensor 10	37.00	E/W (expressway)
	Sensor 11	51.90	E/W (expressway)
	Sensor 12	59.60	E/W (expressway)
	Sensor 13	75.00	E/W (expressway)
SP-270 (Rodovia Raposo Tavares)	Sensor 14	34.00	E/W (expressway)
	Sensor 15	39.95	E/W (expressway)
SP-021 (Rodoanel Mário Covas)	Sensor 16	18.30	E/W
	Sensor 17	22.30	E/W
SP-348 (Rodovia dos Bandeirantes)	Sensor 18	32.00	N/S
	Sensor 19	49.20	N/S
	Sensor 20	65.00	N/S
	Sensor 21	86.00	N/S
	Sensor 22	131.70	N/S
	Sensor 23	149.60	N/S

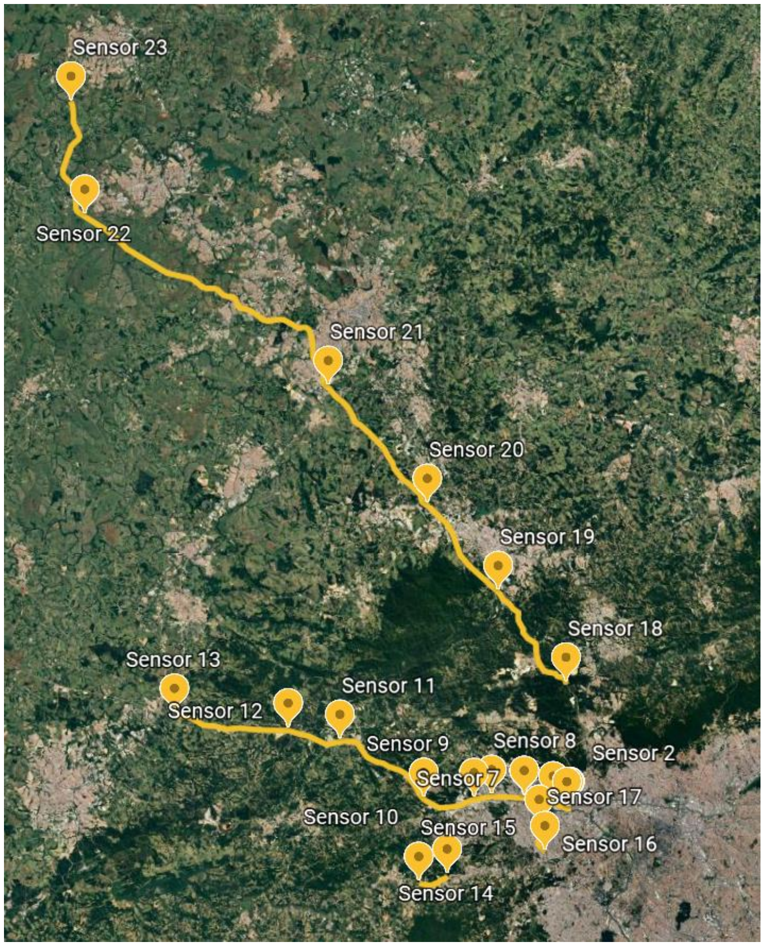


Figure 4 - Location of available sensors.

Among the 23 sensors displayed in Fig. 4, seven were selected considering the data availability and the number of on- and off-ramps. The sensors that did not contain data for both directions or details about the road geometry were excluded. The remaining sensors were analyzed considering the number of on- and off-ramps on the segments between them, choosing two segments with lower, one segment with medium, and two with a higher number of ramps. These criteria led to the sensor 2, 5, 10, 11, 20, 21, 22, and 23. The data from the selected sensors were analyzed regarding traffic flow, density, and average speed to select a single study segment. The segments with better geometry and with at least one problematic date regarding traffic flow were selected. In the end, the segment between sensors 20 and 21 (Figs. 5 and 6) was selected to be the study case.

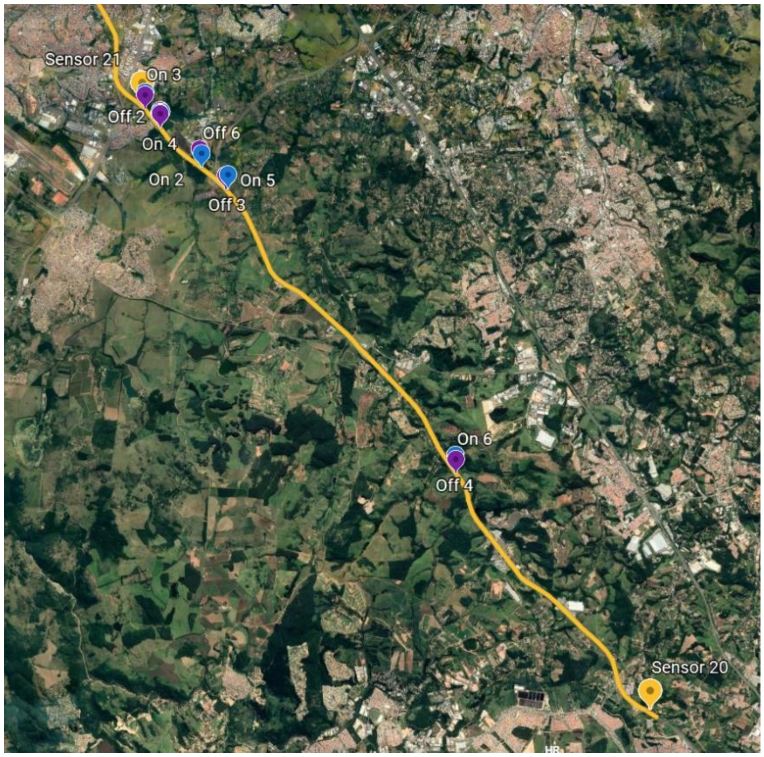


Figure 5 - Sensors 20 and 21 and corresponding on- and off-ramps on the segment.

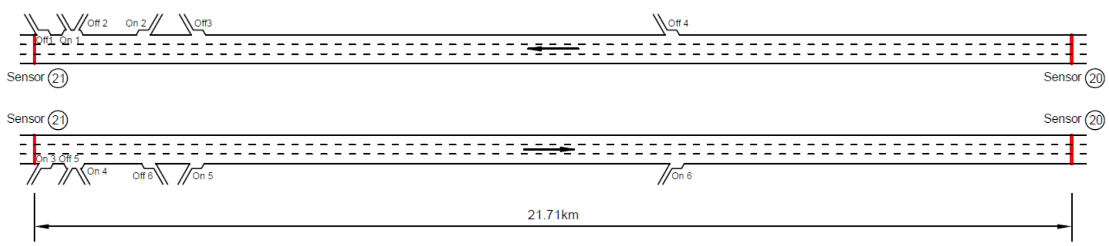


Figure 6 – Highway segment between sensors 20 and 21.

For this segment, an overview of March and April 2014 data is shown in Figs. 7 and 8, regarding traffic directions from south to north and north to south, respectively.

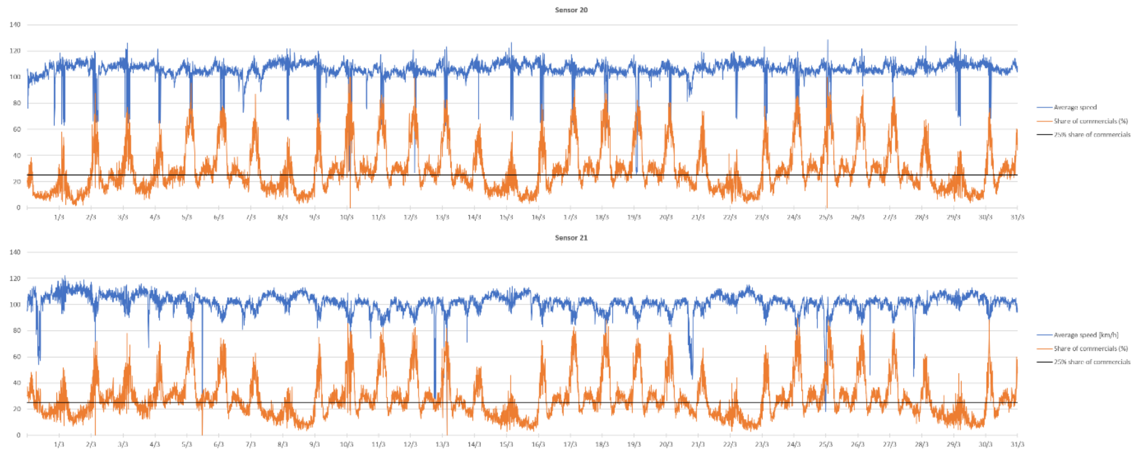


Figure 7 - Average speed and share of commercials on sensors 20 and 21 from south to north on 03/2014.

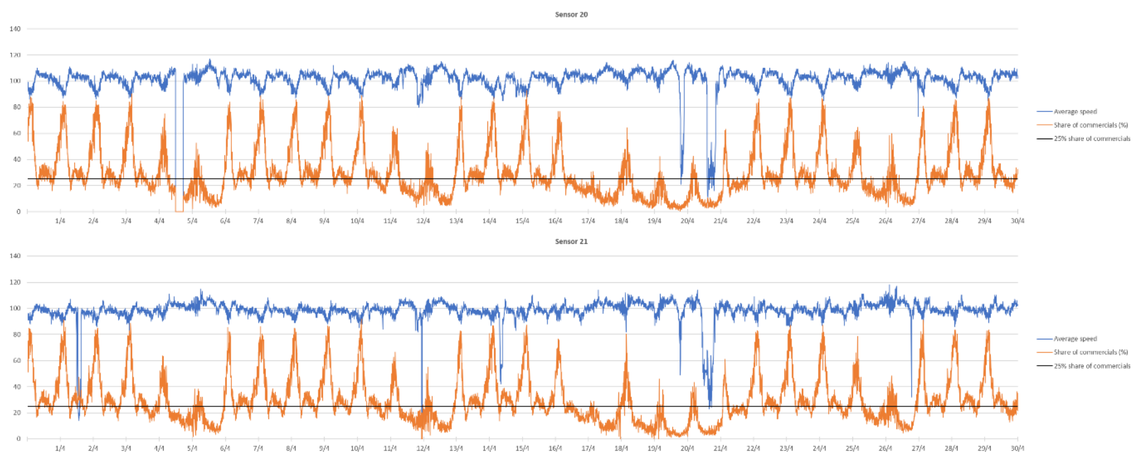


Figure 8 - Average speed and share of commercials on sensors 20 and 21 from north to south on 04/2014.

The share of commercials is high in this section, which is a fundamental parameter for the research goals. Also, there is a direct relationship between the peeks of the share of trucks and the speed drops, especially visible for sensor 20, as presented in Fig. 7. For this reason, for March 2014 in the south-north direction, data from every Tuesday, Wednesday, and Thursday were displayed in graphs, as seen in Figs. 9 to 12. The values obtained are satisfactory for the research, particularly on the days 04/03 and 11/03, because the speed drop is controlled and lasts some hours. Therefore, the data from 13:00h to 20:00h on 11/03 were analyzed, as seen in Fig. 13, since this date presents speed drops on both sensors.

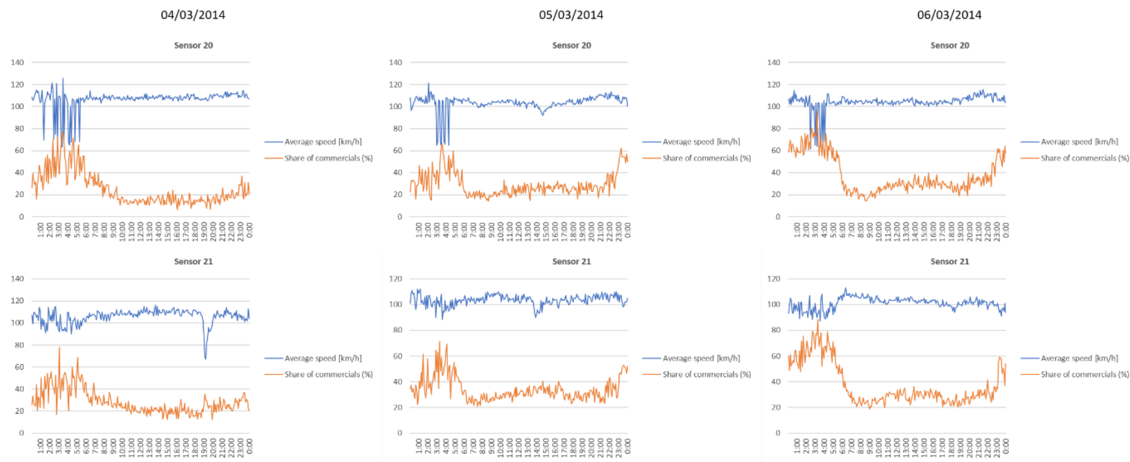


Figure 9 - Average speed and share of commercials from 04 to 06 of March.

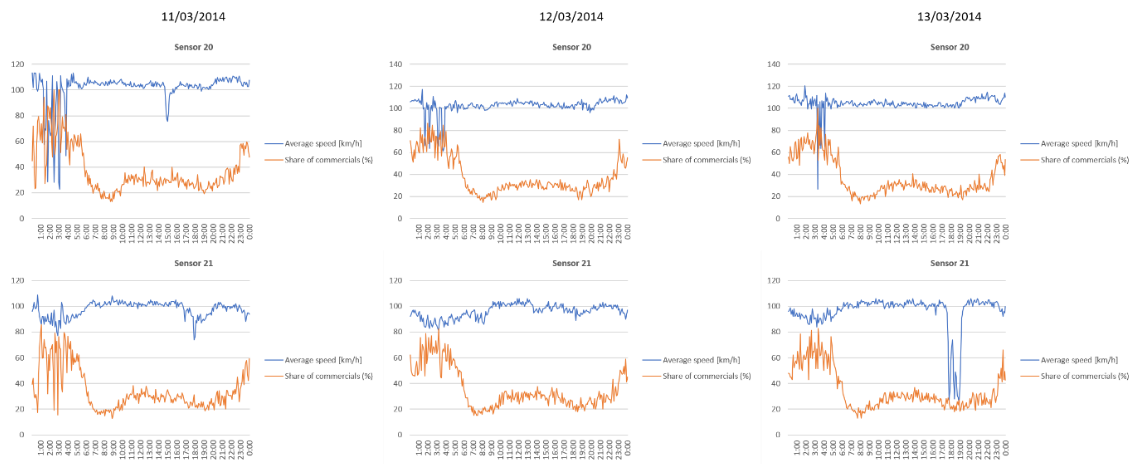


Figure 10 - Average speed and share of commercials from 11 to 13 of March.

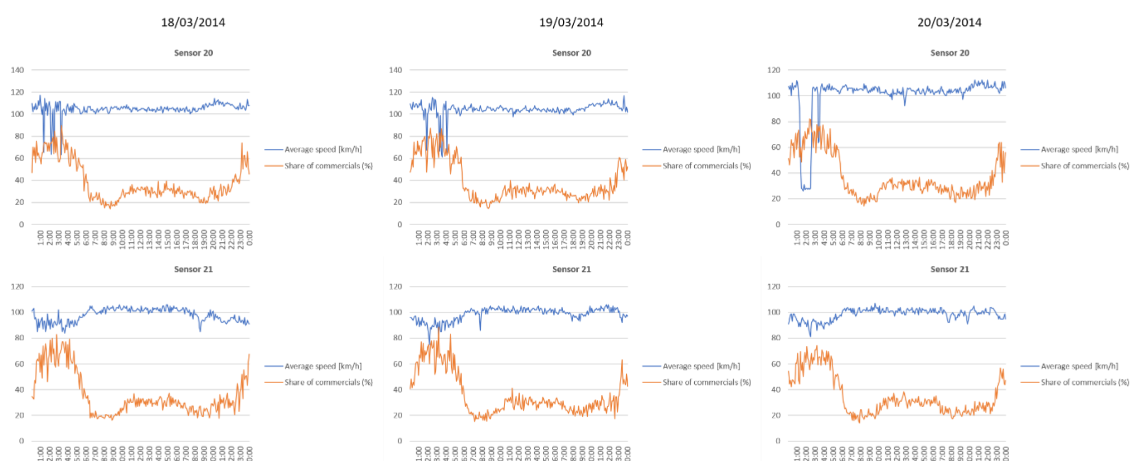


Figure 11 - Average speed and share of commercials from 18 to 20 of March.

Figure 12 - Average speed and share of commercials from 25 to 27 of March.

