Czech Technical University in Prague
Faculty of Transportation Sciences
Department of Transport Telematics

Modeling of Convoy Driving Behaviour

Masters thesis

Bc. Václav Kroupar

Masters programme: Intelligent Transport Systems
Supervisor 1: Doc. Ing. Pavel Hrubeš, Ph.D.
Suprevisor 2: FH-Prof. Emil Simeonov, Dipl.-Ing, Mag

Prague, November 2017

CZECH TECHNICAL UNIVERSITY IN PRAGUE
Faculty of Transportation Sciences
Dean's office
Konviktská 20, 110 00 Prague 1, Czech Republic

K620..Department of Transport Telematics

MASTER'S THESIS ASSIGNMENT

(PROJECT, WORK OF ART)

Student's name and surname (including degrees):

Bc. Václav Kroupar

Code of study programme code and study field of the student:

N 3710 – IS – Intelligent Transport Systems

Theme title (in Czech): **Modelování jízdy vozidel v koloně**

Theme title (in English): Modeling of Convoy Driving Behaviour

Guides for elaboration

During the elaboration of the master's thesis follow the outline below:

- Review of existing modeling approaches,
- Design model and evaluation criterias,
- Evaluate experimental traffic flow data,
- Create simulation of the traffic flow,
- Compare the results of simulation and the experiment, discuss the results.

CTU
CZECH TECHNICAL UNIVERSITY IN PRAGUE

Graphical work range:	Not specified
Accompanying report length:	at least 55 pages (including images, graghs, and tables that are included in accompaing report)
Bibliography:	P. Fritzson, Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach, IEEE Press, 2015 L. Elefteriadou, An introduction to traffic flow theory, Springer, 2014

Master's thesis supervisor: **doc. Ing. Pavel Hrubeš, Ph.D.**

Date of master's thesis assignment: **July 21, 2016**
(date of the first assignment of this work, that has be minimum of 10 months before the deadline of the theses submission based on the standard duration of the study)

Date of master's thesis submission: **November 30, 2017**

a) date of first anticipated submission of the thesis based on the standard study duration and the recommended study time schedule
b) in case of postponing the submission of the thesis, next submission date results from the recommended time schedule

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
L.S.
-3-

doc. Ing. Pavel Hrubeš, Ph.D.
head of the Department
of Transport Telematics

prof. Dr. Ing. Miroslav Svítek, dr. h. c.
dean of the faculty

I confirm assumption of master's thesis assignment.

Bc. Václav Kroupar
Student's name and signature

Prague ....July 1, 2017

Thesis Supervisors:

Doc. Ing. Pavel Hrubeš, Ph.D.
Department of Telecommunication Engineering
Faculty of Transportation Sciences
Czech Technical University in Prague
Konviktská 20
Prague 1, 110 00
Czech Republic

FH-Prof. Emil Simeonov, Dipl.-Ing, Mag
UAC Technikum Wien
Höchstädtplatz 6
1200 Vienna
Austria
Vienna

Declaration

I hereby declare I have written this Master thesis independently and quoted all the sources of information used in accordance with methodological instructions on ethical principles for writing an academic thesis. Moreover, I state that this thesis has neither been submitted nor accepted for any other degree.

In Prague, 30th November 2017

....
Bc. Václav Kroupar

Abstract

Abstract

Rapidly increasing capabilities of the modern day computing is well know phenomena of the current time period. When this computational power is combined with the accurate geo-spatial positioning a new possibilities how to simulate the behavior in the traffic stream are opening.

The focus of this master thesis is on the exploration of the ways how to perform a better simulations based on the well known Car Following Model benefiting from the accurate data. After the application of a model the work continues in a search and validation of a suitable parameters describing the behavior of the driver.

Abstrakt

Rapidně rostoucí možnosti výpočetní techniky jsou známým fenoménem poslední doby, a když se k tomuto výpočetnímu výkonu přidají přesné polohové informace z vozidel otevírají se nové možnosti, jak simulovat chování vozidel v dopravním proudu.

Tato práce si klade za cíl prozkoumávat nové způsoby, jak lépe simulovat dopravní chování na základě ověřených modelů sledu vozidel s využitím reálných dat. Po aplikaci modelu práce pokračuje v hledání a validaci vhodných parametrů popisujících řidičovo chování.

Keywords: car platoon, modeling, car to car communication, C2C, Modelica, car following models, traffic dynamics
Klíčová slova: kolona vozidel, modelování, komunikace vozidlo vozidlo, C2C, Modelica, model sledu vozidel, dopravní dynamika

Acknowledgements

First of all, I would like to express my gratitude to my thesis supervisors, Doc. Ing. Pavel Hruebeš PhD and FH-Prof. Emil Simeonov Dipl.-Ing. Mag. They have been a constant source of encouragement and helped me with numerous problems.

I cannot forget to thank my family members and friends, for their infinite patience, care and tenderness.

Contents

List of Tables

List of Figures

Listings

List of Acronyms

C2C Car to Car. viii, x, 2, 4, 5, 7–11, 22

CFM Car Following Model. 12–14, 46

CSV Comma-separated values. x, 8, 9, 21, 22

CTU Czech Technical University. 2, 4, 46

FD Faculty of Transportation Sciences. 2, 4, 46

GCD Great Circle Distance. 23

GHR Gazis Herman Rothery. 13

GPS Global Positioning System. x, 5, 7–9, 23–25

HMI human machine interface. 5, 8

PCC Pearson correlation coefficient. 14, 34

SP Stationary points. 23, 24, 28, 30, 31, 33–35, 42

SQL Structured Query Language. 22, 24

TACR Technology Agency of the Czech Republic. 4

V2I Vehicle to Infrastructure. 2

Chapter 1

Introduction

"Transportation is the centre of the world! It is the glue of our daily lives. When it goes well, we don't see it. When it goes wrong, it negatively colours our day, makes us feel angry and impotent, curtails our possibilities."

— Robin Chase

1.1 Introduction

Beginning with a quote of Robin Chase - future transportation enthusiast, co-founder of car-sharing company Zip-car as well as strong supporter of collaborative economy - I would like to introduce this master thesis. [2] Future of transportation will always have both sides positive and negative and I hope this thesis will contribute to the first one.

To write this master thesis, I was inspired by three factors. The first one is the modeling itself, the second one is my own life experience, and the third one is the work of a former student of our faculty.

First element - modeling of systems, allows us to extend borders of human knowledge by recreating almost every existing object in the universe, putting it to known or unknown environment, and bringing them to its limits, or even beyond them. And this all is possible with decreased risks and dangers for real actors. Therefore, we can find many projections of small or large pieces of our world - starting with atoms and molecules all the way up to the simulations of the universe - modeled and closely examined by many scientists of all kind. Traffic modeling is especially helpful when it comes to the transportation of people It is highly irrational to waste time and energy by constructing inappropriate communications. With macro and micro simulations we can eliminate this and detect possible bottlenecks problem even before the traffic reaches the critical point. In close future vehicles are going to be equipped with C2C and Vehicle to Infrastructure (V2I) modules allowing a continuous exchange of the information during the driving and using them to influence their direction. Therefore, it is important to predict the movement of surrounding vehicles. Moreover, this is all possible because of the vast increase of computational power modern computers provides and development of new modeling techniques. With lower effort more people can create their own simulations and suggest improvements to their environment.

The second factor is the possibility of gathering traffic flow data. Most of the drivers in their lives experience struggles with heavy traffic and another trouble connected with delays. Czech Technical University (CTU) Faculty of Transportation Sciences (FD) had an opportunity to be part of the carefully prepared C2C experiment and provided me with the output. The set of data is interesting itself, because of the rareness of this kind of information from the whole connected convoy and because of the imperfections and errors in the dataset.

Finally, the third aspect is the work of former student of CTU - FD - Klára Menglerová. She created a mathematical model which, based on simple physics, should be able to simulate a behavior of a real person driving in convoy. However, yet no-one tried to create a simulation using this model. I believe that continuing a work of our present or former colleagues is our privilege and I would like to extend her work and use the opportunity to compare it with the real-world data.

1.2 Goals of the Master Thesis

Based on the inspirations mentioned above there are several goals of this master thesis:

1. Start reviewing current existing modeling approaches and techniques that could be used further on during the practical part of the work;

2. Design the model itself and evaluation criteria to recognize the quality of the outcomes from the model;

3. Examine characteristics of data available and make adjustments and smoothings

when are needed;

4. Create a simulations of the traffic flow with different parameters; and

5. Compare the results of simulation and the experiment and discuss the results.

These are the main and general goals of this theses and I will address them in deeper details further in the text. Anyway, there is one important difference in the modeling to be mentioned in prior. It is possible to simulate course of the ride based on the parameters of the driver as well as to extract characteristics of the driver and the vehicle based on the driving records. This master thesis is addressing both problems.

Chapter 2

Connected vehicle experiment

This chapter is describing a basics of the scientific experiment as well as the experiment Coflow done in cooperation of the FD CTU, a private company Dynavix [1] and Technology Agency of the Czech Republic (TACR)

2.1 C2C Experiment

Experiments are the ways how to support or to refute a hypothesis which are the reasons to start an experiment. In fact an experiments are in the human nature since the earlies days of our lives. Every child starts experimenting with the world around proving a hypothesis how to walk or how to speak. Experiments are staying with us until the day of our days. Goal of scientific experiment is not fundamentally different from the other experiments, but ti emphasizes on careful preparation, controlled environment as well as evaluation methods.

This part is describing the origin of a data I am using later on as an input of the simulations. The data come from an research project Coflow organised by Dynavix- this company is developing navigation systems, Fleet and Work-flow management solutions, Truck solutions and CarPC solutions, with support of TACR and evaluation support from the side of the FD CTU

The main objective of the project was to analyze stop and go waves in the traffic and the influence of additional information produced by the C2C system. The expectations were to reduce the negative effects of stop and go waves on the traffic flow.

[1] www.dynavix.com

The works on the project started in the year 2014 by analyzing the traffic flow problems and proposing a possible ways how to describe them. In 2015 the researchers have started with the simulations of the stop and go waves and started developing their own C2C communication devices and suitable in-vehicle equipment for displaying of the information. During the first half of 2016 researchers have finished the development phase and had started the prove-of-the-concept phase. The experiment in the second half of 2016 was the final step of the project.

The Researchers expected to involve 30-50 vehicles for the final testing at the private Tchořovice airport. Two nearby villages Tchořovice and Kadov owns this airport and they do not longer use it for the original purpose of aviation, instead it is usually used by vehicle markets, local driving school and similar gatherings. It is a great place for driving behavior experiment, because of lack of any regular traffic together with 2km long runaway with the concrete surface [3][4].

The experiment have started on Friday 9th of December 2016 at 9:00 in the morning and continued to the afternoon. There were 33 vehicles and drivers involved most of the time and every vehicle was equipped with two devices:

1. Android smart-phone, which has been used mainly as human machine interface (HMI) and for gathering status android data; and

2. C2C communication device with GPS and WiFi operating on 5 GHz with the dedicated antenna on the roof of the vehicle.

These devices were interconnect to provide an extended driver information about traffic flow ahead the driver. The C2C device was able to receive it's own and position communicate this information with other the devices. HMI provided following information to the driver:

- suggestion to moderated or intensive acceleration;

- suggestion to moderated or intensive deceleration;

- vehicle position on map; and

- distance to closest vehicle.

Drivers were tested with and without this information if there are going to be any change in their driving behavior.

Experiment course is described in the figure 2.1. The biggest rectangular area is representing the runaway and the small circles on the left and right of the figure are representing two traffic cones which were defining the area for the vehicles, each one on the opposite end of the runaway. Small rectangles are the vehicles and the arrowed lines are showing desired route.

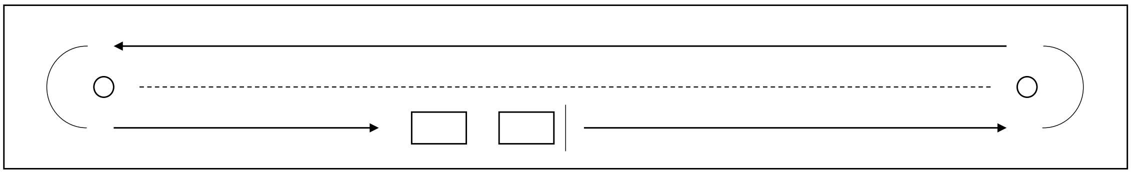


Figure 2.1: Runaway scheme

The experiment was designed the way so the vehicles line up in the half of the runaway and then start cycling around in counter-clockwise direction. After each round the first vehicle stops the starting point and waits for the whole convoy to stop. This basic cycle repeated for different scenarios:

1. 10:04:40 - 10:31:20: driving without any additional information;
2. 10:41:40 - 11:06:20: driving with additional information from second vehicle ahead; and
3. 11:13:30 - 11:39:00: driving with additional information from fourth vehicle ahead.

Because I was not directly involved in the on-field test and I have not found any records from the on field test which would specify the further information about vehicles I had to collect these information by my self from the photos. The authors of these photos were Ing. Martin Langr and Doc. Ing. Pavel Hrebeš Ph.D. some of them capturing the whole scene of the experiment site were very helpful for me. Additional information comes from the website Ultimatespecs [2] which is providing extensive technical details about most of the vehicles manufactured in the last 100 years. The table 2.1 is an output of this small research, showing first 10 vehicles of the convoy. Because it was impossible to obtain further detail than vehicle manufacture, model, generation and approximate year of manufacture the column weight contains average value for the model series. Weight can vary according to the different engines, transmissions, additional equipment and because of the load of the vehicle. Therefore I rounded the value to the 100's kg. This list is not complete, but I consider it satisfactory for needs of following simulations as described in the chapter 4.4

vehicle	detail	length	mass
-	-	*mm*	*kg*
V1	Mercedes C350 - 2012	4526	1600
V2	Ford Focus MkII combi - 2006	4360	1300
V3	Peugeot 206 combi	4028	1100
V4	Skoda Fabia 1	3970	1000
V5	Skoda Fabia 1 combi	4222	1150
V6	Mercedes c320 Sport-Coupe	4343	1400
V7	Skoda Octavia 2 combi	4572	1300
V8	Fiat Panda 2010	3653	900
V9	Ford Focus MkI 3-doors	4337	1200
V10	Opel Astra G caravan 2002	4288	1300

Table 2.1: Details of the vehicles

2.2 Experiment Output

This section is describing parameters of the outputs and their quality. As I have mentioned above there were two sources of data available: C2C device and Android smart phone. The example of .csv file can be seen in 2.2.

Data originating form C2C devices were are containing :

Time stamp indicating the time of the measurement in milliseconds, the device restarts the value of the times tamp every day at midnight so the maximum value is 86 400. Overview of the time stamp range is shown on Figure 2.2.2 b).

Car id is a number which is the same as the position of the vehicle in the convoy. This value is constant, because vehicles did not change order during the course of experiment.

Speed is recording current speed of the vehicle in ms^{-1}.

Latitude and Longitude from the GPS multiplied by 10000000 with the smallest step of 10 cm (Even though the newest GPS specification allows higher accuracy [5])

Acceleration internal is recording the acceleration from accelerometer. This value is depending on the placement of the device and how firmly it is fixed to the vehicle, because it can be influenced by wandering around the drivers cabin.

Acceleration slope is a foundation of the information provided to the surrounding drivers. This information is displayed to the other drivers in their HMIs. This column are the acceleration data parsed into the five categories according to the following scheme:

- below -40 - rapid deceleration;
- from -40 to - 15: light deceleration;
- from -15 to - 15: constant speed;
- from -15 to - 15: light acceleration;
- above 40: rapid acceleration.

0	1	2	3	4	5	6
31 484,00	5,00	$2{,}78 \cdot 10^{-2}$	$4{,}94 \cdot 10^{8}$	$1{,}38 \cdot 10^{8}$	3,00	$2{,}58 \cdot 10^{-2}$
31 484,20	5,00	$5{,}56 \cdot 10^{-3}$	$4{,}94 \cdot 10^{8}$	$1{,}38 \cdot 10^{8}$	3,00	$-1{,}16 \cdot 10^{-2}$
31 484,40	5,00	$9{,}26 \cdot 10^{-3}$	$4{,}94 \cdot 10^{8}$	$1{,}38 \cdot 10^{8}$	3,00	$-4{,}68 \cdot 10^{-2}$
31 484,80	5,00	$1{,}85 \cdot 10^{-3}$	$4{,}94 \cdot 10^{8}$	$1{,}38 \cdot 10^{8}$	3,00	$-8{,}19 \cdot 10^{-2}$

Table 2.2: C2C .CSV example

Second data set is originating from the android-based smart phones. The example from the same vehicle can be seen on the table2.2 Data originating from logger were following:

Time stamp in milliseconds is the same parameter as in previous device, the main difference is mobile devices are counting it from 1st January 1970 (Unix Epoch Time) [6].

Speed the same as the previous C2C dataset, ms^{-1}.

Latitude and longitude the same as the previous C2C dataset. GPS geo-coded in WGS 84.