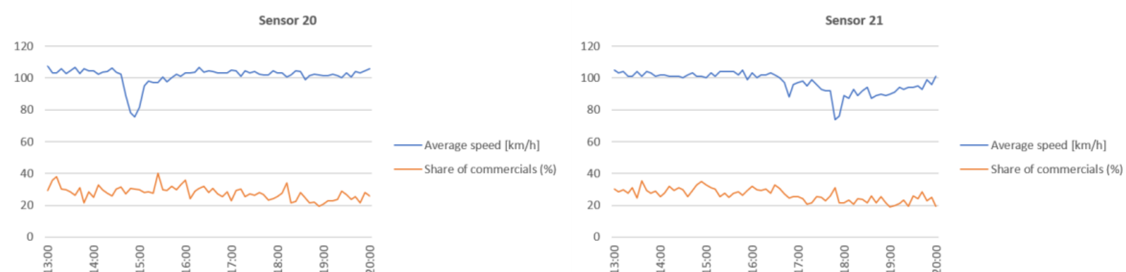


Figure 13 - Average speed and share of commercials on sensors 20 and 21 from south to north on 11/03/2014 from 13:00h to 20:00h.

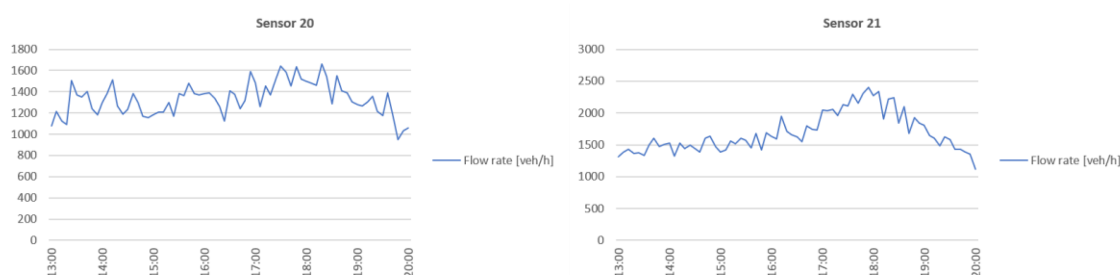


Figure 14 - Flow rate on sensors 20 and 21 from south to north on 11/03/2014 from 13:00h to 20:00h.

Finally, the segment between sensors 20 and 21 was selected to continue with the research, which will initially use the data from the 11th of March 2014. The highway segment presented in Fig. 15 allows a better comprehension of the section near sensor 21 and will be useful to the research's following steps.

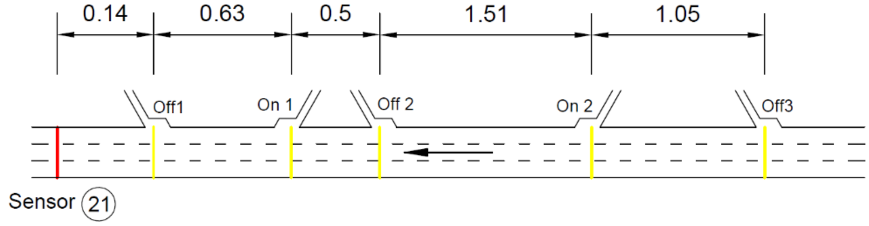


Figure 15 - Section from segment between sensors 20 and 21 around sensor 21 with area lengths in [km].

5. Calibration of the simulation model

5.1. Creating the highway segment in VISSIM

To create the selected highway segment in VISSIM, the OpenStreetMap tool was used as a background (Fig. 16). The section between sensors 20 and 21 has three lanes, while the on- and off-ramps are one or two depending on the direction (Fig. 17).

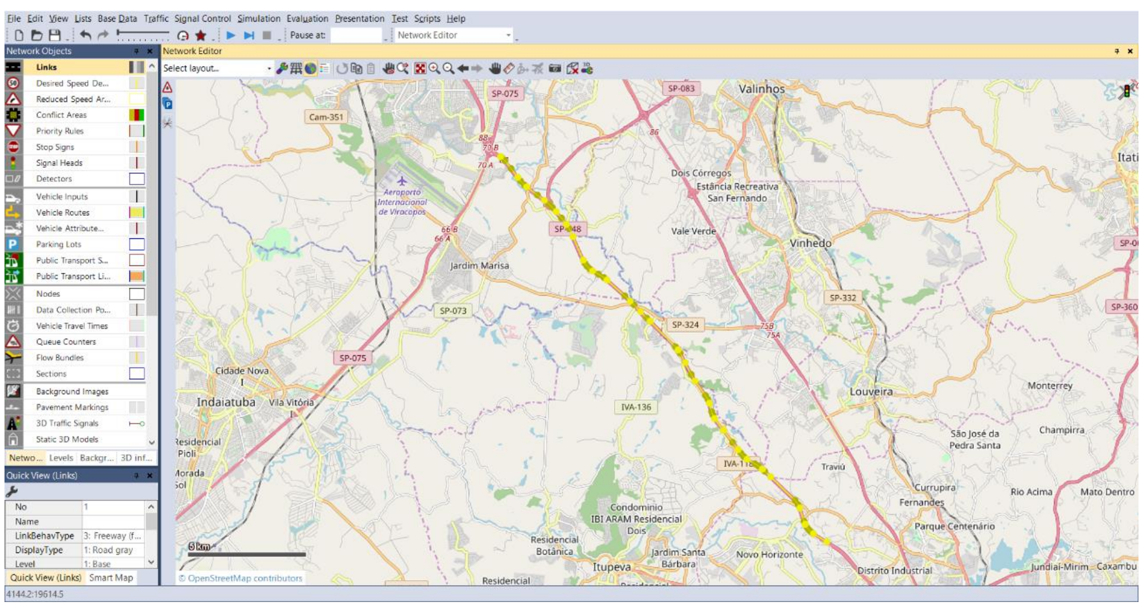


Figure 16 - Segment between sensors 20 and 21 in VISSIM.

Figure 17 - On- and off-ramps in VISSIM.

5.2. Connecting VISSIM and MATLAB

To obtain the output data from a VISSIM simulation using MATLAB, it is necessary to use the COM interface [8, 12, 15, 17, 18, 21]. For that, one has to register the COM Server in VISSIM, as presented in Fig. 18. This enables the activation of the server in MATLAB using:

```
vis = actxserver('VISSIM.vissim.200'),
```

where '200' refers to the VISSIM's version used, being 2020 the one in this case.

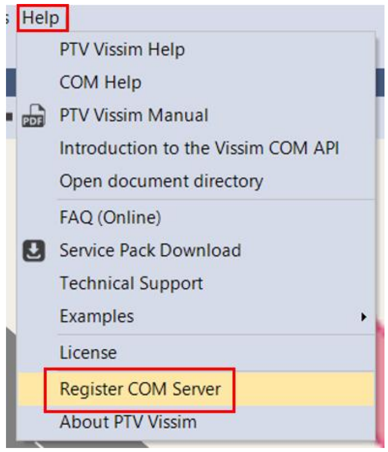


Figure 18 - Registering the COM Server.

To access the methods of a VISSIM COM object, as well as their return value types and arguments, and to obtain the object's properties, the following commands can be used:

vis.methods,

vis.invoke,

vis.fields,

where 'vis' is the VISSIM COM object, previously determined. The MATLAB responses for these commands are presented in Figs. 19 to 21.

Command Window

```
Methods for class COM.VISSIM_vissim_200:

ApplyModelTransferFile                    SaveNet
AttValue                                  SaveNetAs
BringToFront                              SetResultsFolder
CalculateVisumAssignment                  SuspendUpdateGUI
Exit                                      VissimTest
ExportVisum                               addproperty
GenerateModelTransferFile                 constructorargs
GenerateModelTransferFileBetweenFiles     delete
ImportANM                                 deleteproperty
ImportCoordinateRoutes                    events
ImportOpenDrive                           get
ImportResults                             interfaces
ImportSynchro                             invoke
LoadLayout                                load
LoadNet                                   move
LoadProject                               propedit
Log                                       release
New                                       save
PlaceUnderScenarioManagement              send
ResumeUpdateGUI                           set
SaveLayout
```

Figure 19 – MATLAB response for the vis.methods command.

Command Window

```
>> vis.invoke;
    AttValue = Variant AttValue(handle, string)
    New = void New(handle)
    LoadNet = void LoadNet(handle, Variant(Optional))
    LoadProject = void LoadProject(handle, string)
    PlaceUnderScenarioManagement = void PlaceUnderScenarioManagement(handle, string, string)
    SaveNet = void SaveNet(handle, Variant(Optional))
    SaveNetAs = void SaveNetAs(handle, Variant(Optional))
    LoadLayout = void LoadLayout(handle, Variant(Optional))
    SaveLayout = void SaveLayout(handle, Variant(Optional))
    ImportANM = void ImportANM(handle, Variant(Optional))
    ImportCoordinateRoutes = void ImportCoordinateRoutes(handle, string, bool, bool, Variant(Optional))
    BringToFront = void BringToFront(handle)
    Exit = void Exit(handle)
    ImportSynchro = void ImportSynchro(handle, string, Variant(Optional))
    ExportVisum = void ExportVisum(handle, string, Variant(Optional))
    ImportResults = void ImportResults(handle, Variant(Optional))
    VissimTest = void VissimTest(handle, int32, string, string, string)
    SuspendUpdateGUI = void SuspendUpdateGUI(handle)
    ResumeUpdateGUI = void ResumeUpdateGUI(handle, Variant(Optional))
    CalculateVisumAssignment = void CalculateVisumAssignment(handle)
    GenerateModelTransferFile = void GenerateModelTransferFile(handle, string, string, Variant(Optional)
    GenerateModelTransferFileBetweenFiles = void GenerateModelTransferFileBetweenFiles(handle, string, s
    ApplyModelTransferFile = void ApplyModelTransferFile(handle, string)
    SetResultsFolder = void SetResultsFolder(handle, string)
    Log = void Log(handle, MessagePriority, string)
    ImportOpenDrive = void ImportOpenDrive(handle, string)
```

Figure 20 - MATLAB response for the vis.invoke command.

```
Command Window
>> vis.fields

ans =

    'Net'
    'Simulation'
    'Evaluation'
    'Graphics'
    'ScenarioManagement'
    'Presentation'
    'LicenseInfo'
    'Optima'
```

Figure 21 - MATLAB response for the vis.fields command.

To set parameters and get results from a VISSIM simulation, the network must be created in VISSIM and accessed in MATLAB. It can then be loaded, as well as the layout, with the commands:

vis.LoadNet('path\file.inpx'),

vis.LoadLayout('path\file.layx'),

where the 'path' is referent to the path to access the directory where the VISSIM file is saved, and 'file' is the name of the file.

Inside the main object 'vis,' it is possible to access other objects such as 'Simulation' and 'Net' to set simulation parameters or get results. The two mentioned objects are created with:

sim = vis.Simulation,

vnet = vis.Net.

Once the object 'sim' is created, the basic parameters of the simulation such as the period, the time of a step, the random seed, and the variable decision can be defined using:

sim.set('AttValue', 'Attribute', Value),

where 'Attribute' is a parameter that belongs to the 'sim' object, as presented in Fig. 22, and 'Value' is its value.

>> Vissim - COM > ISimulation

ISimulation Attributes

Overview Collapse All

Summary

Identifier	Short name	Long name
Comment	Comment	Comment
IsRunning	IsRunning	Is running
IsSimBreakAtActive	IsSimBreakAtActive	Is simulation break at active
MesoSectsForMicroSim	MesoSectsForMicroSim	Meso - sections for microscopic simulation
NumCores	NumCores	Number of cores
NumRuns	NumRuns	Number of runs
RandSeed	RandSeed	Random seed
RandSeedIncr	RandSeedIncr	Random seed increment
RetroSync	RetroSync	Retrospective synchronization
SimBreakAt	SimBreakAt	Simulation break at
SimMode	SimMode	Simulation mode
SimPeriod	SimPeriod	Simulation period
SimRes	SimRes	Simulation resolution
SimSec	SimSec	Simulation second
SimSpeed	SimSpeed	Simulation speed
SimTmOfDay	SimTmOfDay	Simulation time (time of day)
StartDate	StartDate	Start date
StartTm	StartTm	Start time
StartTmOfDay	StartTmOfDay	Start time (time of day)
UseAllCores	UseAllCores	Use all cores
UseMaxSimSpeed	UseMaxSimSpeed	Use maximum simulation speed
VolumeIncrDynAssign	VolumeIncrDynAssign	Volume increment (dynamic assignment)

Figure 22 - Simulation attributes.

The 'Net' attributes can also be accessed, in VISSIM, in Help > COM Help. One of these parameters is the DataCollectionMeasurements, which allows one to access the measurements from the DataCollectionPoints created in VISSIM. The commands for that are:

```
datapoints = vnet.DataCollectionMeasurements,

datapoint1 = datapoints.ItemByKey(1).
```

Finally, to run the simulation, it is necessary to use the command 'RunSingleStep,' which depends on the period and time of a step previously defined. The loop for running the VISSIM simulation can be written as:

```
for i = 0 : (period*step_time)
   sim.RunSingleStep;
end
```

While the simulation is running, it is possible to obtain data collected using the DataCollectionMeasurements, such as the speed and the number of vehicles:

```
speed = datapoint1.get('AttValue','Speed(Current, Current, All)');
```

```
veh = datapoint1.get('AttValue','Vehs(Current, Current, All)');
```

Once the desired values are obtained, the simulation is easily ended with the command below:

```
vis.release;
```

The MATLAB script used in the first simulation is presented in sequence.

```
clear
clc

vis = actxserver('VISSIM.vissim.200');
vis.methods;
vis.invoke;
vis.fields;

vis.LoadNet('C:\Users\helen\Documents\Zagreb\Research\02 Adding VSLC to the Brazil
model\VISSIM\VSLC.inpx');
vis.LoadLayout('C:\Users\helen\Documents\Zagreb\Research\02 Adding VSLC to the
Brazil model\VISSIM\VSLC.layx');

sim = vis.Simulation;
vnet = vis.Net;

period = 1080;
step_time = 10;
rand_seed = 40;
variable_decision = 360;
extension = variable_decision/step_time;
sim.set('AttValue','SimPeriod',period + extension);
sim.set('AttValue','SimRes',step_time);
sim.set('AttValue','RandSeed',rand_seed);
set(vis.Graphics,'AttValue','Quickmode',1);

datapoints = vnet.DataCollectionMeasurements;
datapoint1 = datapoints.ItemByKey(1);
```

```
sum_of_speed = 0;
final_Speed = [];
sum_of_veh = 0;
num_veh = [];
sum_of_cars = 0;
num_cars = [];
control = 0;

for i = 0 : (period*step_time)
   sim.RunSingleStep;

      speed = datapoint1.get('AttValue','Speed(Current, Current, All)');
      veh = datapoint1.get('AttValue','Vehs(Current, Current, All)');
      cars = datapoint1.get('AttValue','Vehs(Current, Current, 10)');

      if isnan(speed) == 0 && speed ~= control
         sum_of_speed = sum_of_speed + speed;
         control = speed;
      end

      if rem((i/step_time)+1, variable_decision) == 0
         sum_of_veh = sum_of_veh + veh;
         sum_of_cars = sum_of_cars + cars;

         final_Speed = [final_Speed; sum_of_speed/sum_of_veh];
         num_veh = [num_veh; sum_of_veh];
         num_cars = [num_cars; sum_of_cars];

         sum_of_speed = 0;
         sum_of_veh = 0;
         sum_of_cars = 0;

      end

end

vis.release;
disp('The end')
```

5.3. Estimating on- and off-ramps flow

Aiming the flow distribution in the on- and off-ramps, it is first necessary to verify the difference of total volume of vehicles, the volume of commercials, and flow between the two sensors (Figs. 23 to 25).

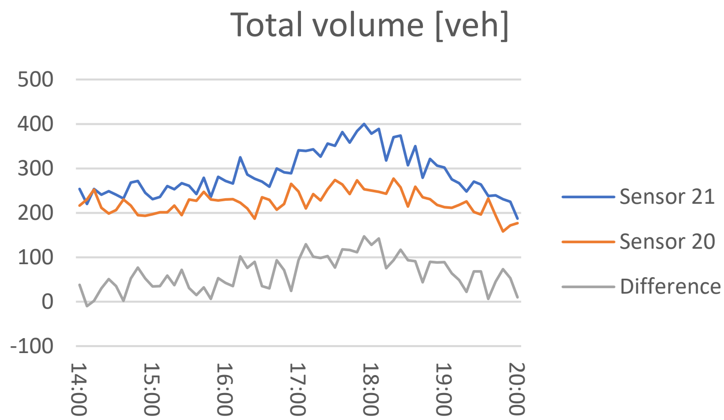


Figure 23 - Volume difference of vehicles on 11/03/2014.

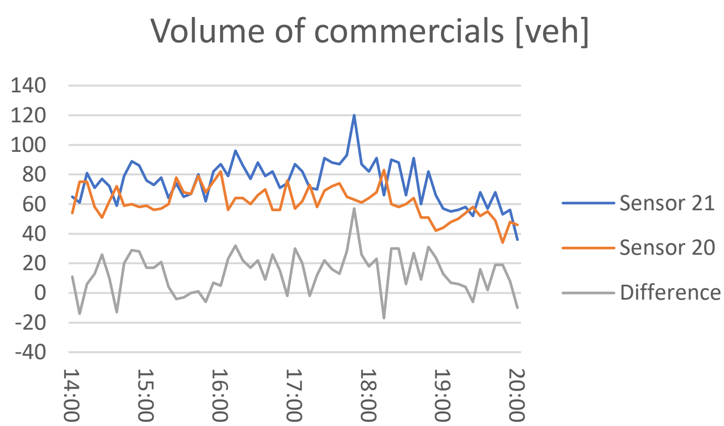


Figure 24 - Volume difference of commercials on 11/03/2014.

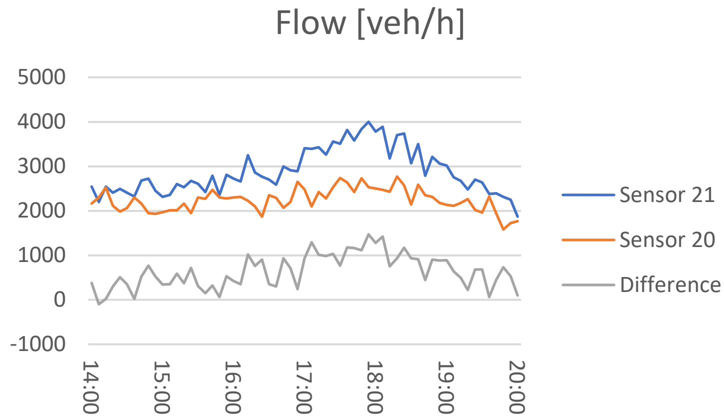


Figure 25 - Flows on 11/03/2014.

Since the traffic volume difference must be divided between two on-ramps and four off-ramps (Fig. 26), the areas accessed by them were analyzed to allow a better volume distribution. The off-ramp Off1 leads to a mostly residential area, but with a bus company on the way, which can indicate a higher number of commercial vehicles leaving the highway. The on-ramp On1 is an on-ramp as a return from the other direction (from sensor 21 to 20) and can be used by vehicles that leave the Industrial district, indicating a probable higher number of commercial vehicles. The same situation is presented for the off-ramp Off2, but to access the other direction. Both ramps On2 and Off3 are connected to a quarry and a residential area, and other highways. The off-ramp Off4 is mostly related to vehicles that come from the amusement and water parks or going to an Outlet, which can indicate a higher volume of vehicles during weekends, with some tourism buses. These characteristics are summarized in Table 2.

Table 2 – On- and off-ramps characteristics.

	Characteristics	**Possible consequence**
Off1	Residential + Bus company	Considerable bus flow
On1	Industrial district	High truck flow
Off2	Industrial district	High truck flow
On2	Residential + Quarry	Considerable truck flow
Off3	Residential + Quarry	Considerable truck flow
Off4	Parks + Outlet	High vehicle flow

The flow data available are from sensors 20 and 21, including the influence of two on-ramps and four off-ramps. The on- and off-ramps data are not at disposal, making it necessary to estimate these values. For the on-ramps, these numbers will be a percentage of the measured flow in the sensor 20 and for the off-ramps, and, for the off-ramps, they will be determined using the relative flow for the different routes, as presented in Fig. 6. The values will be adjusted based on comparing the results obtained with the simulation and the data from sensor 21. Initially, the values were based on the areas' characteristics, presented in Table 2.

Count: 5	VehRoutDec	No	Name	Formula	DestLink	DestPos	RelFlow(0)
1	1	1			2	132,250	0,300
2	1	2			3	161,692	0,300
3	1	3			6	66,235	0,300
4	1	4			9	95,083	0,300
5	1	5			5	21808,...	1,000

Figure 26 - Example of relative traffic flow for different routes.

The relative traffic flow of different types of vehicles can be determined for the vehicles that are entering the system, i.e., for sensor 20 and the on-ramps. Since both on-ramps have similar characteristics, the percentage of cars and commercials (trucks and buses) used was the same: the share of commercial vehicles between 25% and 30%, based on the available data from sensor 20. To start, the desired speed distribution and the relative flow were defined as shown in Fig. 27, and these values will be changed according to the desired flow on sensor 21 and to the calibration process demands. For the flow coming from the point where sensor 20 is located, the real traffic data were used for each time interval (from 14:00 to 20:00 in intervals of 360 seconds), as presented in Fig. 28. The same process was followed for the on-ramps, testing different percentages of the total flow on sensor 20, opting for 50% to run the first simulation. Regarding the off-ramps, the volumes are determined by the Static Vehicle Routing Decision (Figs. 29 to 31), in which each route has an associated relative flow. In total, there are ten possible routes presented in Table 3.

Count: 3	VehType	DesSpeedDistr	RelFlow
1	100: Car	120: 120 km/h	0,700
2	200: HGV	80: 80 km/h	0,200
3	300: Bus	80: 80 km/h	0,100

Figure 27 - Start values for the parameters DesSpeedDist and RelFlow.

Count: 61	Cont	TimeInt	Volume	VehComp	VolType
1	☐	0-360	2160,0	1: Default	Stocha...
2	☐	360-720	2300,0	1: Default	Stocha...
3	☐	720- 10...	2520,0	1: Default	Stocha...
4	☐	1080-1...	2110,0	1: Default	Stocha...
5	☐	1440-1...	1980,0	1: Default	Stocha...
6	☐	1800-2...	2060,0	1: Default	Stocha...
7	☐	2160-2...	2300,0	1: Default	Stocha...
8	☐	2520-2...	2160,0	1: Default	Stocha...
9	☐	2880-3...	1950,0	1: Default	Stocha...
10	☐	3240-3...	1930,0	1: Default	Stocha...
11	☐	3600-3...	1970,0	1: Default	Stocha...
12	☐	3960-4...	2010,0	1: Default	Stocha...
13	☐	4320-4...	2010,0	1: Default	Stocha...
14	☐	4680-5...	2160,0	1: Default	Stocha...
15	☐	5040-5...	1950,0	1: Default	Stocha...
16	☐	5400-5...	2300,0	1: Default	Stocha...
17	☐	5760-6...	2270,0	1: Default	Stocha...
18	☐	6120-6...	2470,0	1: Default	Stocha...
19	☐	6480-6...	2300,0	1: Default	Stocha...

ntervals | Vehicle Compositions / Relative Flows | Static Vel

Figure 28 - Flow (Volume) referent values from each time interval in sensor 20.

Table 3 - Possible routes.

Start point	Endpoint	
	Off4	1
	Off3	2
	Off2	3
	Off1	4
Sensor 20	Sensor 21	5
	Off2	6
	Off1	7
On2	Sensor 21	8
	Off1	9
On1	Sensor 21	10

Count: 5	VehRoutDec	No	Name	Formula	DestLink	DestPos	RelFlow(0)
1	1	1			2	132,250	0,300
2	1	2			3	161,692	0,300
3	1	3			6	66,235	0,300
4	1	4			9	95,083	0,300
5	1	5			5	21808,...	1,000

Figure 29 - Static Vehicle Routing Decision from Sensor 20.

Count: 3	VehRoutDec	No	Name	Formula	DestLink	DestPos	RelFlow(0)
1	2	1			6	66,113	0,300
2	2	2			9	94,999	0,300
3	2	3			5	21807,...	1,000

Figure 30 - Static Vehicle Routing Decision from on-ramp 2.

Count: 2	VehRoutDec	No	Name	Formula	DestLink	DestPos	RelFlow(0)
1	3	1			9	96,448	0,300
2	3	2			5	21809,...	1,000

Figure 31 - Static Vehicle Routing Decision from on-ramp 1.

The results obtained from the mentioned flow distribution, using 40 as a random seed, were compared to real data (Figs. 32 to 34). The GEH statistic [16] was used to compare the volume values numerically and to verify if the simulation model can accurately represent the real traffic situation. The formula used is presented below:

$$GEH = \sqrt{2 * \frac{(E-V)^2}{(E+V)}}, \qquad (1),$$

where E is the hourly traffic volume obtained from the VISSIM model, and V is referent to the real-world hourly traffic data. Moreover, GEH is measured in [$\sqrt{veh/h}$] and should be less than 5 for more than 85% of the individual links. Other than that, the simulation results can be analyzed following the classification presented in Table 4.

Table 4 - GEH classification.

GEH less than 5	Acceptable fit, probably OK.
GEH between 5 and 10	Caution: possible model error or corrupt data.
GEH greater than 10	Warning: high probability of modeling error or corrupt data.

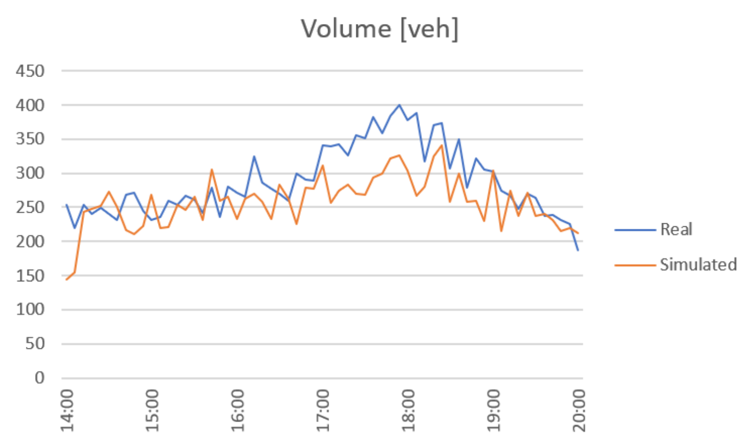


Figure 32 - Comparison of volumes of vehicles on sensor 21.

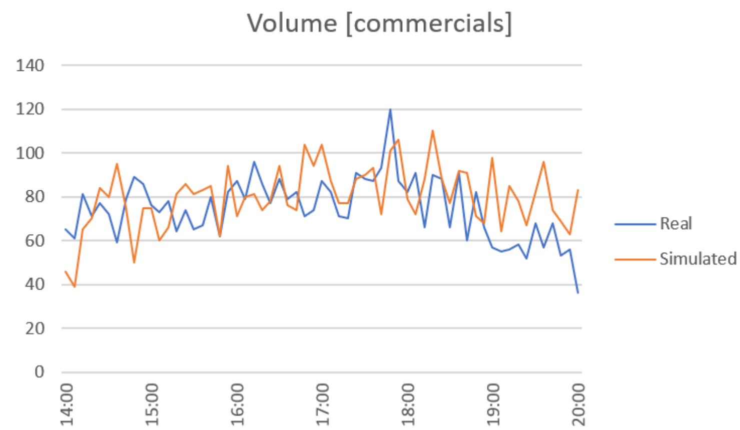


Figure 33 - Comparison of volumes of commercials on sensor 21.

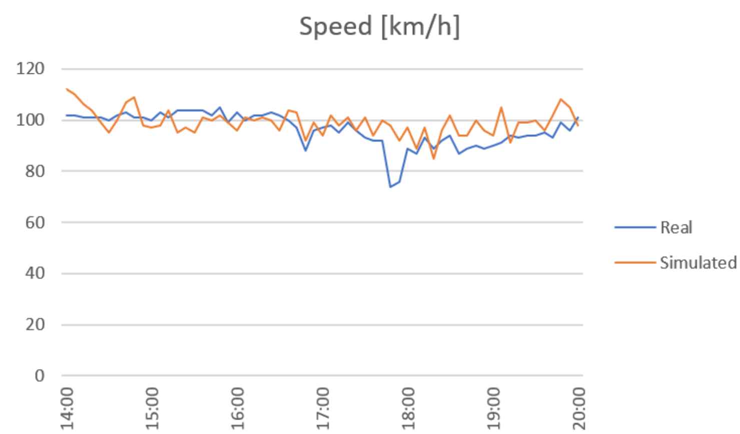


Figure 34 - Comparison of average speed on sensor 21.

The GEH statistics results are presented in Table 5, and the values indicate that the model is not an accurate representation of reality. For this reason, it is necessary to adjust it through the alteration of appropriate VISSIM parameters.

Table 5 - GEH values for sensor 21.

Hour	14:00	15:00	16:00	17:00	18:00	19:00
GEH	5.3	0.2	4.8	11.8	10.2	2.3

5.4. Truck's performance

Considering the high percentage of trucks in the analyzed segment, the truck performance parameters in VISSIM are potential influencers on the model's adequacy with reality. The default Maximum Acceleration and

Desired Acceleration functions for trucks (HGV) are presented in Figs. 35 and 36. Table 6 presents the minimum and maximum values of weight and power distributions for trucks.

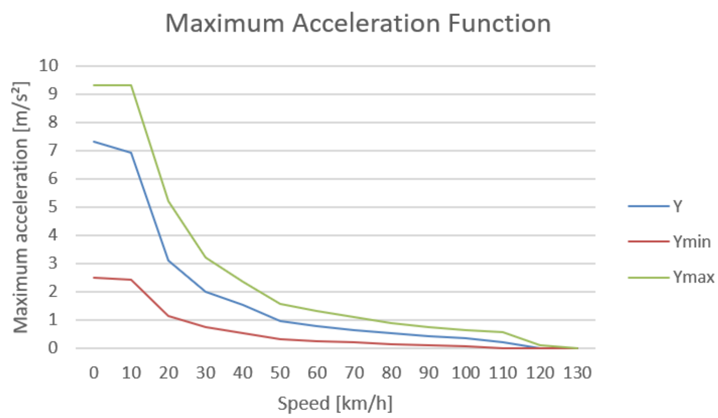


Figure 35 – VISSIM Default Maximum Acceleration Function for trucks.

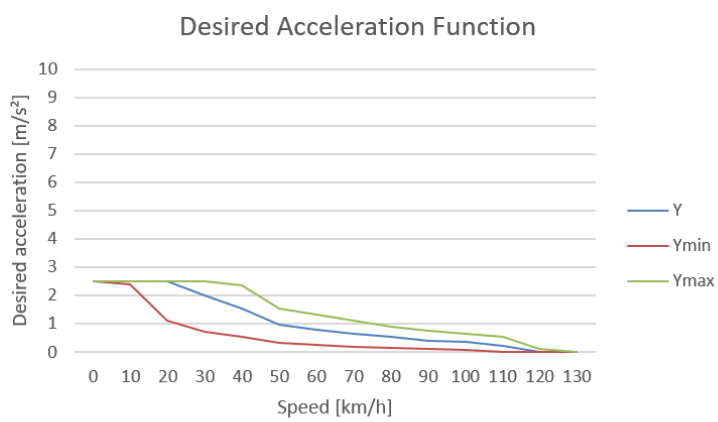


Figure 36 – VISSIM Default Desired Acceleration Function for trucks.

Table 6 - Default values for Weight and Power distributions for trucks.

	Minimum	**Maximum**
Weight Distribution	2,800 kg	40,000 kg
Power Distribution	150 kW	400 kW

However, the default values are based on the German trucks, which differ considerably from the Brazilian fleet. In Brazil, the trucks' fleet is heterogeneous, consisting of four different types of trucks [2, 6], as presented in Table 7, with the distribution displayed in Table 8.

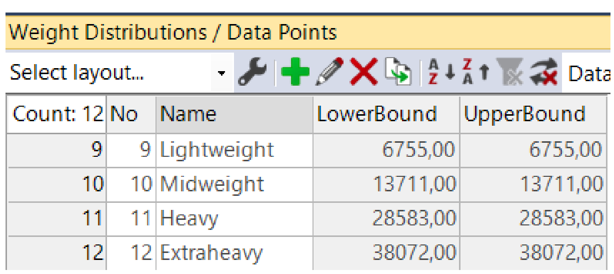
Table 7 - Trucks' categories proposed by Cunha, Modotti, and Setti (2008) [4].

Category	Axles	Average length (m)	Average GVM (t)	Power/average mass (kW/t)
Lightweight	2	6.8	6.755	14.81
Midweight	3	9.1	13.711	9.29
Heavy	4 and 5	15.5	28.583	8.79
Extra-heavy	More than 6	17.7	38.072	7.20

Table 8 - Share of trucks' types.

Category	Share [%]
Lightweight	39.60
Midweight	27.10
Heavy	19.86
Extra-heavy	13.44

Considering the characteristics of the Brazilian fleet, four new types of vehicles were created in VISSIM with adequate weight (Fig. 37), length, power (Fig. 38), and share of the volume (Fig. 39). The length, in VISSIM, depends on the 2D/3D model used, which also changes other parameters. For this reason, the same model (HGV - EU 04 Tractor.v3d) was used for the four types of trucks, but with different scales (Fig. 40), since the width and high do not considerably affect the results. The other parameters were left as VISSIM's default for HGV, and the new types of vehicles were inserted in a new class called 'Truck.'


Weight Distributions / Data Points

Select layout... Data

Count: 12	No	Name	LowerBound	UpperBound
9	9	Lightweight	6755,00	6755,00
10	10	Midweight	13711,00	13711,00
11	11	Heavy	28583,00	28583,00
12	12	Extraheavy	38072,00	38072,00

Figure 37 - Weight Distributions for new types of vehicles.