0	1	2	3
$1{,}48 \cdot 10^{12}$	$-1{,}00$	50,41	14,05
$1{,}48 \cdot 10^{12}$	0,00	49,43	13,80
$1{,}48 \cdot 10^{12}$	0,00	49,43	13,80
$1{,}48 \cdot 10^{12}$	0,00	49,43	13,80

Table 2.3: GPS logger .CSV example

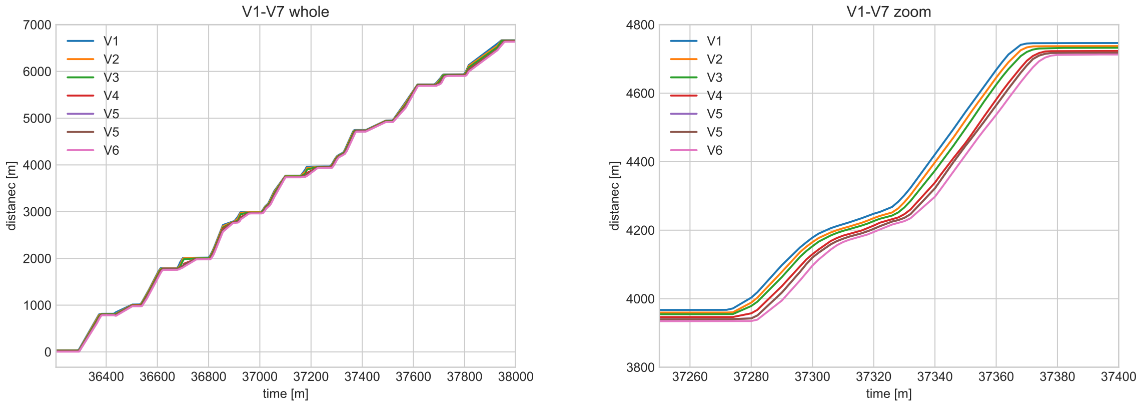


Figure 2.2: Time - Distance example

2.2.1 Missing data

Unfortunately not all the devices provided the data in highest possible quality and some of the data are missing completely. Table 2.2.1 is describing in detail that form 33 vehicles only 18 contained both data sets, respectively 8 C2C data was missing and 7 from smart-phones. This problem is not surprising for smart-phones, because every device can be different and no participant guaranteed which smart-phone is bringing. On the other hand the missing data from C2C devices are a small disappointment, because they were design specifically for this purpose and not saving course of the experiment is a problem.

Vehicles	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
C2C	x	x	x	x	x	x	x	x	x	x	-	-	x	x	x	x	x
Android	x	-	-	x	x	-	x	x	x	x*	x*	x	-	x	x	x	x

Vehicles	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
C2C	-	x	x	x	x	x	x	x	x	x	x	-	-	-	-	-
Android	x	x	-	x	x	x	x	x	x	x	-	x	x	-	x	x

Table 2.4: Missing Files

2.2.2 Unstable Time-stamp

The Second and from my perspective bigger problem to mention is the incoherency of the time-stamps in the data. With this issue it is problematic to match the corresponding records and create good picture of what was happening on the testing site. The figure 2.2.2 is showing dependency of time-stamp value and number of the record form vehicles 1, 4, 5, 7, 8, 9 ,14 and 15. In the ideal state it would be a simple line / linear dependency, but as you can see almost every set of data is containing some type of smaller or bigger steps. These steps can have two possible explanations:

- Synchronization of the time; or
- Missing data

Synchronization would have be done in a smaller deviations from ideal state, but such a big steps as we can see at V1 (2.2.2 a)) are missing data from smart-phone. You can see that only two vehicles with constant time-stamps (1 and 5), on the other hand the other vehicles have interruptions during the time sampling. Very interesting error is V1 which has different slope then every other sample. This seems like the time appears slower for the vehicle one.

C2C data on the other hand are rather stable and only bigger jump is around time 3550 when all of them were probably disconnected for a short moment. Important starting point is after 10:00:00 AM (3600) when vehicles start to circle around the track.

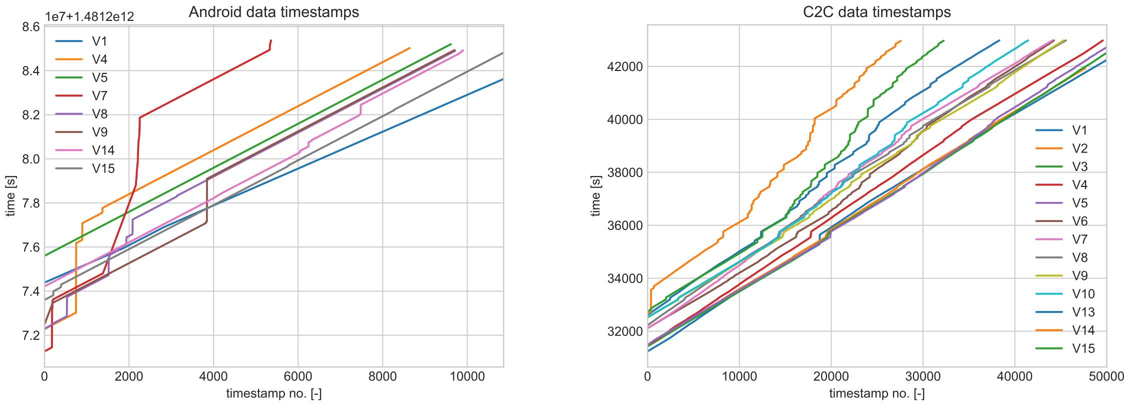


Figure 2.3: Data coherency problem

2.3 Chapter Conclusion

After describing of the experiment I have performed the data analysis of the both data sources (smart phones and C2C device. After this I have decided to work with C2C dataset in the further works and simulations for the reason of better reliability and higher accuracy of this data set.

Chapter 3

Traffic Modeling

In this chapter I would like to describe basics of traffic modeling and focus on the model created by Klára Menglerová in her Phd. thesis. [7] In the next chapters I am describing how I have used these foundations and implemented them.

3.1 Traffic modeling introduction

Literature divides traffic modeling by level of detail into three main groups [8]:

- Macroscopic modeling for large scale areas (state, metropolitan etc.) mainly used for developing of new traffic strategies. This approach is using traffic flow performance characteristics (flow, speed, density) and characteristics of the network (maximal speed, maximal capacity, etc.)

- Microscopic modeling is an approach which is simulating behavior of every vehicle, pedestrian or cyclist in smaller scale (junction, parking space, lane, etc.) When it is compared with macroscopic simulation for the same area it reaches higher accuracy in the cost of the computational time.

- Mesoscopic modeling is combining approaches of macroscopic and microscopic approach trying to benefit from both. It is using the same principle as the macroscopic method, but for a smaller areas.

One part of micro simulations I am going to use in this thesis is Car Following Model (CFM). CFM describes the processes how the drivers follow each other in the traffic stream. When compared CFM to the movement of a single vehicle more restrictions come in place. At first the following vehicle cannot collide with vehicle it follows. Furthermore it has to respect the safety distance between them and it has to remain sufficient speed [8].

One of the first and fundamental works covering the history of CFM is work of Brackstone and McDonald [9] as well as more recent works of another authors (V. Papathanasopoulou for example [10]).

- 1958 - Gazis Herman Rothery (GHR) model [11]. This model was based on a hypothesis that a driver's acceleration was proportional to change of the vehicle velocity, or deviation from a set of following distance, which could be self dependent. After the introduction the model was many times reviewed and improved by another researches.

- 1959 - Safety distance or collision avoidance models [12]. This model seeks a safe following distance using basic Newtonian equations of motion. The drivers in the model acted 'unpredictably' and the coefficients of the model were statistically calibrated after several runs of the testing vehicle.

- 1959 - Linear (Helly) models [13]. This model was based on similar principles as GHR model, but Helly added acceleration according to whether the vehicle in front was breaking or accelerating.

- 1963 - Psychophysical or action point models (AP) [14]. The concept that this model introduces is that the drivers are able to distinguish when they are approaching a vehicle from further distance. The author based his assumption on visual input of the driver - primarily apparent size of the vehicle.

- 1992 - Fuzzy logic-based models [15]. This model is introducing 'fuzzification' of the GHR model and changes the distribution of the simulation attributes.

- 2000's - Re-Exploring CFM in new conditions such as free-flowing, approaching, close-following, car-following, emergency breaking, and stop-and-go. [16]

This means that at first we have to Choice of the output variable is eternally dependent on the creator of the system.

Pearson correlation coefficient (PCC) As evaluation criteria I have chosen PCC, because of the time - distance representation of the results is linear and steadily increasing through the time. And those are the criteria to use PCC The goal is to search the simulation against with better correlation against the real values. Equation of PCC is in the 3.1. [17]

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.1}$$

3.2 Description of the Model

This section is going to describe the simulation model created by Klára Menglerová [7] - described mainly in the chapter 4 of her thesis. The model can be classified as CFM, because it is describing the movement of the second vehicle in the vehicle-pair based on the leading vehicle. In figure 3.2 you can see all the forces impacting the vehicle as well the other variables.

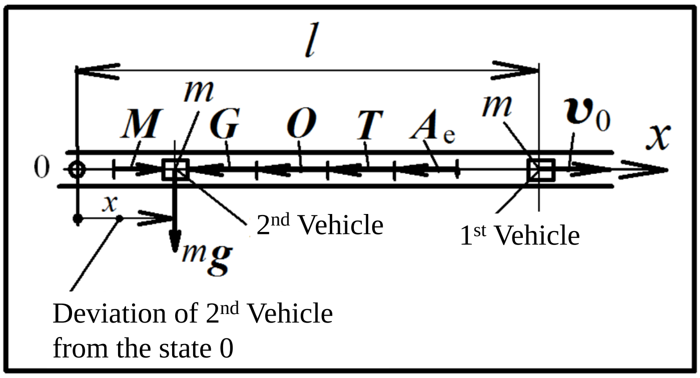


Figure 3.1: Model variables

Safety distance l is the distance between two vehicles recommended for safe driving. Because this force changes with the speed of the vehicle is I have also introduced a safety time t_s which is recognized by many road safety authorities to be 2 seconds [18], [19]. $l = vt_s$

The model is defined as follows: Vehicle drives in a constant velocity in straight, horizontal trajectory v_0. The first vehicle is followed by vehicle no. 2 and we consider only longitudinal interactions between those two vehicles. Simplifying the problem, we can consider both vehicles as two point particle with mass m. We connect the vehicles by an axis with staring point 0 in the equilibrium state of second vehicle (which is safety distance l). As we can see on figure 3.2. Core of this model is an interaction of the following vehicle with the first one, which is described by five forces M, T, G, O, A_e. Forces G and O are describing how is the driver in the following vehicle able to react on the changes in the traffic flow. The other forces are describing the other influencing parameters of the environment.

Binding force G is reflecting how is the driver able to keep constant distance between the vehicles. In reality this force is strongly depending on the skills of the driver and especially on the control delay. This force is considered to be linearly dependent on the distance between two following vehicles l. Therefore, it is linearly dependent on the deviation x from the equilibrium 0. The Greater the deviation is, the greater is the force on the second vehicle.

$$G = -cx, c > 0 \tag{3.2}$$

Where parameter c is describing the the ability of the driver to react on changing distance during the driving.

Damping force O is dependent on the first derivation of the distance between the vehicles. Therefore, it is reflecting effort of driver to keep the same speed as the previous vehicle. Similarly to the Binding force, this one is dependent on the deviation between first and second vehicle, but to the velocity instead of distance.

$$O = k\frac{d}{dt} = -k\frac{dx}{dt}, k > 0 \tag{3.3}$$

This force increases with increase of the relative velocity which is acting against the acceleration. Therefore the damping force is equal to zero when the vehicles are driving the same speed and Parameter k is describing the ability of the driver to react on changing speed during the driving

Motor forces M is describing a propulsion of the vehicle which variates during the time. This force is not reflecting only the motor as the name suggests but also the breaking system of the vehicle. the force negative in the case of breaking This force is responsible for two main actions:

1. to even friction forces;

2. to accelerate and decelerate when the vehicle is not operating in the platoon.

$$M = M(t) \tag{3.4}$$

Friction force F is describing all the passive forces: friction of all the mechanical elements and a rolling friction of the vehicle reducing the motor power.

$$F = Nf = mg\mu \tag{3.5}$$

N is normal force perpendicular to the surface and the μ is a friction coefficient. A vehicle with a rubber tires reach the values from 0.0062 to 0.015 on a concrete [20].

Aerodynamic drag A_e is another type of friction known as well as the air resistance. This force is acting opposite to the movement of any object through a surrounding fluid.

$$A_e = \vartheta(\frac{dx}{dt})^2\vartheta = c_x S \rho_v/2 \tag{3.6}$$

c_x is a drag coefficient, S is a cross section area of the vehicle - perpendicular to the velocity and ρ is a density of the fluid. For low speeds this force is relatively small in comparison to the friction forces.

3.3 Forces interaction

When we assemble the equation of motion for the second vehicle according to the figure 3.2 we get the following equation:

$$m\frac{d^2x}{dt^2} = -G - O - T - A_e + M \tag{3.7}$$

We can omit aerodynamic drag A_e for two reasons. Firstly it is significant only in the higher speeds, secondly it acts the same at every vehicle in the convoy. After replacing forces in 3.7 by equations 3.2, 3.3, 3.4 and 3.5 and introducing following variables:

$$2A = k/m, B^2 = c/m \tag{3.8}$$

we receive:

$$\frac{d^2x}{dt^2} + 2A + \frac{dx}{dt} + B^2x = \frac{M(t)}{m} - gf = Q(t) \tag{3.9}$$

For sake of simplicity we assume the motor force M is equal to friction force T so we can neglect it: $Q(t) = \frac{M(t)}{m} - gf = 0$ and receive:

$$\frac{d^2x}{dt^2} + 2A + \frac{dx}{dt} + B^2x = 0 \tag{3.10}$$

Equation 3.10 is homogeneous linear differential equation of the second order with constant coefficients which can be solved by Lagrange variation of the parameters. Therefore we can expect a followings solution in the form: $x(t) = Ce^{\alpha t}$, $\frac{dx}{dt} = Ce^{\alpha t} = ax$, $\frac{d^2x}{dt^2} = C^2e^{\alpha t} = a^2x$ with time independent constants $C_1$and C_2.

$$\alpha^2 + 2A\alpha + B^2 = 0 \tag{3.11}$$

$$x(t) = C_1e^{\alpha_1 t} + C_2e^{\alpha_2 t} \tag{3.12}$$

Roots of quadratic equation 3.11 are: $\alpha_{1,2} = -A \pm \sqrt{A^2 - B^2}$. Solution and possible ways how the following vehicle will return to the equilibrium depends on sign of $A^2 - B^2$, therefore we have three following possibilities of the solution (fig. 3.3):

1. $A^2 < B^2$: simple harmonic oscillation - $k^2 < 4mc$ in this setting second vehicle oscillates around equilibrium with period: $T_1 = \frac{2\pi}{B\sqrt{1-A^2/B^2}}$ illustrated on figure 3.3a).

2. $A^2 > B^2$: Over-damped harmonic oscillation - $k^2 > 4mc$ is not oscillating - figure 3.3b)

3. $A^2 = B^2$: Critically-damped harmonic oscillation -$k^2 = 4mc$ is limit figure 3.3b)

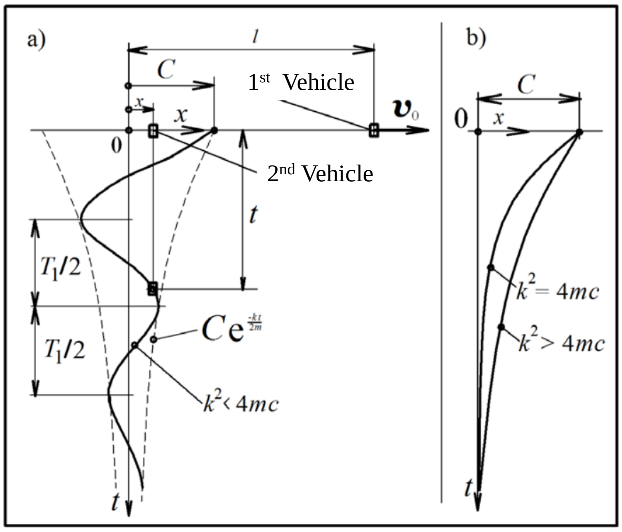


Figure 3.2: Returning of the following vehicle to the equilibrium

3.4 Chapter Conclusion

After reviewing the other sources as well as the proposed theoretical model I have found it possible to implement it in following work.

Mathematical model of Klára Menglerová is possible to implement and examine. In general there are two possible ways how the second vehicle is reaching equilibrium. This equilibrium is defined by safety distance l and safety time t_s. Either it will oscillate around ideal position, or it will reach it slowly. This depends on two crucial characteristics of a driver: ability to keep the safety distance and ability to keep the same speed as previous vehicle.