Power Distributions / Data Points

Select layout...

Count: 10	No	Name	LowerBound	UpperBound
7	7	Lightweight	100,04	100,04
8	8	Midweight	127,38	127,38
9	9	Heavy	251,24	251,24
10	10	Extraheavy	274,12	274,12

Figure 38 - Power Distributions for new types of vehicles.

Count: 6	VehType	DesSpeedDistr	RelFlow
1	100: Car	120: 120 km/h	0,700
2	300: Bus	80: 80 km/h	0,100
3	630: Lig...	80: 80 km/h	0,080
4	640: Mi...	80: 80 km/h	0,050
5	650: He...	80: 80 km/h	0,040
6	660: Ext...	80: 80 km/h	0,030

Figure 39 - Relative Flow for new types of vehicles.

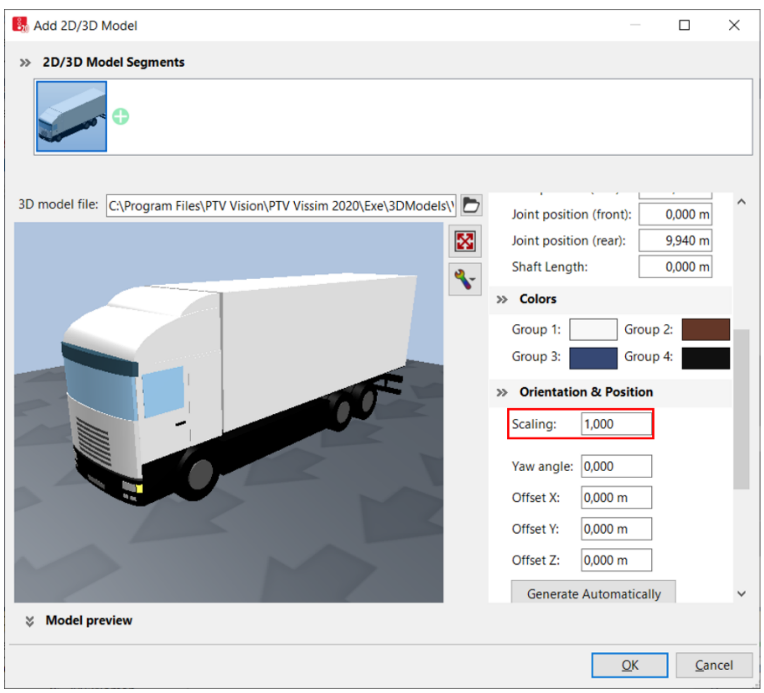


Figure 40 - Changing model scale.

The results (GEH values) obtained with this configuration are presented in Table 9. Since the changes were not significant, before following with other parameters changes, the input volume values on the on-ramps were changed for 60% of the volume on sensor 20, and the GEH values obtained are displayed in Table 10. The difference is notable, and the values are closer to 5, meaning that the simulation is closer to reality. Since more than 85% of the individual links present GEH inferior to 5, this condition is satisfied, and the model is considered adequate to reality. However, other parameters will be changed to achieve better values for the volume of commercial vehicles and the average speed.

Table 9 - GEH values for sensor 21 after changing trucks' characteristics.

Hour	14:00	15:00	16:00	17:00	18:00	19:00
GEH	5.4	0.2	4.7	11.7	10.5	2.2

Table 10 - GEH values for sensor 21 after changing the volumes on the on-ramps.

Hour	14:00	15:00	16:00	17:00	18:00	19:00
GEH	0.6	4.6	1.8	6.1	2.8	3.2

The change of trucks' length, previously done by altering the scale, was changed manually, without changing the other parameters of HGV - EU 04 Tractor.v3d, such as width and high. These values can be changing, accessing the 2D/3D Models in the Base Data option (Fig. 41).

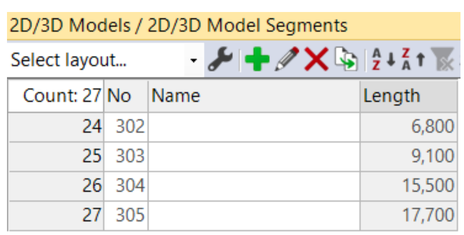


Figure 41 - Changing model length.

5.5. Existing VISSIM parameters

To facilitate the simulations, VISSIM has default rules and parameters. These characteristics can be altered but can also be used to simplify the work in the software when they are compatible with the analyzed segments.

The car-following logic used in VISSIM is the Wiedemann model [9, 20], which is based on an interactive process involving acceleration and deceleration of the vehicles. The basis is two vehicles, being one leader and one follower. The following vehicle must adapt to the speed changes of the leader, which is not a defined parameter. It means that this model consists of a constant vehicle's speed change. The parameters used to determine these changes differ from the 1974 version to the one released in 1999, being the last one more adequate to highways. The default parameters used on the Wiedemann model's latest version are nine, being CCO the standstill distance, and CC1 the headway time. The CC2, in turn, is following parameter variation, which restricts the longitudinal oscillations and regulates the safety distance variation. The CC3 is the threshold for entering the following behavior, representing the necessary time to start to decelerate before reaching the safety distance. Parameters CC4 and CC5 are the negative and positive following thresholds that control the two analyzed vehicles' acceleration difference. Finally, the parameter CC6 is the speed dependency of oscillation; the parameter CC7 is the oscillation acceleration; the parameter CC8 is the standstill acceleration, and the parameter CC9 is the desired acceleration for the speed value of 80 km/h.

Regarding the lane change model used in VISSIM, it was developed in 1978 by Sparmann [11], and it is based on finding an appropriate gap for a vehicle to change its lane. This process is ruled by three main circumstances, considering the desirability, the favorability, and the possibility of lane changing, and can be either necessary or facultative. The first has as an objective to reach a highway connector, and the second applies to cases when there is the possibility of increasing the speed by changing to a faster lane. Other parameters considered in the lane change model are:

- Waiting time before diffusion: maximum time a vehicle can wait on an emergency stop before being removed from the simulation;
- Minimum headway (front/rear): minimum distance between two vehicles necessary to the lane changing, in standstill condition;
- Safety distance reduction factor: reduces the safety distance during the lane change, multiplying the original safety distance by the reduction factor;
- Maximum deceleration for cooperative braking: maximum vehicle's deceleration to allow the lane change of a second vehicle;
- Overtake reduced speed areas: if this option is not checked in VISSIM, the vehicles do not start a lane change directly upstream

of a reduced speed area and disregard these areas on the target lane;

- Advanced merging: allows more vehicles to change lane earlier;
- Cooperative lane changing: vehicles try to switch lanes instead of just one vehicle moving to another lane;
- Maximum speed difference: is taken into account in the case of cooperative lane changing;
- Maximum collision time: is taken into account in the case of cooperative lane changing.

The most significant parameters and their default values for both models are presented in Table 11. For the Brazilian highways, the freeway configuration in VISSIM is better adjusted according to the values presented in Table 12 [4]. Apart from the parameter OBSVEH, the parameter CC1, and the parameters from D1 to D4, the values from Table 12 were applied to the model in VISSIM, obtaining the GEH values displayed in Table 13.

Table 11 - Car following and lane change parameters.

Parameter	Description	Model	Default	Minimum	Maximum
OBSVEH	Observed Vehicles	Car following	2	1	6
CC0	Standstill distance	Car following	1.50	0.50	3.00
CC1	Headway time	Car following	0.90	0.20	1.50
CC2	Following variation	Car following	4.00	0.50	8.00
CC3	Threshold for entering following	Car following	-8.00	-15.00	-2.00
CC4	Negative following threshold	Car following	-0.35	-2.00	-0.10
CC5	Positive following threshold	Car following	0.35	0.10	2.00
MH	Min. Headway - front/rear	Lane change	0.50	0.50	2.00
SD	Safety distance reduction factor	Lane change	0.60	0.10	0.60
MD	Max. Decel. For cooperative breaking	Lane change	-3.00	-9.00	-1.00
TSL	To slower lane if collision time above	Lane change	0.00	0.00	10.00

Table 12 - Statistic measurements for the best individuals.

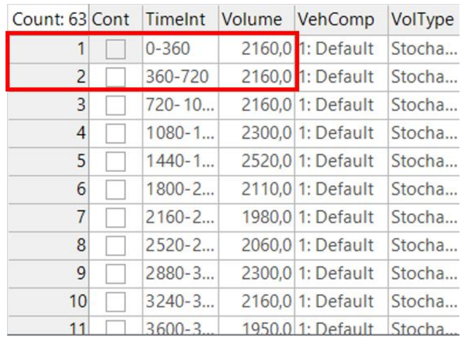
Measurement	Best	Average	Mode	1st quartile	2nd quartile	Median
OBSVEH	4	4	4	4	4	4
CC0	2.56	2.22	2.56	1.83	2.56	2.56
CC1	0.75	0.75	0.75	0.75	0.75	0.75
CC2	7.13	7.04	7.13	7.13	7.13	7.13
CC3	-12.92	-11.31	-12.92	-12.92	-8.47	-12.92
CC4	-0.54	-0.60	-0.54	-0.54	-0.54	-0.54
CC5	0.69	0.74	0.69	0.69	0.69	0.69
MH	1.75	1.54	1.75	1.51	1.75	1.75
SD	0.39	0.41	0.39	0.39	0.39	0.39
MD	-1.28	-2.48	-1.28	-3.61	-1.28	-1.28
TSL	3.11	3.93	3.11	3.11	3.25	3.11
D1	1.00	0.99	1.00	0.96	1.00	1.00
D2	0.81	0.81	0.81	0.81	0.81	0.81
D3	0.72	0.72	0.72	0.72	0.72	0.72
D4	0.60	0.60	0.60	0.60	0.60	0.60

Table 13 - GEH values for sensor 21 after changing drivers' behavior parameters.

Hour	14:00	15:00	16:00	17:00	18:00	19:00
GEH	0.5	5.1	0.9	7.5	5.5	8.1

5.6. Adding a warm-up period and slow down areas

In the pursuit of better simulation results, a warm-up period of 12 minutes was created, using the volume of vehicles of 14:00h (Fig. 42). This alteration allows the highway to have a similar number of vehicles on the road at the beginning of the selected period. Other than that, slow down areas were created around the on and off-ramps (Fig. 43), simulating the natural vehicles' movement. The GEH values obtained for this configuration are presented in Tables 14 and 15, and the comparative graphs of simulation and reality values are presented in Figs. 44 and 45. The results are not satisfactory, being necessary to change the input flow or analyze other possible parameters to alter.



Count: 63	Cont	TimeInt	Volume	VehComp	VolType
1		0-360	2160,0	1: Default	Stocha...
2		360-720	2160,0	1: Default	Stocha...
3		720- 10...	2160,0	1: Default	Stocha...
4		1080-1...	2300,0	1: Default	Stocha...
5		1440-1...	2520,0	1: Default	Stocha...
6		1800-2...	2110,0	1: Default	Stocha...
7		2160-2...	1980,0	1: Default	Stocha...
8		2520-2...	2060,0	1: Default	Stocha...
9		2880-3...	2300,0	1: Default	Stocha...
10		3240-3...	2160,0	1: Default	Stocha...
11		3600-3...	1950,0	1: Default	Stocha...

Figure 42 - Warm-up period.

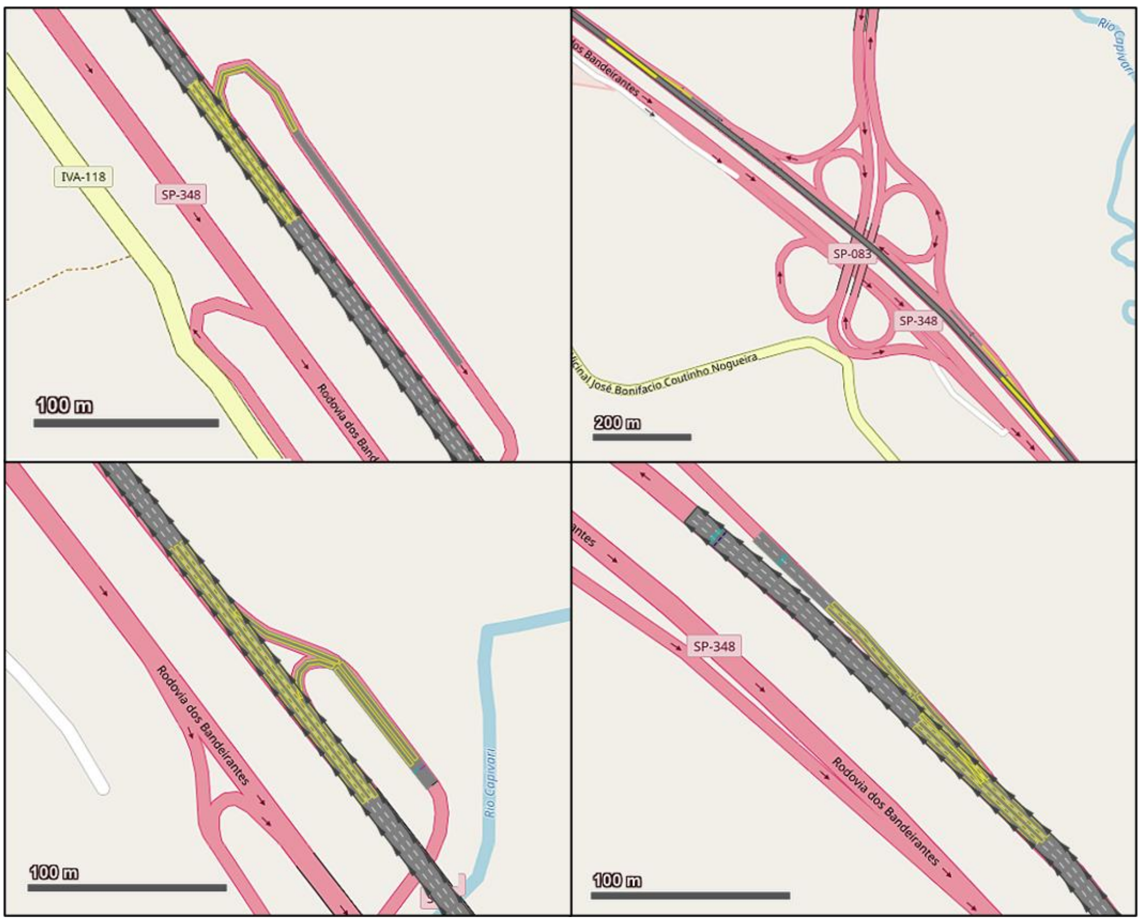


Figure 43 - Slow down areas (in yellow) near the on and off-ramps.

Table 14 - GEH values for sensor 20 with the warm-up period and slow down areas.

Hour	14:00	15:00	16:00	17:00	18:00	19:00
GEH	0.9	0.4	0.8	1.5	1.3	1.1

Table 15 - GEH values for sensor 21 with the warm-up period and slow down areas.

Hour	14:00	15:00	16:00	17:00	18:00	19:00
GEH	5.3	4.4	2.7	13.8	11.0	3.6

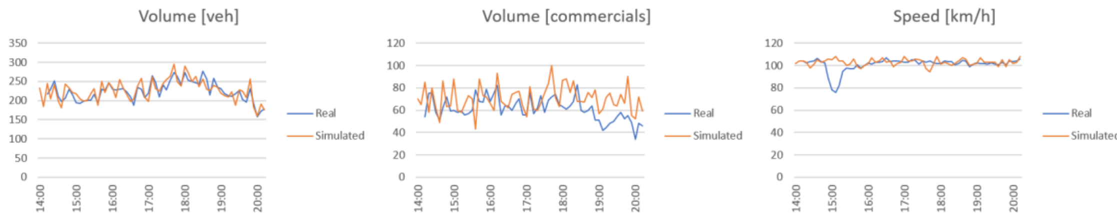


Figure 44 – Comparison of traffic volume for sensor 20.

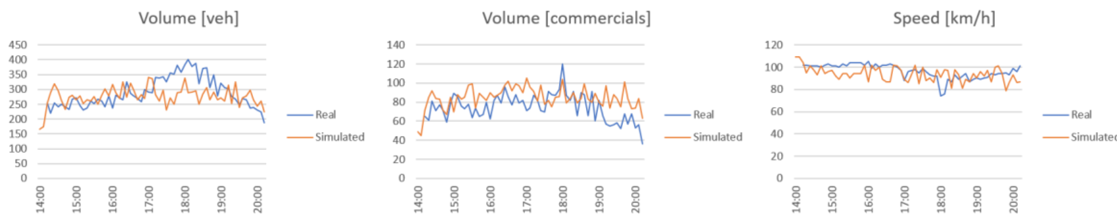


Figure 45 - Comparison of traffic volume for sensor 21.