Chapter 4

Simulation

This chapter is explaining the ways I have implemented the mathematical models described in the chapter 3.2, how are these models connected and how do they interact. During the course of work I have created multiple prototypes and experimented with more approaches. I will mention their differences and challenges I have encountered during my work.

4.1 Modelica

In the course of work I have used two programming languages while working on this master thesis, namely Modelica and Python. Because Modelica is different from another conventional programming language like C, C++, Java, Python, etc. I am going to introduce Modelica's basic principles as well as strengths and weaknesses.

Modelica is in the development process since 1996 by non-profit Modelica Association [1] which is non-periodically releasing official versions of the language. Currently the latest version is 3.4 released in April 2017. The project connects experts, industry professionals, academical workers as well as students of different fields from all over the world [21].

Authors specify Modelica as free, equation-based, object oriented modeling language which has multi-domain modeling capability. Every model has to be determined (number of variables is the same as number of equations) in the order to work correctly [1].

Listing 4.1 is showing basic structure of Modelica model which is stored in the form of .mo files. The model has two main components: the variables and the equations. First

[1] www.modelica.org

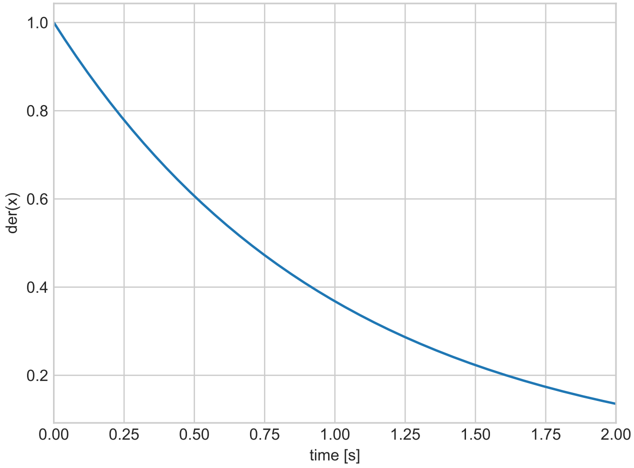
section of the code is listing all the variables which are used in the model together with their properties (input/output, units, quantities, etc.) Second section is collecting the equations which are describing bonds in the model. The concept of equations is another thing which is different from the standard programming languages, the equation doesn't mean assignment, but rather equality of the variables.

The benefit is that the model has no predefined causality so when the Modelica simulation engine processes the file, it also decides on the causality of the system itself.

Listing 4.1: Modelica Hello World [1]

```
model HelloWorld
  Real x(start = 1,fixed=true);
  parameter Real a = 1;
equation
  der(x) = a * x;
end HelloWorld;
```

Figure 4.1: Modelica Hello World output



Modelica processes the model in a following way (figure 4.1): At first Modelica transforms the equations and variables and assigns the causality of the problem based on the inputs and outputs. Then recompiles the model from .mo files into the C code which no longer provides a flexible model, but the the model is assigned to solve a specific system of equations. After this step Modelica generates platform-dependent executable file, which can run without a Modelica environment. Modelica also generates a files for initial settings of the simulation, therefore it is possible to change the simulation parameters and inputs without re-compiling and safe significant amount of time. The executable file produces a results in the form of CSV or MATLAB files.

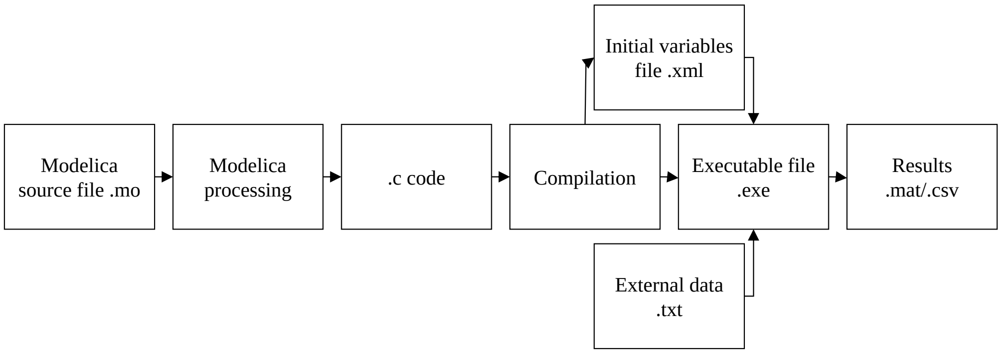


Figure 4.2: Modelica code processing

The general options of the model are start time, stop time and interval or number of intervals. These parameters define the quality of the simulation, therefore the results can differ form another simulation of the same model with different settings.

4.2 Creating a Modelica model

At first, I have created a Modelica model of a vehicle calculating the forces interactions, the safety distance and its own position. To complete this task there are different components - you can see scheme of the vehicle in figure 4.2.

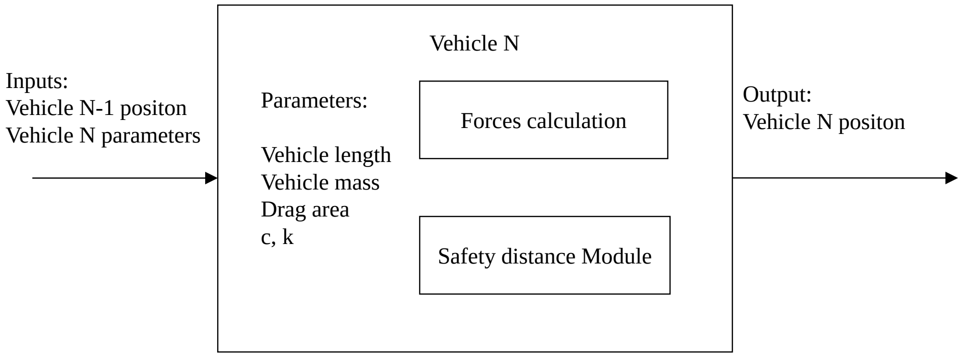


Figure 4.3: Vehicle object scheme

Class Vehicle Driver is containing the key parameters which are representing a person driving the vehicle. These parameters are c (drivers ability to keep the safety distance), k (drivers ability to keep the same speed as the previous vehicle) and safety time.

Class Safety Distance calculates the distance based on the time needed to stop the vehicle. Because the driver has to decide about the safety of the ride this module connects the data from Driver class and uses it as an input of fundamental formula $v = st_s$. The safety time t_s is considered to be 2 seconds for all the simulations. When the velocity decreases under the level of minimal safety distance $l_m in$ this module assigns $l_m in$ distance instead of the distance calculated.

Class Vehicle is the main class containing the two previous classes, parameters of the vehicle (mass, front area, etc.) as well as the equations for the interaction with the previous vehicle. This class is also calculating the initial conditions before the vehicles start moving, because the normal behavior is not to start from the same zero point.

Class Environment is encapsulating the Vehicle class and provides the surroundings of the vehicle, mainly the previous vehicle. This class is also responsible for the loading of the position data and setting every parameter of the simulation correctly and simplifies the manipulation of the data for the further evaluation.

Small differences between the models for different purposes are described in the section 4.4 As well code examples are part of this master thesis in appendix A and B

4.3 Data processing

Chapter 2.2 is discussing two data sets available, the first one is the output of the smart phones and the second one is the output of the C2C devices. I have chosen the data from the dedicated C2C devices, because they had a better coverage, stability and a higher sample rate. This data were not sufficient to use without any modification for the purpose of the simulations. To reach a better quality I have performed some operations such as pairing of the data to vehicle-pairs, re-calculation of the distances, data puring, etc.

Data pairing to obtain the vehicle-pairs. At first I have created a MS Access database and imported the data from all the CSV files. Having data available in the form of database made it easier to filter all the records and match the timestamps using few Structured Query Language (SQL) queries. As you can see in an example of a simple query in the listing 4.2 the database contains one table for every vehicle and the main key is the time-stamp of the vehicle. Matching the same timestamps creates the vehicle-pair for given time period. In my case this period was form 10:03:20 AM (36200) to 10:33:20 AM (380000) I was also trying a different approach with one table for all the vehicles, but this made requesting specific vehicle-pairs too complicated and more time consuming.

Listing 4.2: SQL sample

```
SELECT C2c_01.timestamp, C2c_01.car_id, C2c_01.latitude, C2c_01.longitude, c2c_01.acceleration_slope,
       C2c_02.car_id, C2c_02.latitude,C2c_02.longitude, c2c_02.acceleration_slope
FROM C2c_01, C2c_02
WHERE (C2c_01.timestamp)=[C2c_02].[timestamp]
  AND (C2c_01.timestamp)>36200
  AND (C2c_01.timestamp)<38000;
```

After the extraction of the data-pairs I have decided to work on the datasets with Python where I found two very useful libraries Pandas [2] for data analysis and NumPy [3] for scientific computing. Then I have created two data sets for a vehicle pairs V1-V2, V2-V3, V3-V4, V4-V5, V5-V6, V6-V7:

1. Time - Distance records in the full period; and

2. Time - Distance records without Stationary points (SP).

The second data set without a SP is simulating behavior of the vehicles in a more fluent environment than the stop and go waves.

Calculating distances between the vehicles. At first, the data output from GPS is using a WGS 84 reference system to geocode the data with an accuracy around 1 cm [5] For the calculation of distance between coordinates I have decided to use Great Circle Distance (GCD) method. GCD is simplifying a shape of the Earth to the shape of a sphere and the shortest distances between two coordinates are an arks on the surface of this sphere. This method is appropriate for calculating short distances in the same altitude and surface of the airport is even and the the high sample rate is making the distance between records short. [22] We can review specific equation 4.1, where r_E is

[2] pandas.pydata.org

[3] numpy.org

radius of the Earth assumed 6371 km.

$$x = acos(cos(\frac{\pi}{2}lat_1)cos(\frac{\pi}{2}-lat_2)+sin(\frac{\pi}{2}-long_1)sin(\frac{\pi}{2}-lat_2)cos(long_1-long_2))r_E \quad (4.1)$$

Then I have calculated the distance driven by the vehicles as a cumulative sum of the distances between the records. I have also encountered problem with the cumulative error of the sums, which was causing a differences up to tenths of meters between the vehicles in the vehicle-pair during the whole time of measurement. This error was caused by a jitter in GPS measurements. I have recognized it, because some records of the standing vehicles were moving back and forward by few centimeters when the vehicle was obviously in stable position. To eliminate this error, I have calculated the positions of following vehicles by subtracting the distance between the vehicles from the cumulative distance of the first vehicle. This approach resolved into a more coherent results in the comparison with a two independent cumulative sums. Therefore, the error is same for both of the vehicles and for the purpose of the simulation can be ignored.

Removing the stationary records and creating the second data set. Initially I have intended to perform this in an extended SQL query, which was supposed to ignore the records with the acceleration below 0.3 and above -0.3. But the problem was the flexibility of this solution, because the Microsoft Access was hard to access from any other program than MS Office. Even though Microsoft provides some interfaces for the programmers and developers I was not successful with connecting Python and MS Access. The reason to have flexibility was to have a possibility to easily change the range in the queries while I was finding the best threshold where the stationary data are sensitively removed without removing too much of the dataset in motion. Therefore I have performed the whole procedure in the Python. After removing data, I had to re-sample the timestamps, and keep the timestamps in the gaps created by the combination of the pairing and the missing data. To do this I have measured the largest gap before removing he stationary data set and shifted every other gap bigger than the first one after removing the SP.

Minimal distance between the vehicles - the last thing to calculate before I could start with the simulations was l_min - minimal distance between the vehicles as I described previously 3.2. Because in this theses I have considered every vehicle as a point particle with a given length l_v, then the minimal safety distance l_min can be defined as half of the length of first vehicle plus half of the length of the second vehicle and complementary distance l_c

$$l_min = l_v1/2 + l_v2/2 + l_c \tag{4.2}$$

Complementary distance l_c is describing real-life situation when drivers keep additional distance in front of their own vehicles.

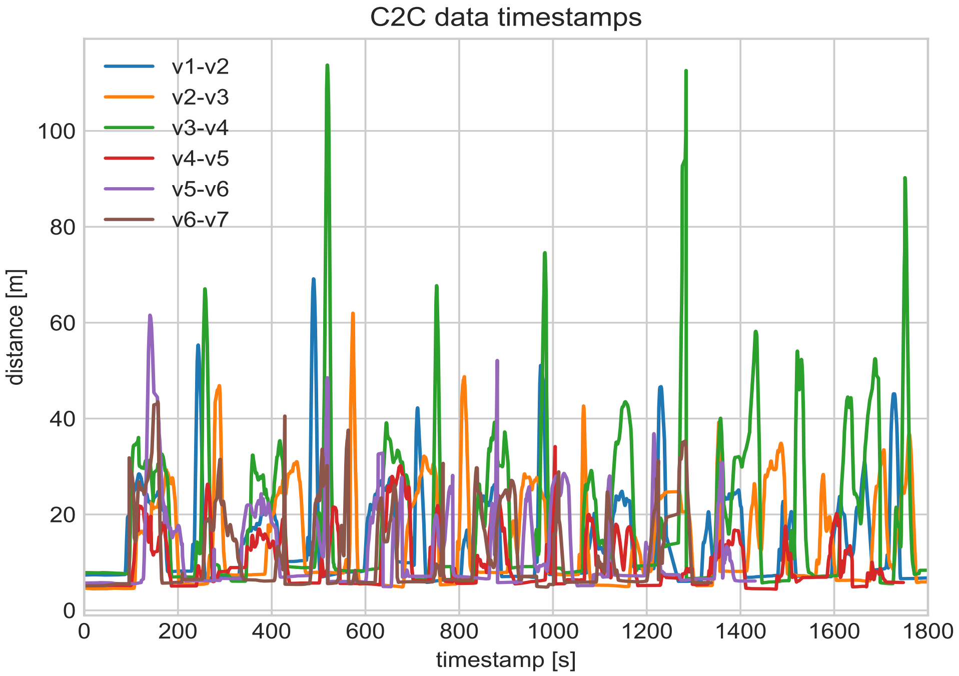


Figure 4.4: Minimal distance between the vehicles

In the figures 4.3 and 4.3 you can see a course of the distances for vehicles v1 - v7 and in the table 4.3 specific values extracted from the comparison. Values in column 'estimation' were calculated based on equation 4.2 with $l_c = 0$. This means that the values in column 'difference' correspond to the l_c values. When we compare the results, we can see there is no general dependence rule connecting the values. On the other hand the sample is not big enough to make any conclusions form it, the expected length of the vehicle can vary from the real value, error of GPS is not considered and the experiment was not aimed to explore minimal distances between the vehicles. Good indicator of respectable results is the fact that no value of l_c is below 0, which would mean collision. For further simulations I have used the values of measured distances between the vehicle-pairs.

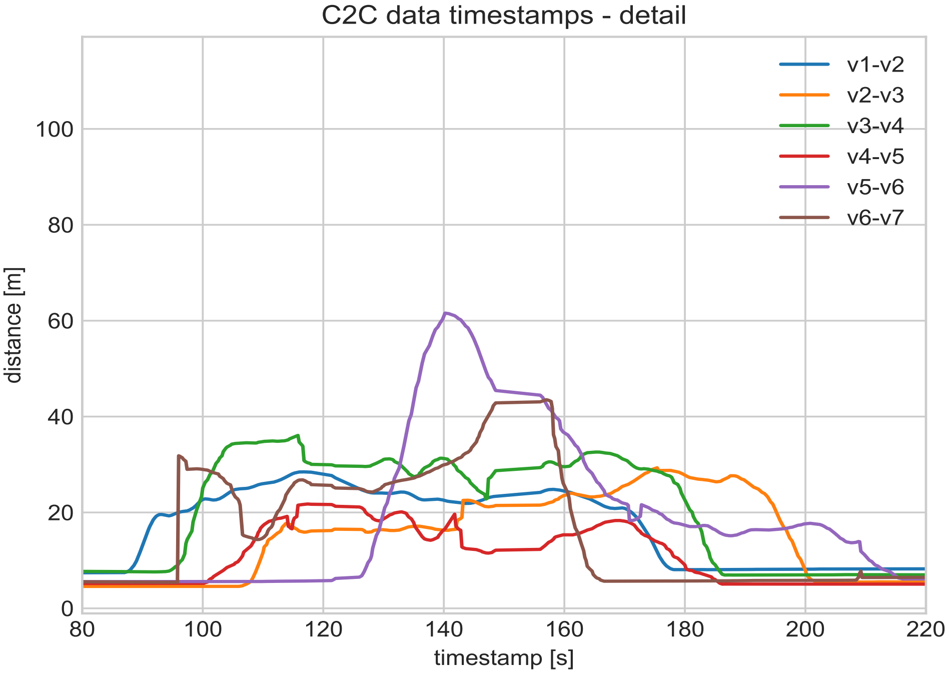


Figure 4.5: Minimal distance between the vehicles - detail

vehicle-pair	measured	estimation	difference
-	m	m	m
v1-v2	5.831	4.443	1.388
v2-v3	4.453	4.194	0.259
v3-v4	5.525	3.999	1.526
v4-v5	4.408	4.096	0.312
v5-v6	4.906	4.283	0.623
v6-v7	4.824	4.458	0.366

Table 4.1: Minimal distances between the vehicles

4.4 Optimum search

In this part I would like to describe the methods how I was simulating the traffic conditions based on the data from the Coflow experiment, which is presented in a greater detail in the chapter 2.1. To simulate the movements with the best accuracy it is important to use the parameters of the driver, which are describing the correct driver behavior. To find the best matching c and k parameters I have developed three methods:

1. Searching c and k coefficients by searching span area;
2. Searching c and k coefficients using heuristic; and
3. Reversed c and k coefficients search.