To obtain a better result, the Vehicle Composition and the Static Vehicle Routing Decisions were modified. The new values for the composition are presented in Fig. 46. The Static Vehicle Routing Decisions, in turn, were changed per hour to obtain a model closer to reality. With the new parameters, the GEH values were satisfactory, as presented in Tables 16 and 17. The comparative graphs are shown in Figs. 47 and 48.

Count: 6	VehType	DesSpeedDistr	RelFlow
1	100: Car	120: 120 km/h	0,750
2	300: Bus	80: 80 km/h	0,100
3	630: Lightweight	80: 80 km/h	0,060
4	640: Midweight	80: 80 km/h	0,040
5	650: Heavy	80: 80 km/h	0,030
6	660: Extraheavy	80: 80 km/h	0,020

Figure 46 - New Vehicle Composition.

Table 16 - Final GEH values for sensor 20.

Hour	14:00	15:00	16:00	17:00	18:00	19:00
GEH	0.9	0.4	0.8	1.5	1.3	1.1

Table 17 - Final GEH values for sensor 21.

Hour	14:00	15:00	16:00	17:00	18:00	19:00
GEH	4.7	5.0	1.6	3.9	1.2	2.7

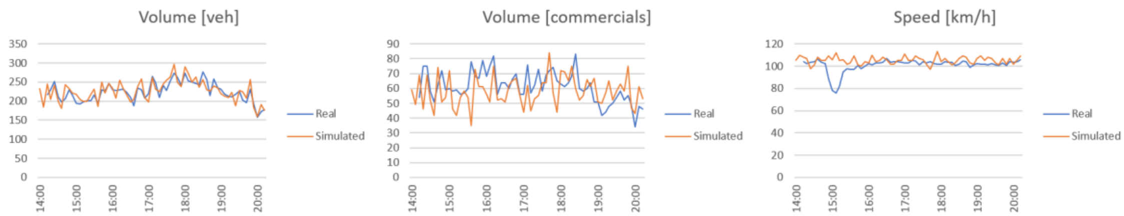


Figure 47 - Final comparison for sensor 20.

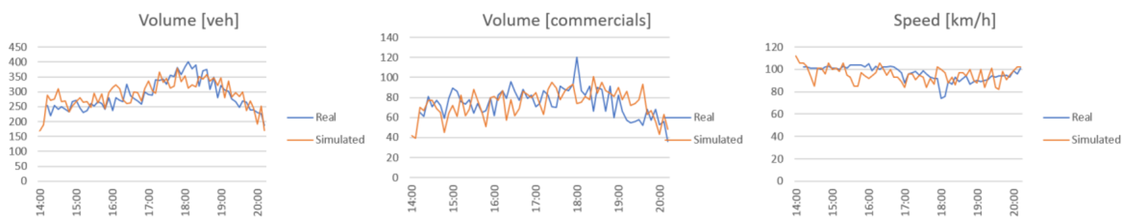


Figure 48 - Final comparison for sensor 21.

6. Application of VSLC

This section describes the traffic data analysis step prior application of VSLC to improve the traffic situation on the simulated highway segment. Two methods for VSLC are considered, as mentioned above. Only the first one (rule-based VSLC) was implemented and tested using the data analysis described in this section. The second one is described as a concept for future work.

6.1. Speed-Flow

Speed-flow data from March 2014 were plotted in point clouds (Fig. 49) to better understand the dependence between these two measured traffic parameters on the segment of the study [1]. Furthermore, attempting to get a more complete point cloud, data from 2012 to 2014 were plotted, as presented in Fig. 50. These graphs show clearly that the speed starts to drop considerably after the highway reaches around 5,000 vehicles/hour. Fig. 51 presents the adjusted curve to the point cloud, considering the congested part. The curve is a polynomial of the second degree, and it was extended to cover the speed of 100 km/h to allow a better threshold analysis, which will be exposed in chapter 7.

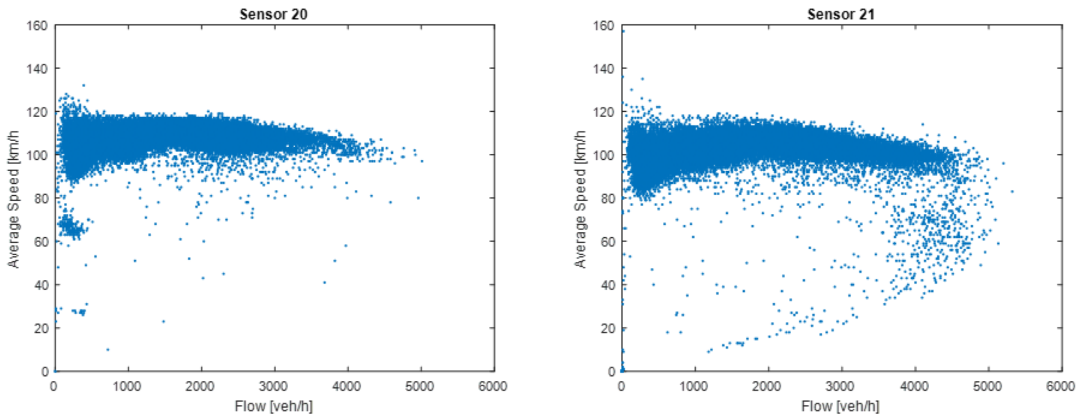


Figure 49 - Dependence between average speed and flow on sensors 20 and 21 in March 2014

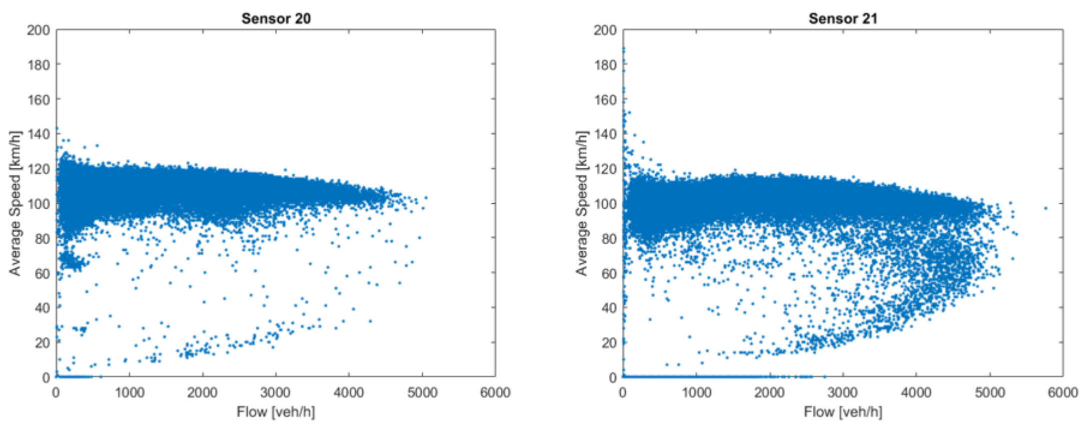


Figure 50 - Dependence between average speed and flow on sensors 20 and 21 from 2012 to 2014.

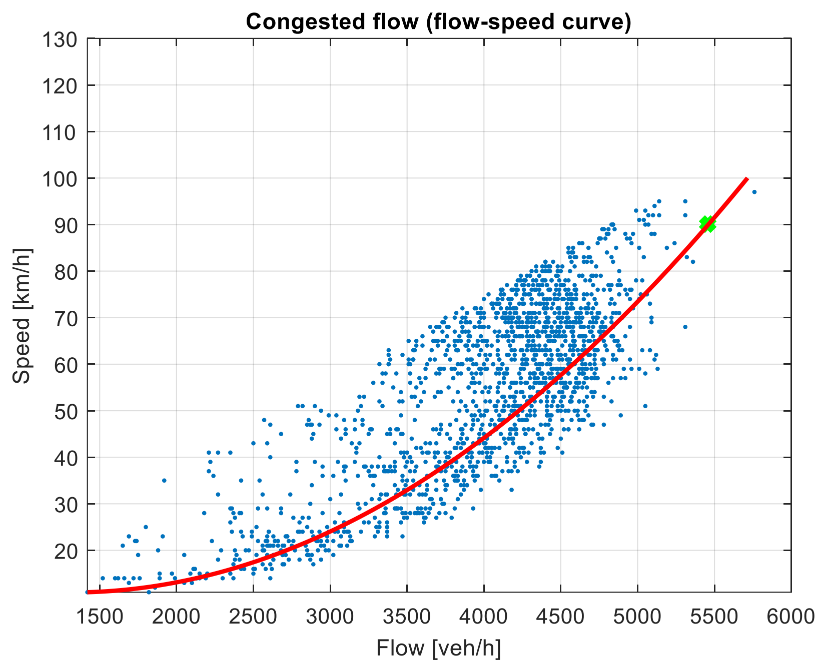


Figure 51 - Adjusted curve to the data for the congested part.

6.2. Share of commercials per lane

The average speed and the share of commercials were disposed of in graphs below (Fig. 52), for the 11/03/2014, from 14:00 to 20:00h, which will be the envisaged period for simulation. The disaggregated data for each lane were also plotted (Fig. 53) to analyze the traffic distribution. As a result, it is possible to conclude that on lane 1 (the one on the most left side), the speed is higher and the share of commercials is really low, and on lane 3 (the one on the most right side), the speed is lower due to the high share of commercial vehicles. In that sense, it is crucial to notice that the highway segment is highly heterogeneous, but the collected traffic data will be treated in an aggregated way, which may lead to differences between simulation and reality.

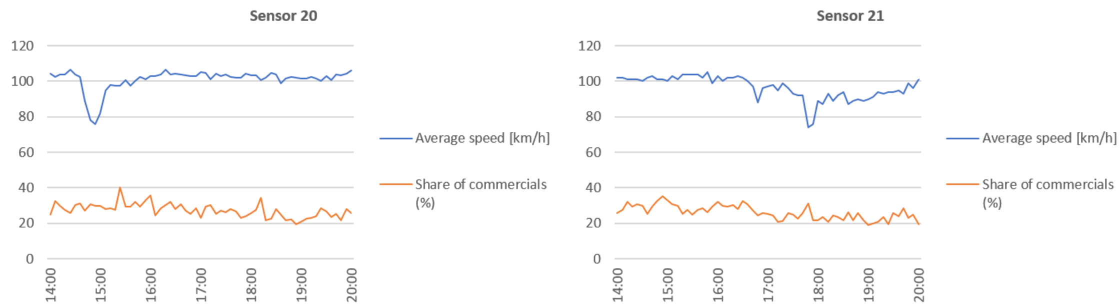


Figure 52 - Average speed and share of commercials on sensors 20 and 21 from south to north on 11/03/2014 from 14:00h to 20:00h.

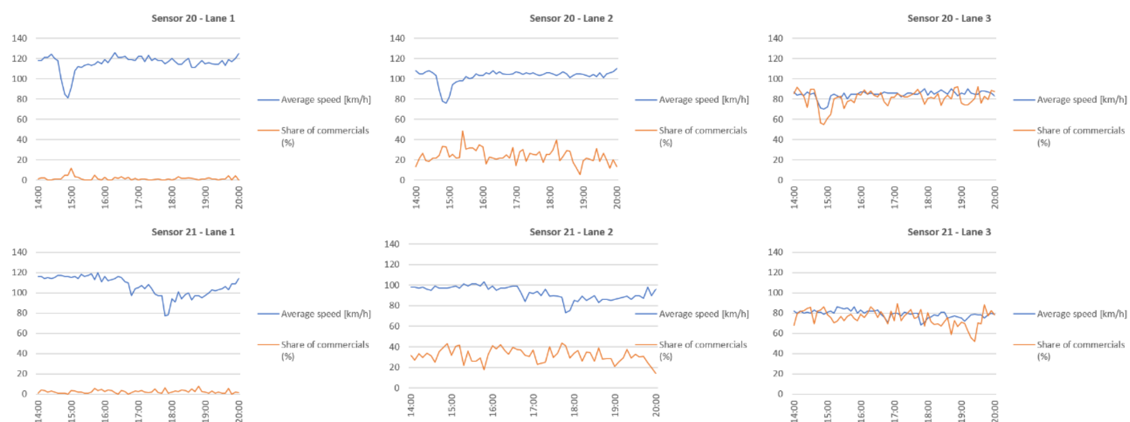


Figure 53 - Average speed and share of commercials per lane on sensors 20 and 21 from south to north on 11/03/2014 from 14:00h to 20:00h.

6.3. Flow-density diagram

To establish the parameters' limit to maintain fluid traffic, flow, and density data, with 6 minutes sampling intervals, were displayed in point clouds (Fig. 54). The following equation calculated the density:

$$Density = \frac{Flow}{Speed}, \qquad (2)$$

where *Density* is in [veh/km], *Flow* is the flow rate in [veh/h], and *Speed* is the corresponding average speed in [km/h]. The data were also plotted in boxplot to determine the outliers (Fig. 55). The least-squares fit was applied to the data of sensor 21 to determine the intersection between the crescent and the decreasing parts of the graphs (Fig. 56) and, in that way, obtain the wanted parameters, such as free-flow speed, capacity, and critical density [1].

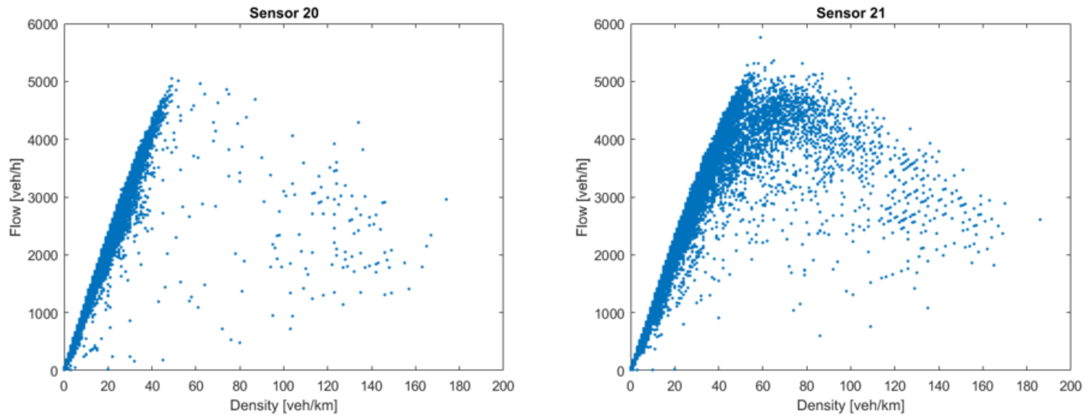


Figure 54 - Dependence between flow and density on sensors 20 and 21 from 2012 to 2014.

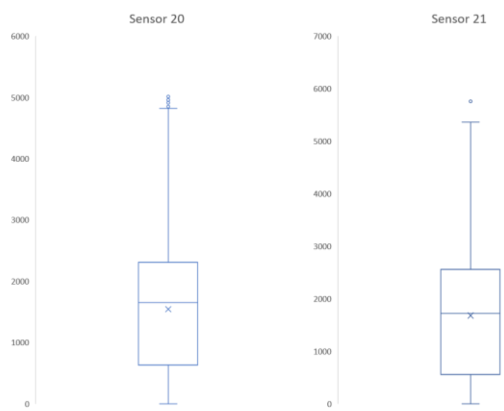


Figure 55 - Boxplots of the flow on sensors 20 and 21 from 2012 to 2014.

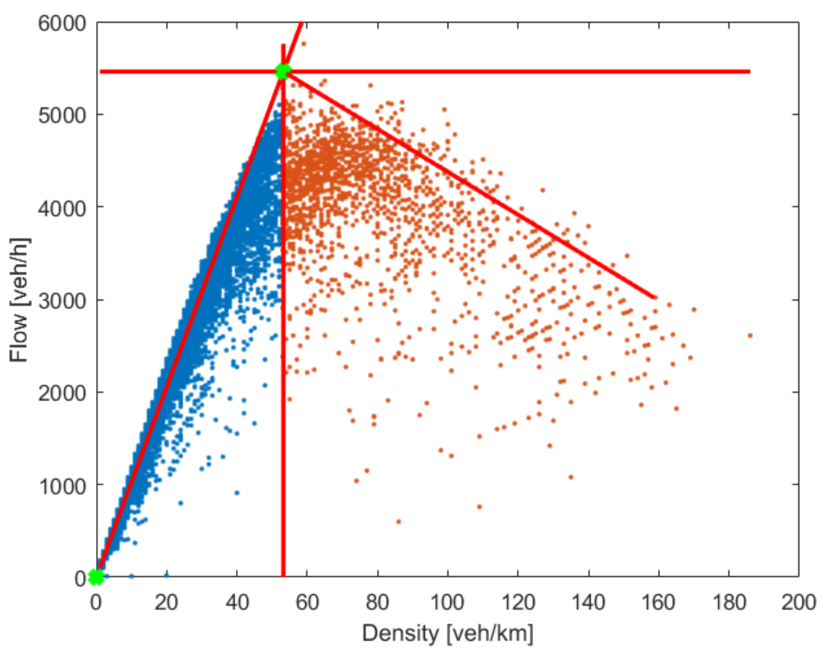


Figure 56 - Least-square fit for the flow x density point cloud.

As a result, it is possible to define the free-flow speed as being 90km/h, the capacity as 5,460 veh/h, and critical density as 53.2 veh/km, considering the three lanes.

Data from sensors 20 and 21 were also plotted, considering the flow and density per lane, which are three in total. The values were obtained, dividing the total flow and density per three and directly from the sensors (Figs. 57 and 58). These two methods allow one to understand vehicles' behavior on the highway, with more commercial vehicles on lane three and higher speeds on lane 1.

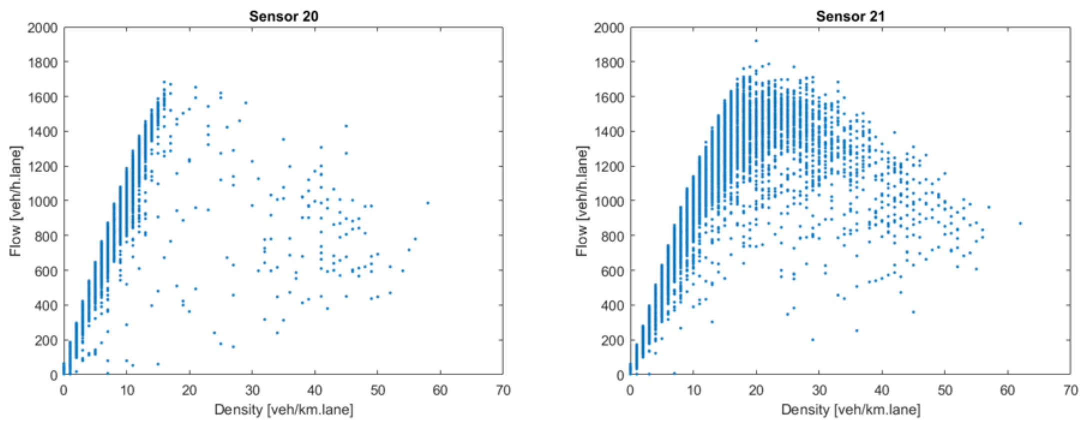


Figure 57 - Dependence between average flow and density per lane on sensors 20 and 21 from 2012 to 2014.

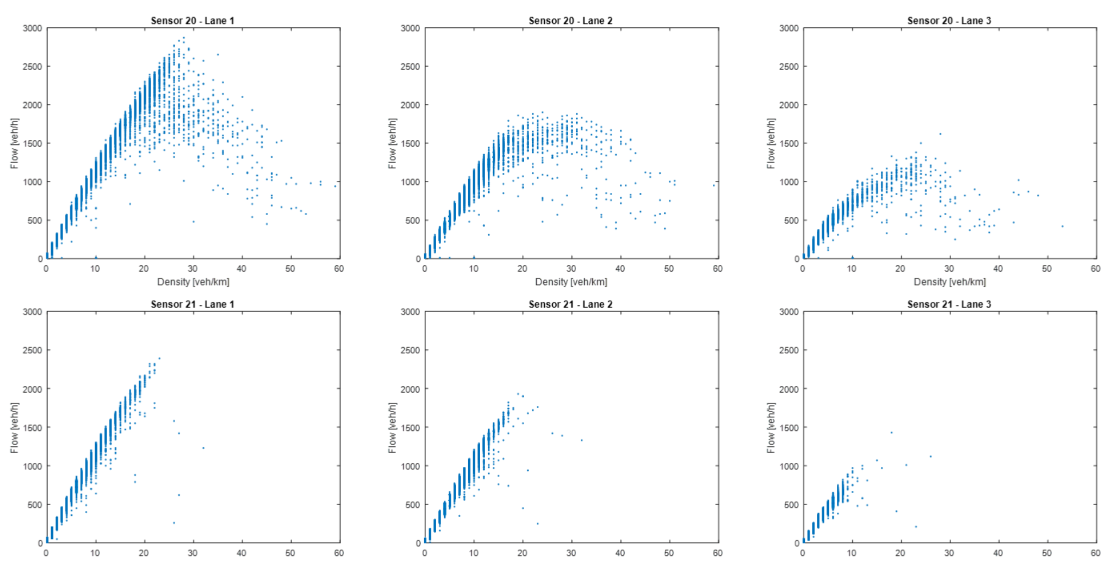


Figure 58 - Dependence between flow and density per lane on sensors 20 and 21 from 2012 to 2014.

7. Creating VSLC based on the speed-flow characteristic

7.1. Threshold values

From Fig. 51, it is possible to obtain the values of flow for a certain speed. These values were rounded (Table 18) and will be used as limits to control the speed limit, which will be reduced or increased according to the flow data obtained from the sensors. It must be noticed here that the lower part of the curvature describing the speed-flow characteristic is used. This part denotes congested traffic where the density is larger than the critical one. The characteristic behavior of smaller speeds connected with smaller flow values is visible. This presents the motivation for applying speed limit values to improve the traffic flow.

Table 18 - Threshold flow values from 50 to 100 km/h speed limits.

Speed [km/h]	Flow [veh/h]
50	4,200
60	4,600
70	4,900
80	5,200
90	5,500
100	5,800

7.2. Oscillation analysis and hysteresis setup

The flow and average speed were plotted separately for the analyzed period on 11/03/2014 to verify these traffic parameters' oscillation trough time (Figs. 59 and 60). Using MATLAB, these data were turned into curves with the leas-square fit method (Figs. 61 and 62). The adjusted curves are polynomial of third-degree. For the flow, the deviation median is around 150 veh/h, and the maximum deviations are 530 and 580 veh/h, respectively, for the points above and under the adjusted curve.

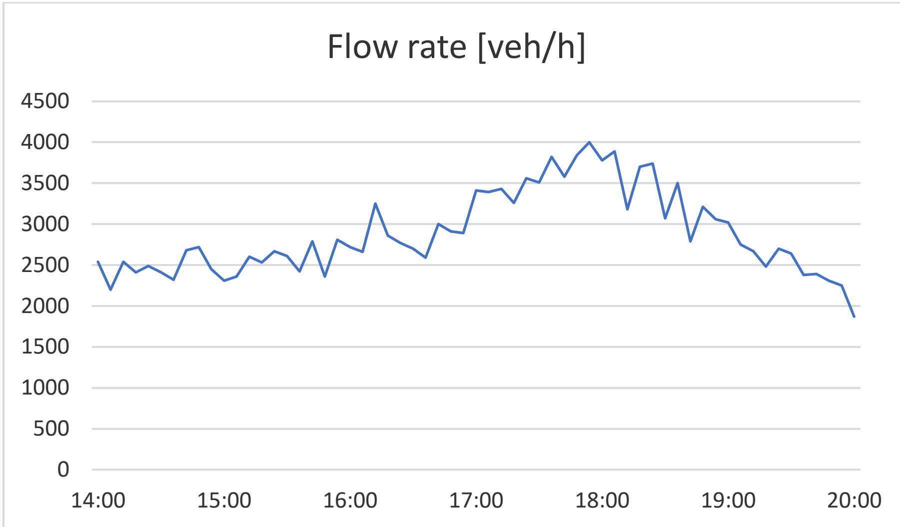


Figure 59 - Flow rate from 14:00 to 20:00 on 11/03/2014.

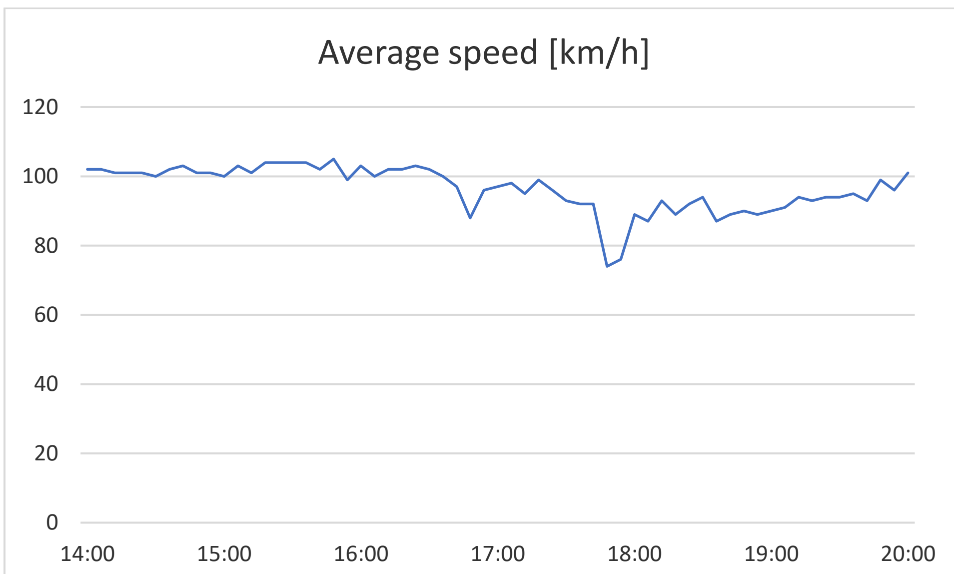


Figure 60 - Average speed from 14:00 to 20:00 on 11/03/2014.

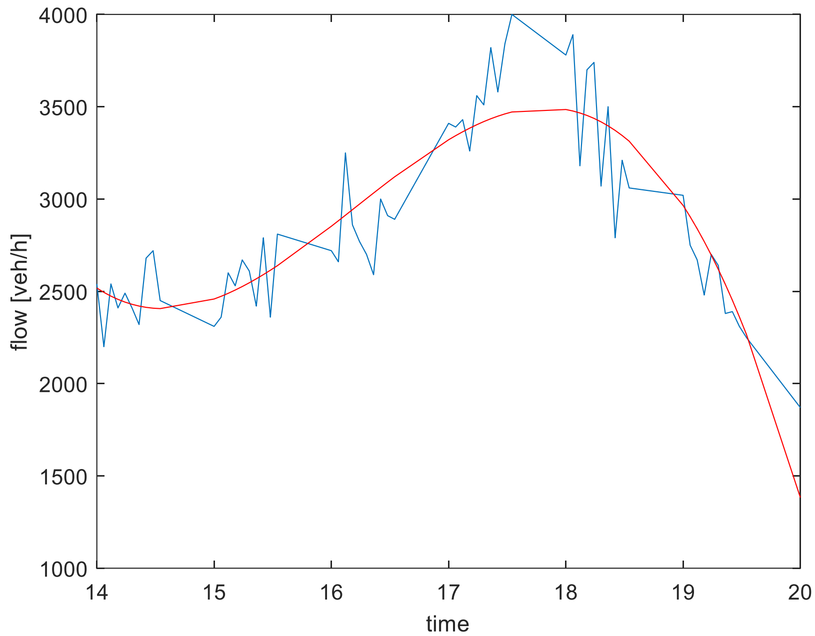


Figure 61 - Adjusted curve for flow data from 14:00 to 20:00 on 11/03/2014.

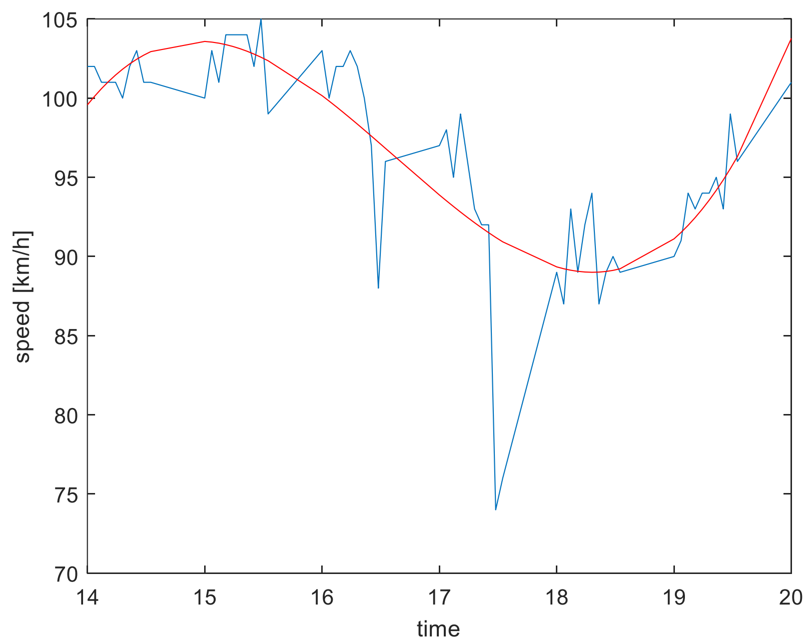


Figure 62 - Adjusted curve for speed data from 14:00 to 20:00 on 11/03/2014.

7.3. Testing using off-line sensor data

The thresholds presented in Table 18 were obtained from data from 2012 to 2014 and could not be applied to the selected date (11/03/2014).

For this reason, new thresholds (Table 19) were determined based on the data from the selected date only (Fig. 63). However, the curve presented in Fig. 63 is concave, and the wanted shape is convex. Due to that fact, the values in Table 19 were obtained with the curve in Fig. 64, which uses all data, but aims the critical point of 11/03/2014 (90km/h and 3,500 veh/h).

Table 19 - Thresholds for 11/03/2014.

Speed [km/h]	Flow [veh/h]
50	2,700
60	2,900
70	3,100
80	3,300
90	3,500
100	3,700

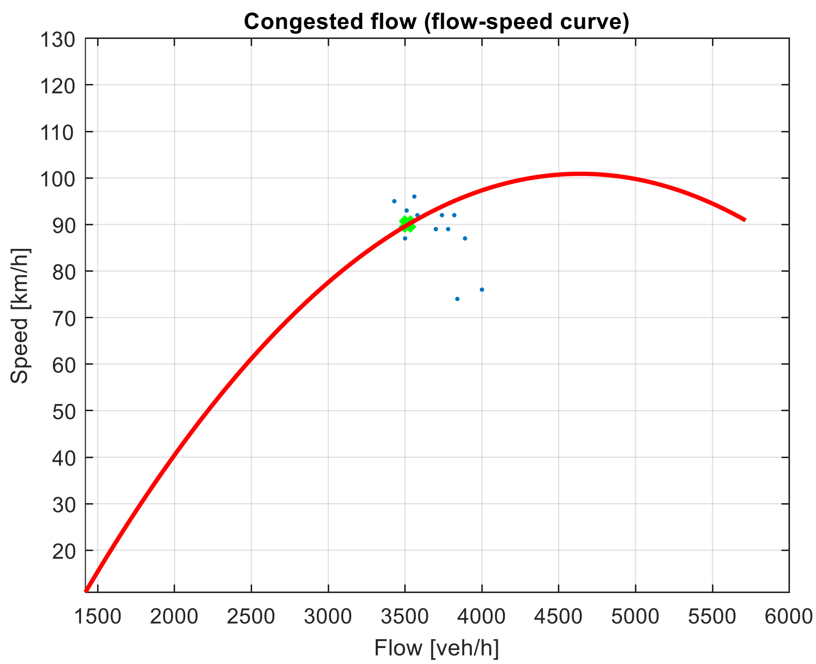


Figure 63 - Curve of congested flow on 11/03/2014.

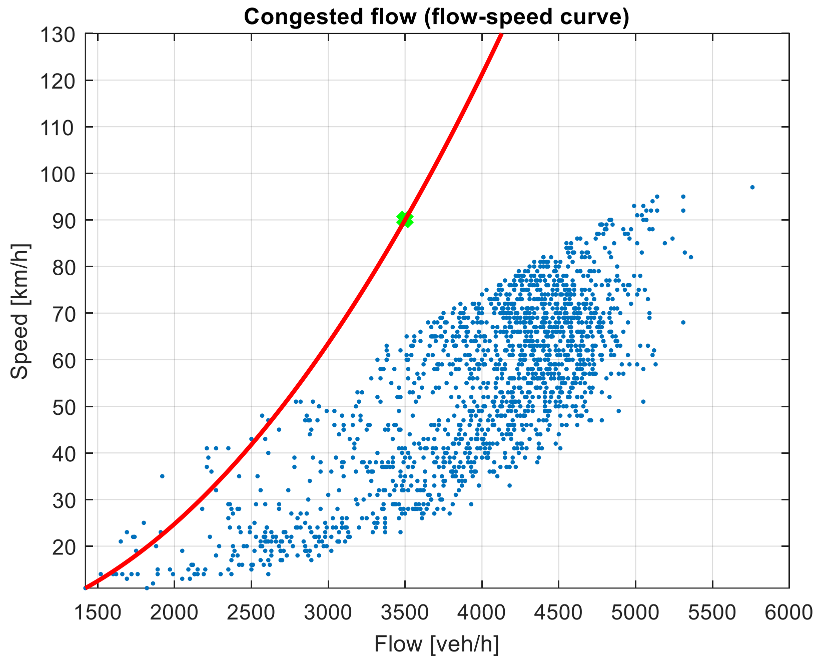


Figure 64 - Curve for new thresholds.