First two approaches are similar to each other. The handling program proceeds the simulation in a multiple cycles with a different set of parameters, then it is compared with the real data set. After one cycle the set of parameters changes and proceeds to another simulation and so on. Results of these simulations are described in the next chapter together with a broader explanation of the display method.

On the other hand the third one is fully using the capabilities of Modelica to extract the parameters c and k from the both trajectories of the vehicles.

4.4.1 c - k search in span area

The first method I have used for searching the c and k parameter was scanning of the domain of a function and evaluating the correlation with the real data. The difference between the *basic* and advanced simulation is that the basic is using the same parameters for all the of the vehicles (table 4.4.1) and safety time of 2 seconds. This was a simple starting point to start with all the simulations.

Vehicle-pair	detail	length	mass
-	-	mm	kg
V1-V7	-	5000	2000

Table 4.2: Basic vehicle model

On the other hand the *advanced* is reflecting the real minimal distances and vehicle lengths (tab. 2.1, tax 4.3). The outcome of this differences are described in the next chapter.

Time complexity of the problem was another parameter I have considered when I started to evaluate the efficiency of the simulation.

As I described previously the complexity of one simulation is dependent on the scale of the simulation. One run of the simulation can take take different amount of time, for example with settings of simulation time = 1000s and interval = 1s, 100 simulations takes 23.558 seconds. With setting of simulation time = 1700s and interval = 0.2s 100 simulation takes 81.217 seconds (this setting was usual for a simulation with SP. Simulations without SP had set interval = 0.2s and time around 850s. In this setting 100 simulations takes 45.891 seconds.

This means iterating through 1000 points takes at least 450 seconds / 7,5 minutes for input set without the SP, 1450 seconds / 24 minutes for set with the SP.

4.4.2 Finding c and k parameters with a heuristic

In the second part of the searching of the c and k values I have decide to use a heuristic to fasten the procedure of finding the best fitting parameters and to increase the quality of the results I have chosen this heuristic based on the results of the basic span search, where I have made a following assumptions:

- The $c - k$ plane has one global maximum;
- The distribution of the correlation is mostly linear.

Having this in mind I have decided to create a heuristic, which can find the highest point in the E3 space without knowing the complete information about the distribution of the correlation. As a surface I am assuming area defined in three dimensional space formed by c, k and $corr$ - correlation between the simulated and measured situation.

The algorithm can be described in a four following steps:

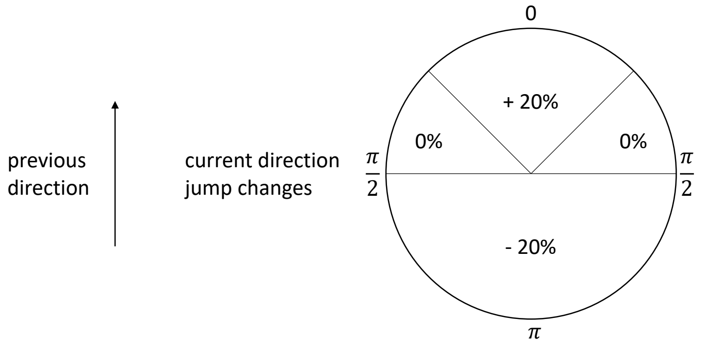
1. Create a three different points in the $c - k$ coordinates;
2. Calculate a correlation of each point;
3. Calculate the steepest slope and its direction;
4. Adjust distance to move to a new location and move in the direction of the steepest slope; and

5. Repeat until the points are jumping around the peak.

In the beginning the algorithm has no information about the surface, so it calculates a three points receiving the first information around the starting point. Based on this plane the next points which the algorithm is going to calculate have the direction of the steepest slope. In this direction I have assumed the biggest gain for the next simulations. Before the algorithm moves to the next position it calculates the deviation from the previous direction. If the direction is similar to the previous one, the algorithm increases the length of the jump. If the previous direction is pointing to the sides or backwards this means that the algorithm had missed the point of higher correlation and therefore reduces the length of the jump. The simulation ends when the jump is lower than the threshold, or after maximal number of the steps.

Parameters of the heuristic were set after multiple experiments as follows: The starting point of the simulation is [100 100 0], distance between the points of the triangle was 10 $kg^{-1}s^2/kg^{-1}s$, minimal length of the jump was set to 3, maximal number of cycles was 100. As you can see no the figure 4.6 the adjustments of the jumps are +20% in the range $\pm\pi/4$, the same in the range $(\pm\pi/4 - \pm\pi/2$ and -20% from $\pm\pi/2$.

Figure 4.6: Estimating c and k constants - without stops



On the figure 4.4.2 you can see the development of the jump lengths in the course of the search of the optimum. There is a pattern repeating in the all of the runs - in the beginning the size of the steps increases as the heuristic is on the right course to the maximum. After first miss of the maximum point the heuristic has to change the course more dramatically as well decrease the jump length.

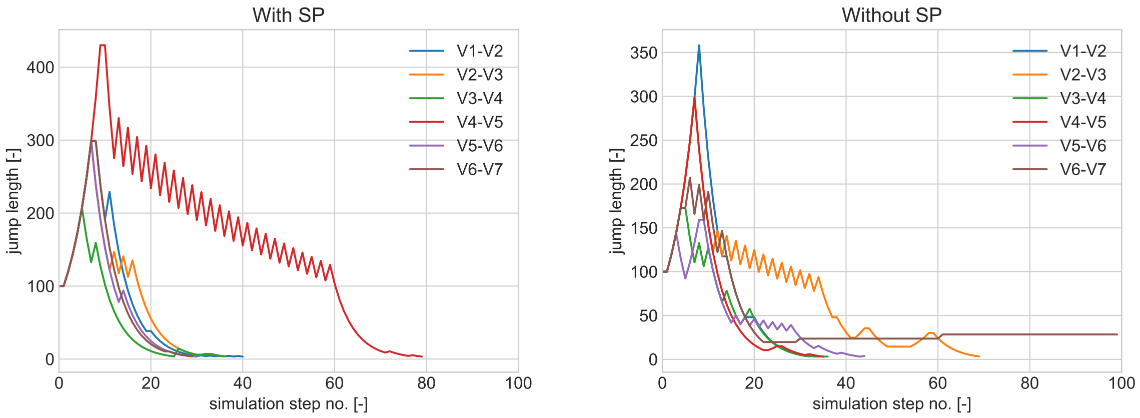


Figure 4.7: Development of the jump length between the steps

Time complexity of the problem is also an issue for this approach. The heuristic is consisting of the same founding block - setting of the simulation parameters and the simulation itself - which takes approximately same time amount of the time as the previous search method (4.4.1). Over this block lays another layer which is setting the position of the points, evaluating the slope and recalculating the distance between the old and the new simulations. This procedures takes a less time than a variance of an external simulations runtime, therefore it can be ignored. This means that every step of heuristic takes time of three simulations. On figure 4.4.2 you can see that the shortest heuristic (with SP) took 30 steps / 90 simulations / 73 seconds. On the other hand the the longest one (without SP) took the maximum of 100 steps / 300 simulations / 135 seconds.

4.4.3 Reversed c and k coefficients search

Last method I have used and implemented is Reversed c and k coefficients search. This is not a conventional method, but rather usage of Modelica simulation engine which is able to proceed slightly adjusted models described in the beginning of this chapter (fig. 4.2). The difference is in the usage position of previous vehicle and position of current vehicle as an inputs together with vehicle length and vehicle mass to calculate c and k.

One challenge I have encountered while implementing this method is determination of the model (number of variables is equal to number of equations). Because of the change, the model was not calculating only one output (vehicle position), but two (c and k), therefore the model was underdetermined. This resulted in testing only one parameter while c or k was a constant. After several tests I have decision to introduce one more equation to bind c and k together. The problem is which parameter to choose. I have been experimenting with ration ($ratio = c/k$), subtraction ($c = k - subb$) and derivation ($c = der(k)$). All of the results were of similar nature and shared similar imperfection.