Using Python, an initial test to control the speed limit resulted in the values presented in Fig. 65. Using the collected data, the speed presents frequent change, not being ideal. For this reason, hysteresis was applied, with an oscillation value of 250 veh/h, which resulted in the right graph of Fig 65. Low oscillation values were tested and did not lead to the desired results, i.e., frequent speed limit changes were still notable in some points. By comparing the obtained speed limits to the Figs. 61 and 62, it is possible to see a dependence between speed limit, flow, and average speed of vehicles. The total flow [veh/h] and the average speed of vehicles [km/h] are also presented in Fig. 66 to better illustrate the dependence with the speed limit changes. The higher the flow, the higher the speed limit, but lower is the average speed. This result indicates that the flow is not congested and that the thresholds must be reevaluated. It is important to notice that this is a primary analysis of the implemented controller code, and the controller acts on the measured parameters, not correcting the flow according to the speed limit changes.

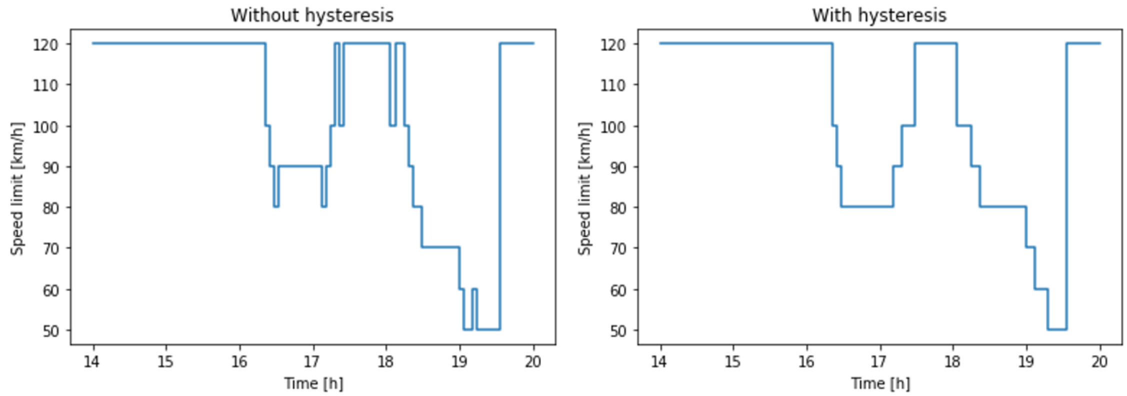


Figure 65 - Speed limits without and with hysteresis.

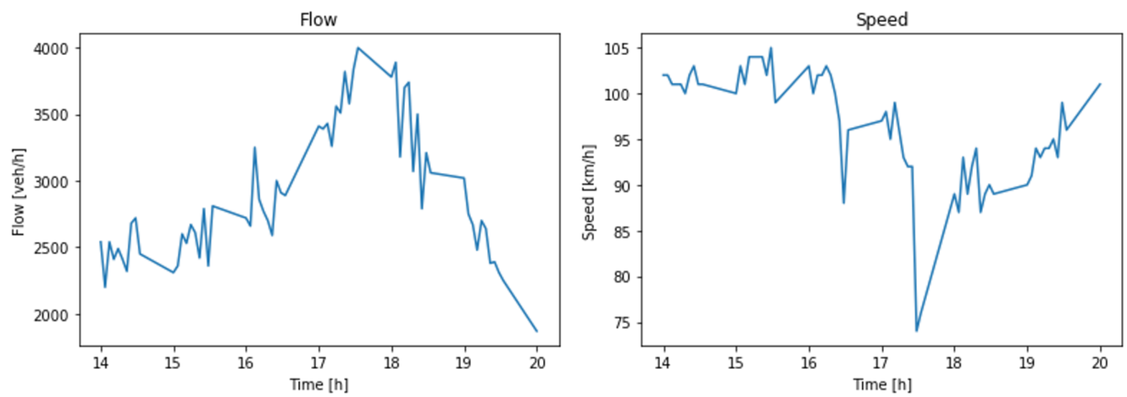


Figure 66 - Flow and average speed on 11/03/2014 from 14:00 to 20:00hs.

7.4. Ruled-based VSLC

The first method used to implement the VSLC to the simulation was the rule-based one. A rule-based system consists of a system that involves curated rules, i.e., does not use an automatic rule inference. To do that, the VISSIM COM interface was used, as well as the MATLAB code presented in section 5.2. With the calibrated model, the Python script, used to test a basic VSLC using off-line sensor data, was utilized as a base to implement the VSLC in MATLAB. This approach is based on the step function, in which pre-determined values activate the VSLC and controls the speed limit without sudden changes.

In VISSIM, it is possible to change the desired speed distribution during the simulation [5, 16, 20, 22]. This can be made through Desire Speed Decisions points (Fig. 67), which defines from each point the speed distribution will change, i.e., where the VSLC will be positioned. Each lane must contain one Desire Speed Decisions point to change the speed distribution by changeable speed limits in all lanes. Other than that, each point is divided into vehicle classes so that the speed can be different according to the vehicle type.

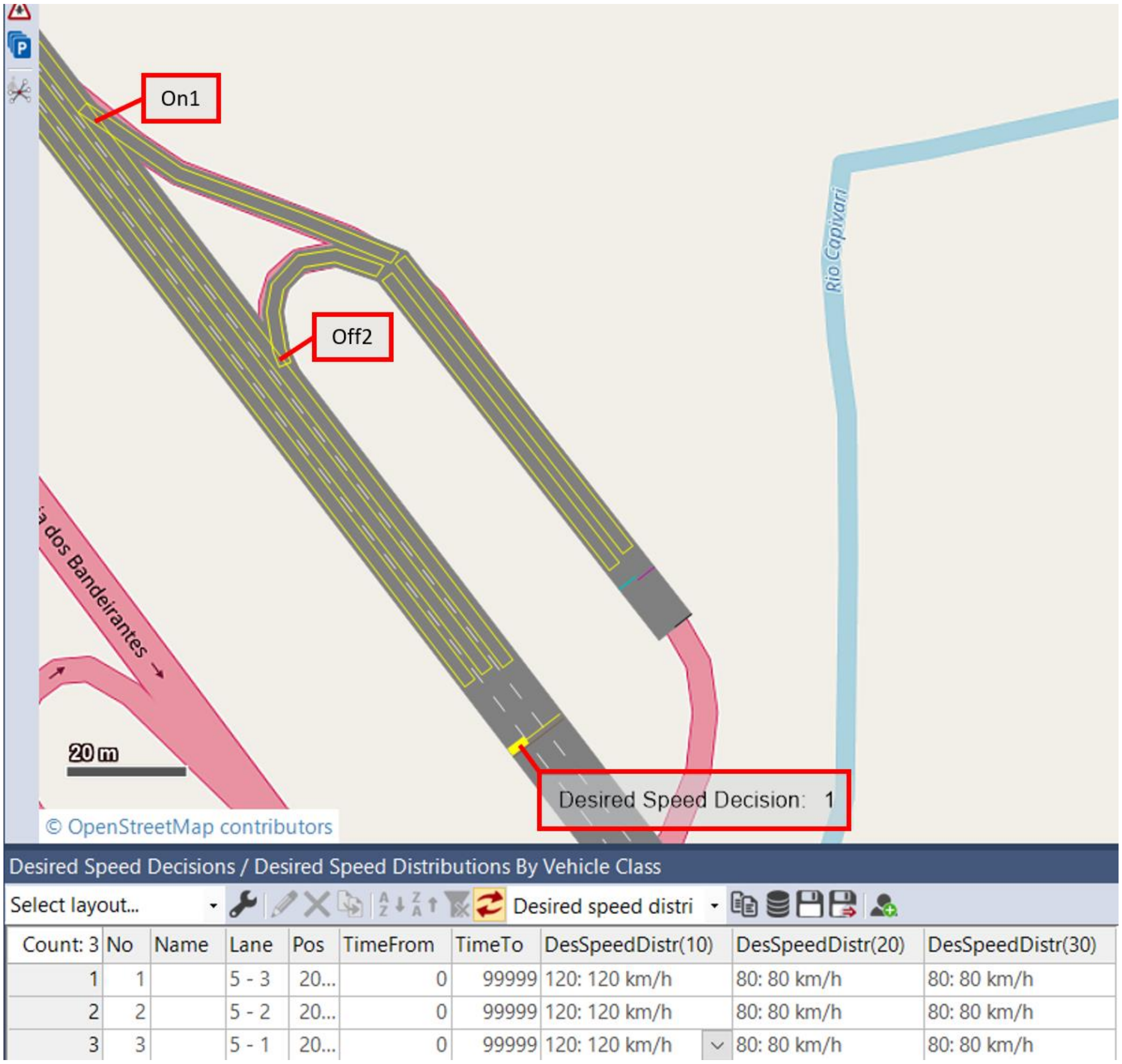


Figure 67 - Desired Speed Decision points.

The access to these points via MATLAB can be either to collect values or to change them, using the following lines:

```
desspeed = vnet.DesSpeedDec;

desspeed1 = desspeed.ItemByKey(1);

desspeed1.get('AttValue','DesSpeedDistr(10)'));
desspeed1.get('AttValue','DesSpeedDistr(20)'));
```

```
desspeed1.get('AttValue','DesSpeedDistr(30)'));

desspeed1.set('AttValue','DesSpeedDistr(10)',120);
desspeed1.set('AttValue','DesSpeedDistr(20)',80);
desspeed1.set('AttValue','DesSpeedDistr(30)',80);
```

where ItemByKey(1) is referent to lane 1 and the values 10, 20, and 30 are for cars, HGV, and bus, respectively. Using these commands and the Python script, a rule-based VSLC was implemented in MATLAB.

The results obtained were not satisfactory, which led to new threshold values presented in Table 1. In Figs. 3 to 5, it is possible to see the results with the new values and using 90 km/h as the critical speed. Finally, in Table 2, one can verify the average speed values for the simulation with and without the VSLC.

Table 20 - New threshold values.

Speed [km/h]	Flow [veh/h]
50	3,000
60	3,200
70	3,400
80	3,600
90	3,800
100	4,000

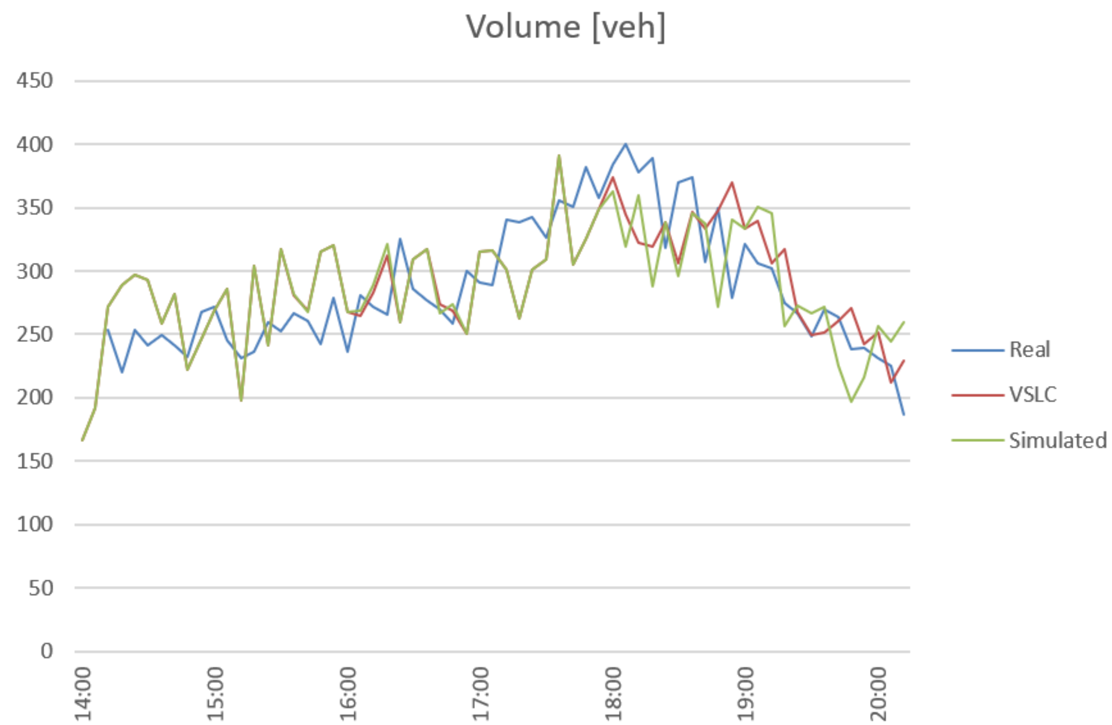


Figure 68 - Total volume comparison.

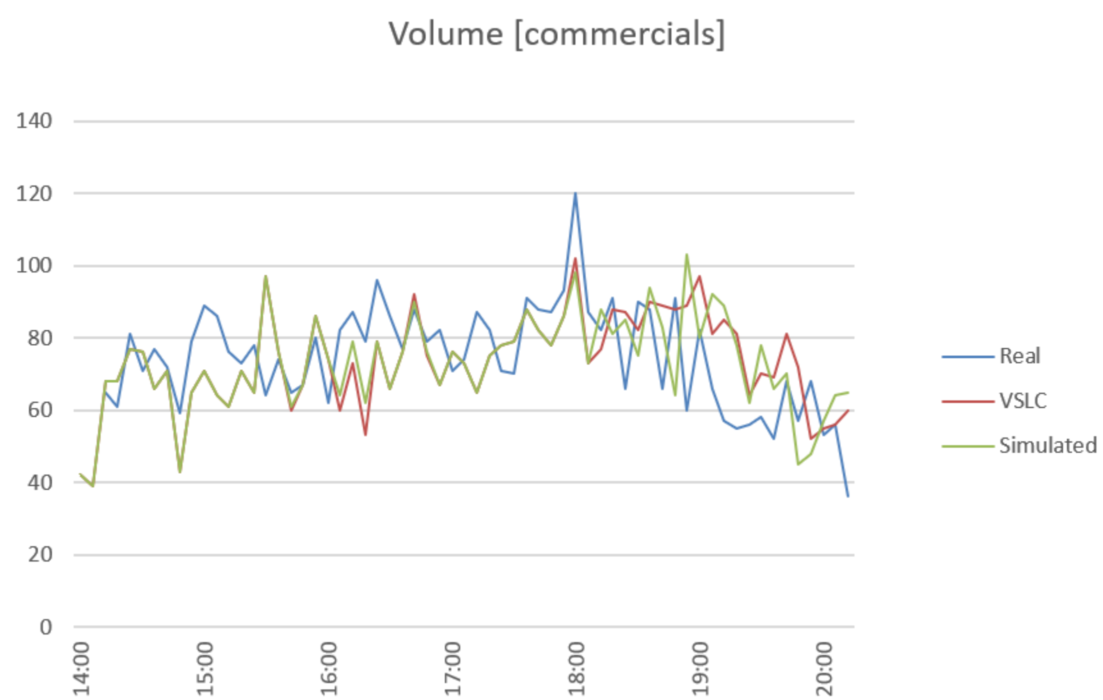


Figure 69 - Volume of commercials comparison.

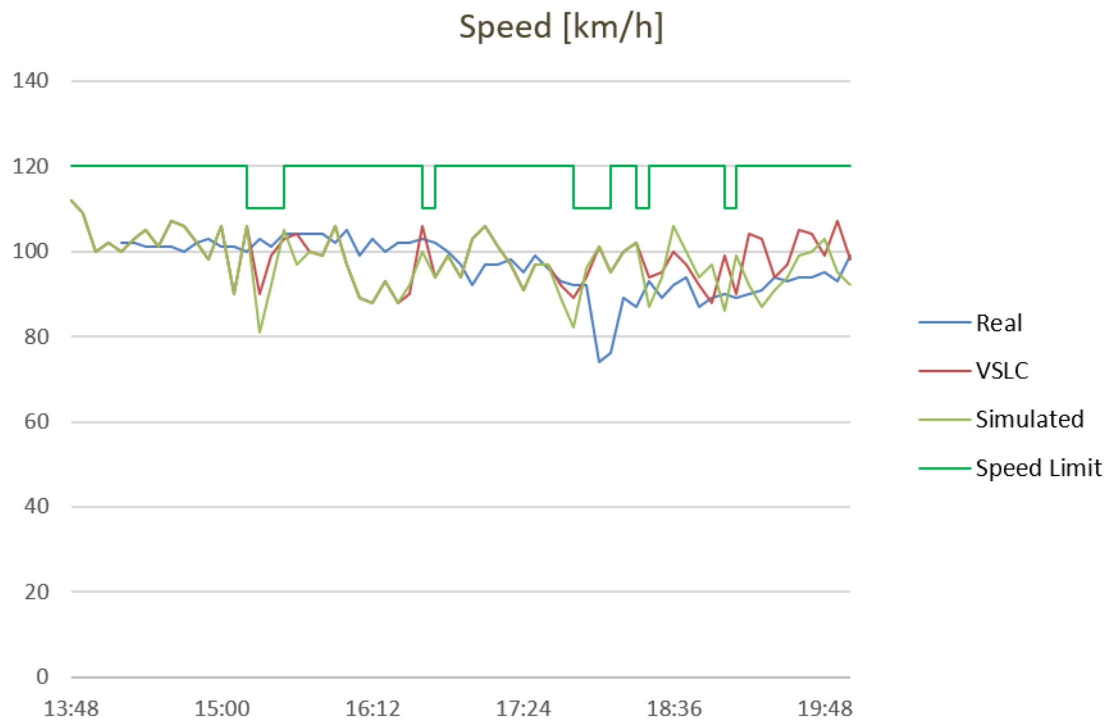


Figure 70 - Speed comparison.

Table 21 - Average speed comparison.

Hour	**14:00**	**15:00**	**16:00**	**17:00**	**18:00**	**19:00**	**Total**
Simulated	102.4	98.2	93.4	95.9	97.6	94.6	96.9
VSLC	102.4	100.3	93.8	96.7	96.4	100.2	98.3

It is possible to observe that the traffic condition was improved in some points and receded in others. The results did not improve 100% of all the possible traffic situations, but this is expected considering the highway characteristics, the simulation response and, majorly, the high share of commercial vehicles, which have a speed limit of 80 km/h according to Brazilian law. In turn, the average speed, considering the whole period, was increased by 1.4%. Due to the previously mentioned aspects, this improvement is acceptable.

7.5. Fuzzy Logic VSLC

Fuzzy Logic was the second method used to implement the VSLC to the simulation. This logic is a computational paradigm that is based on human reasoning, i.e., evaluating the world in imprecise terms and responding with precise actions. In this context, human brain reasoning (uncertainties) and machine reasoning (precise valuations) are combined to solve complicated problems like VSLC.

The basic features of a Fuzzy Logic reasoning model are fuzzy sets, membership functions, fuzzy rules, and outputs. Fuzzy sets are sets whose elements have degrees of membership, i.e., it is a mathematical model of vague qualitative or quantitative data, usually generated through the natural language. A membership function of a fuzzy set is a generalization of the indicator function for classical sets, representing the degree of truth as an extension of valuation. This function can be of variable shapes, being the triangular, trapezoidal, and the Gaussian membership functions the most common (Fig. 3). In turn, the fuzzy rules determine the degree of correlation of a premise and a consequence, differently than crisp logic, where a premise can only be true or false. Finally, the outputs, i.e., the consequences, are the response values of a Fuzzy Logic model, also being arranged in membership functions.

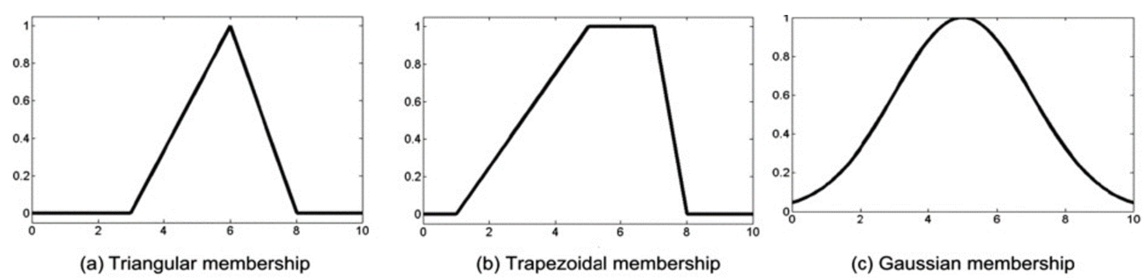


Figure 71 - Membership function shapes.

The Fuzzy Logic process consists of three main stages: the fuzzification of the input variables, the construction of inference, and the defuzzification of the output variable membership function [14]. The fuzzification is the first stage and is responsible for converting crisp values into a set of membership variables, which are defined by membership functions. The second stage is composed by determining the fuzzy rules, or knowledge base, which is created based on operator experience, expert opinions, and system knowledge. As previously mentioned, these rules are responsible for setting the degree of correlation between premises and consequences. Finally, at the defuzzification stage, the consequences, or outputs, are converted into crisp values so that it is possible to analyze them and/or follow with other computational mathematical processes. To implement the VSLC using Fuzzy Logic, the MATLAB tool "Fuzzy Logic

Designer" can be applied (Fig. 4), and this something to be considered in continuation of the described research.

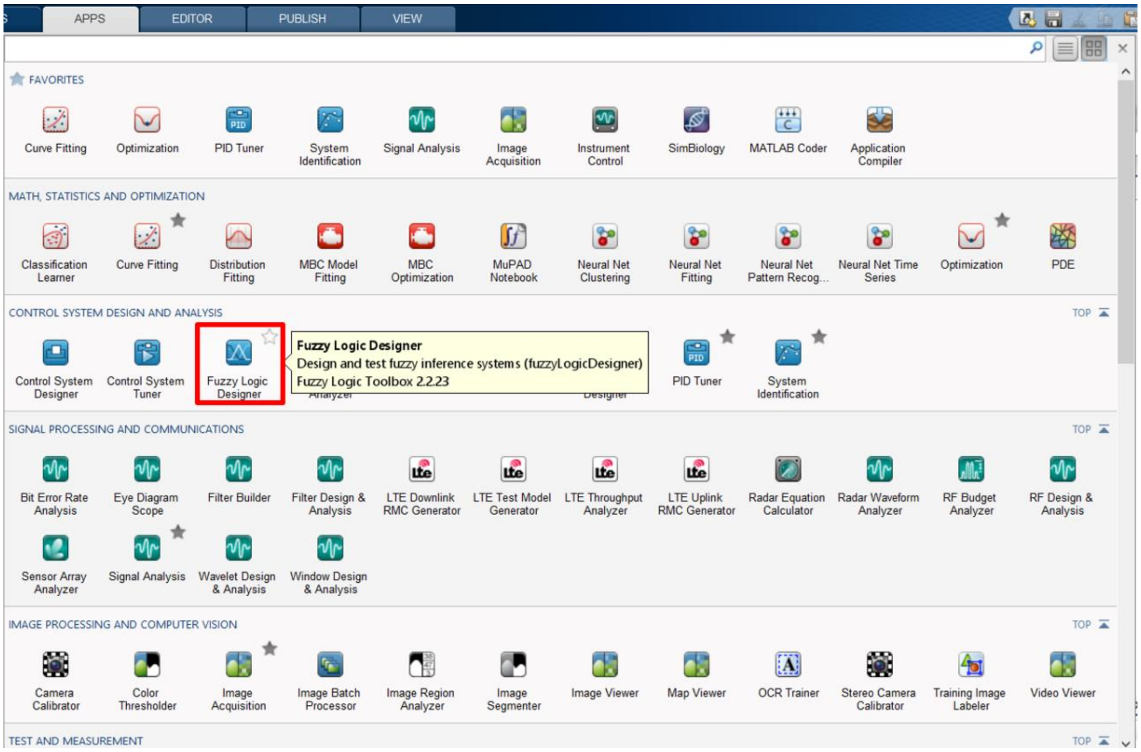


Figure 72 - Fuzzy Logic Designer tool in MATLAB.

8. Conclusion

This technical report results from Helena Vizioli's traineeship in the Department of Intelligent Transportation Systems (ITS), Faculty of Transport and Traffic Sciences (University of Zagreb). The internship was financed by the ERASMUS+ student mobility program to improve the collaboration between the University of Zagreb, Croatia, and the University of São Paulo, Brazil. To support this internship, the company PTV granted a student license of the microscopic simulator VISSIM.

The available highways traffic data were analyzed, and the basic steps to work with VISSIM and MATLAB were learned. Other than that, how to connect them via COM interface was also studied. Regarding the analysis, the parameters taken into consideration were data availability, highway segment length, number of on- and off-ramps of the segment, and data's relevance for the research's aim. As a result, the segment between sensors 20 and 21 was selected, more precisely this segment's data on the 11th March 2014 between 13:00 and 20:00 hours. To understand the basics of the programming languages and VISSIM, both platforms' elemental operators, commands, and features were studied and presented in this report.

Using VISSIM, the chosen segment was created, and the flow data of sensor 20 were input. The traffic flow on the on and off-ramps was randomly distributed for the first analysis. The VISSIM COM interface was deeply studied to obtain the output using MATLAB. The first simulation results were used to calibrate the model, i.e., to change the parameters according to the need since the VISSIM's default parameters are not always compatible with real-world data.

As previously mentioned, the connection of VISSIM and MATLAB allowed the evaluation of the simulation. To obtain a model representing reality with a certain level of accuracy, parameters such as input flow and vehicle distribution though all possible routes were changed. Each simulation in the calibration procedure was analyzed based on the GEH statistics, reaching a satisfactory result.

To implement the VSLC, it was first necessary to analyze the thresholds to create a first simple rule-based VSLC and on the first steps to create a simulation in VISSIM, studying which parameters had to be changed. Regarding the thresholds analysis, needed parameters were obtained by plotting the data from 3 yeas (2012-2014). The values, however, did not fit the data from the selected date (11/02/2014). Thus, new values were defined, and a first VSLC was implemented using MATLAB, obtaining graphs of the speed limit through time.

A small improvement in the average speed was noticed after implementing the VSLC into the VISSIM simulation with the adjusted parameters and threshold values. Considering the whole period, from 14:00 to 20:00 on 11.03.2014, there was an increase of 1.4% on the average speed. Some problematic periods were positively affected by the VSLC, while others did not present an ideal result. In general, the controller worked, and the points in which the traffic condition was not improved can be justified by the highway characteristics, the simulation response, and, majorly, the high share of commercial vehicles, which have a speed limit of 80 km/h. In this way, for possible future work, it is interesting to reduce speed variability. Other than that, the calibration can be improved, as well as the VSLC itself. Finally, in the future, the VSLC can be implemented using a Fuzzy-Logic-based approach.

Acknowledgment

The ERASMUS student exchange program financed the internship of the student Helena Tanoue Vizioli according to the agreements between the University of Zagreb, Croatia, and the University of São Paulo, Brazil. The authors thank the company PTV for granting a research license of the microscopic traffic simulator VISSIM and the assistant Krešimir Kušić for helping with the MATLAB implementation of the VSLC code.

Bibliography

1. BETHONICO, Felipe Costa, **Calibração de simuladores microscópicos de tráfego através de medidas macroscópicas**, Tese de Doutorado, Universidade de São Paulo, 2016.
2. CARVALHO, Luan Guilherme Staichak, **Calibração do modelo de desempenho de caminhões do Vissim**, Tese de Doutorado, Universidade de São Paulo, 2018.
3. CARVALHO, Luan Guilherme Staichak, SETTI, José Reynaldo, **Calibração do modelo de desempenho de caminhões do Vissim usando dados de GPS embarcado**. In: ANPET. Anais do XXXII Congresso de Pesquisa e Ensino em Transportes da ANPET. Rio de Janeiro, RJ. 2018. pp. 3601-3612.
4. DA CUNHA, André Luiz Barbosa Nunes, DE MORAES MODOTTI, Maíra, SETTI, José Reynaldo. **Classificação de caminhões através de agrupamento por análise de cluster,** In: ANPET. Anais do XXII Congresso de Pesquisa e Ensino em Transportes da ANPET. Fortaleza/CE. pp. 1447-1459. 2008.
5. DERVISOGLU, Gunes et al., **Automatic calibration of the fundamental diagram and empirical observations on capacity**, In: Transportation Research Board 88th Annual Meeting, 2009.
6. HOLM, Peter et al., **Traffic analysis toolbox volume IV: Guidelines for applying corsim microsimulation modeling software**, United States, Department of Transportation, Intelligent Transportation Systems Joint Program Office, 2007.
7. HOUCQUE, David, **Introduction to Matlab for Engineering Students**, Northwestern University, Aug 2015.
8. Iubunam, **Vissim COM Example Code File**. IUbunam's Archive, 2020. In: <http://www.iubunam.com/vissim/vissim-com-example-code-file/>. Accessed 05/06/2020.
9. LACERDA, Victor Macêdo, CASTRO-NETO, M. M., **Considerações sobre a calibração do modelo de car-following do vissim para vias arteriais urbanas,** In: Anais do XXVIII Congresso da Associação Nacional de Pesquisa e Ensino em Transportes, ANPET, Curitiba, PR. 2014.
10. LIU, Xiaoyue Cathy; HAGHIGHI, Nima, **Travel-Time Reliability in Simulation and Planning Models: Utah Case Study (SHRP2 L04 IAP Round 7),** Final report: UT-19.21 2019.
11. LLANQUE, Ayala; ROSEMARY, Janneth. **Procedimento para identificação dos principais parâmetros dos microssimuladores a serem considerados no processo de calibração**, Dissertação (Mestrado em Transportes)—Universidade de Brasília, Brasília, 2013.
12. MORA, Pedro; VIANNA, Felipe, **Guia PTV Vissim 9**. Universidade Presbiteriana Mackenzie, In: < https://www.mackenzie.br/fileadmin/OLD/62/ARQUIVOS/PUBLIC/user_upload/_imported/fileadmin/LABGEO/Trabalhos/2017/PTV_VISSIM_-_Manual_do_usuario__versao_1_.pdf>, Accessed 03/16/2020
13. NORONHA, Henrique Dela Bruna et al., **Estudo comparativo de dois microssimuladores de tráfego-Aimsun e Vissim**, CC (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Engenharia Civil, 2016.

14. RESENDE, Paulo Tarso Vilela de. **Custos logísticos no Brasil 2017**. In: <https://www.fdc.org.br/conhecimento/publicacoes/relatorio-de-pesquisa-33324>. Accessed 03/16/2020.
15. SAVOLAINEN, Roope et al. **Modeling and analyzing Helsinki's traffic network using a microscopic simulator**, School of electrical engineering, Aalto University, Master thesis, 2018.
16. SKRUŽNÝ, Filip. **Control System Development of Traffic Signal Control in MATLAB for PTV VISSIM**. Change, v. 1, n. 1, pp. 2, 2015.
17. TETTAMANTI, Tamás; VARGA, István. **Development of road traffic control by using integrated VISSIM-MATLAB simulation environment**. Periodica Polytechnica Civil Engineering, v. 56, n. 1, pp. 43-49, 2012.
18. TETTAMANTI, Tamas. **The beginning steps of VISSIM COM programming with MATLAB script,** Budapest University of Technology, Dept. of Vehicle and Transportation Control, In: http://www.kjit.bme.hu/images/trafficlab/Research/VISSIM/MATLAB_Vissim-COM.pdf, Accessed 03/16/2020
19. VISION, P. T. V. **VISSIM 5.30-05 User Manual**. Planung Transport Verkehr AG, 2011.
20. VISION, P. T. V. **VISSIM 5.40-user manual**. Product manual, Planung Transport Verkehr AG, 2009.
21. WALLAME Jorge, **MatlabPTV_Vissim-Interface**. GitHub, 2014. In: <https://github.com/wallamejorge/MatlabPTV_Vissim-Interface>. Accessed 05/06/2020.
22. WON, Jongsun; LARUE, Betsy. **PTV Vissim - Speed Definition**. PTV Group Traffic, 2020. In: <https://ptvtraffic.us/resources/speed-definition/>. Accessed 06/10/2020.
23. André Luiz Cunha, José Reynaldo Setti, T**ruck Equivalence Factors for Divided, Multilane Highways in Brazil**, Procedia - Social and Behavioral Sciences, Volume 16, 2011, pp. 248-258, ISSN: 1877-0428, DOI: 10.1016/j.sbspro.2011.04.447.
24. CUNHA, André Luiz Barbosa Nunes da. **Avaliação do impacto da medida de desempenho no equivalente veicular de caminhões**. 2007. Dissertação (Mestrado em Planejamento e Operação de Sistemas de Transportes) - Escola de Engenharia de São Carlos, University of São Paulo, São Carlos, 2007. DOI: 10.11606/D.18.2007.tde-27112007-094400. Accessed: 11/03/2020
25. CUNHA A.L., BESSA J.E., SETTI J.R., **Genetic Algorithm for the Calibration of Vehicle Performance Models of Microscopic Traffic Simulators**. In: Lopes L.S., Lau N., Mariano P., Rocha L.M. (eds) Progress in Artificial Intelligence. EPIA 2009. Lecture Notes in Computer Science, vol 5816, 2009, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04686-5_1
26. CARVALHO, Luan Guilherme Staichak, SETTI J.R., **Construção de Perfis de Velocidade de Caminhões Utilizandofiltro Gaussiano e Regressões Lineares em Dados ee GPS**, In: Anais do XXXI Congresso da Associação Nacional de Pesquisa e Ensino em Transportes, ANPET, Recife, 2017.
27. CONTRAN, In: <https://www.gov.br/infraestrutura/pt-br/assuntos/transito/conteudo-denatran/resolucoes-contran>, Accessed: 06/03/2020