Usability of this method is well summarized on the figure 4.4.3. As you can see the resulting c parameter is not stable and its values are in the range $< -2700; 28000 >$ (not all the maximums are displayed on the left). The mean of c is 15.51 and variance 273382.32. This means that the model of the driver is not stable during the time of the simulation, which is actually real representation of the driver. Unfortunately previous model needed a constant coefficients. I have overcame this challenge by loading c and k form the file and smoothing them with moving average, but the question stays: how to obtain this data in prior the simulation when the real data are not available.

Because of this reasons I have not extend the work on this problem, but there are suggestions how to improve it in future work in the conclusion of this thesis (chapter 6).

Time complexity of the problem of one run of dataset without SP takes 0.8 second and of the dataset with SP takes 1.85 seconds to proceed. There is no need for another runs. This is the lowest time demanding application from the three introduced in this chapter.

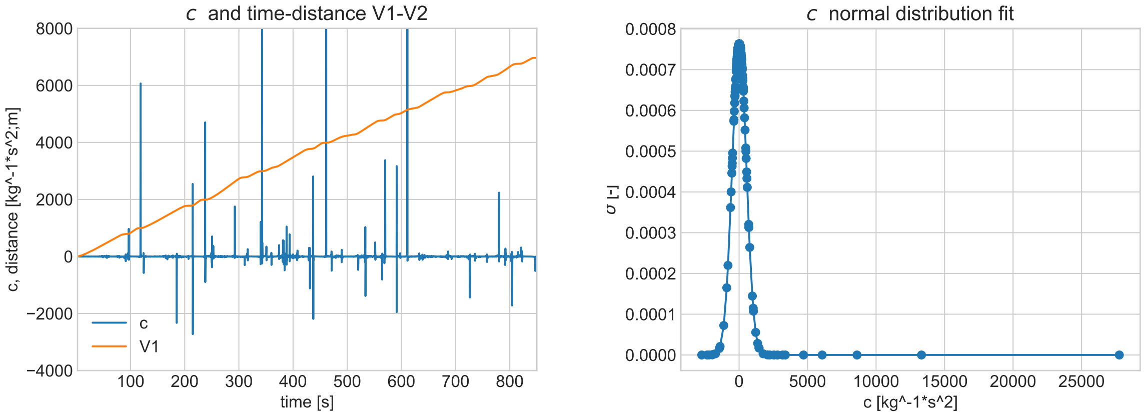


Figure 4.8: c in reversed search - without SP

4.5 Chapter conclusion

I have implemented the model introduced in chapter 3 and adjusted the available data sets for purposes of the simulations. One data set for stop-and-go traffic flow and second for more fluent representation of the traffic flow.

Also I have completed three methods how to find proper coefficients of the model. First two methods are aiming to find constant values which fit the best to the driver by simulating with different settings and comparing it to the real state. The third one is extracting qualities of the driver from both his ride and previous ride.

Results of the first two are discussed in the following chapter. Problematic of the third one is described only here and future possibilities in the conclusion of this thesis.

Chapter 5

Results Comparison

This chapter is presenting the output of the simulation methods described in previous chapter 3. I have divided the results into two subgroups based on the inputs.

- with SP; and
- without SP.

These groups are not comparable with one another. Group one with SP is a slightly edited output of the experiment describe in the chapter 2 and the second group is imitating a more fluent traffic flow.

As the domain of the function I have chosen range of $c = 0 - 2000kg^{-1}s^2$ and $k = 0 - 5000kg^{-1}s$ for simulation of vehicle-pairs V1-V2, V2-V3, V3-V4, V4-V5 and range of $c = 0 - 2000kg^{-1}s^2$ and $k = 0 - 9000kg^{-1}s$ for V5-V6, V6-V7. And as a simulation step I have chosen 100 for both c and k. This results in 1000 tests for one vehicle-pair, respectively 1800 for the V5-V6 and V6-V7.

In the figures showing results of simple search (basic and advanced) I have decided to show only the top 2% of the results, because showing the whole domain makes the figures unreadable.

5.1 Display method

For a better understanding of the figures in the section 5. I would like to introduce the logic behind them. The main goal these figures is to summarize the output of numerous simulations with different parameters. Every figure is displaying 3 variables $c[kg^{-1}s^2]$,

$k[kg^{-1}s^2]$ and correlation. On the axis x is the coefficient c and on the axis y is the coefficient k, the dots displayed on the coordinates shows the correlation of one simulation.

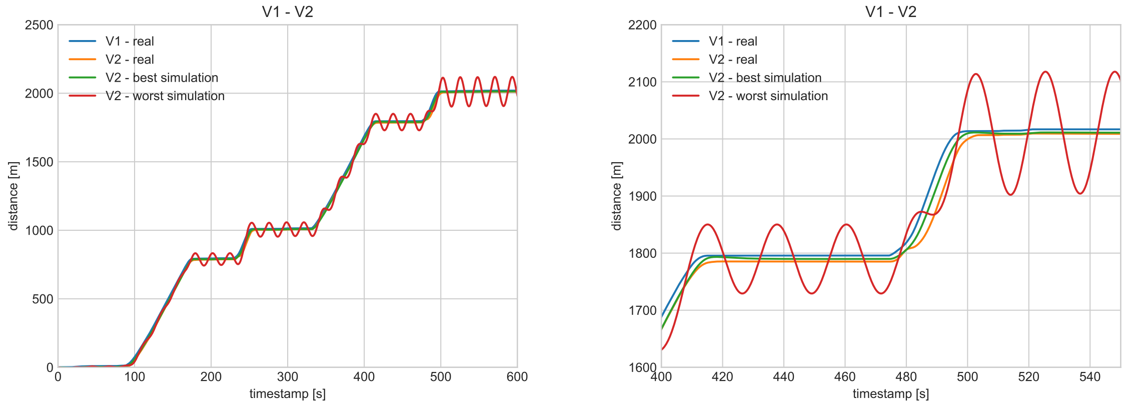


Figure 5.1: Simulation of the V1-V2 - with the stationary points

Normalized Correlation in the figures of this chapter there are a dots marked as a corr (result of PCC as I have described previously in chapter 3.1). The correlation is normalized to the scale $< 0 - 1 >$ from the range of the results displayed. The real, non-normalized, differences of the correlation are in the orders of magnitude from 10^{-8} to 10^{-6}

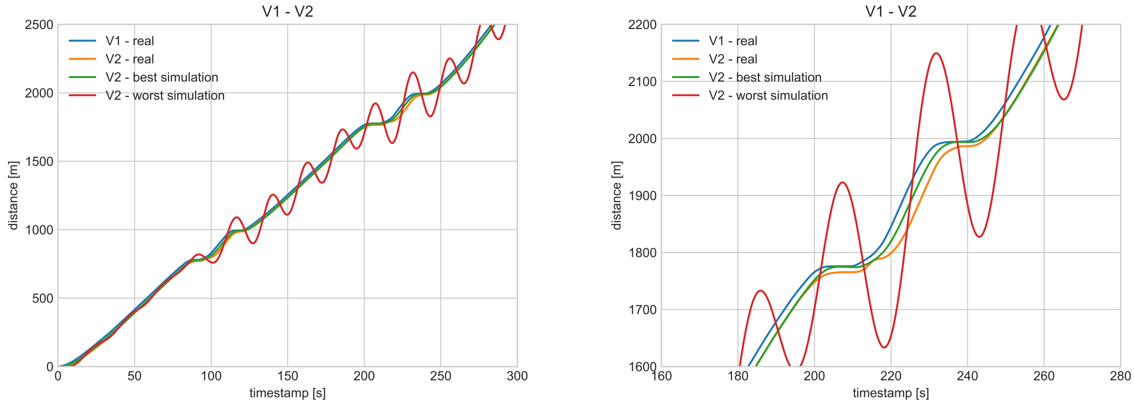


Figure 5.2: Simulation of the V1-V2 - without the stationary points

In the figures 5.1 and 5.1, you can see example of the simulations. I have chosen to demonstrate the simulation of vehicle pair V1-V2 with parameters $mass_{V2} = 1300kg$ and $length_{V2} = 4.526m$, based on the exhaustive table 2.1 with all the vehicles. The figure 5.1 is consisting of standard input withSP and two output of simulation, the 'best'

[1] results with the parameters $c = 100kg^{-1}s^2$ and $k = 800kg^{-1}s$ and the worst result with parameters $c = 100kg^{-1}s^2$ and $k = 0kg^{-1}s$ (the worst and the 'best' parameters calculated in our scope) The other figure 5.1 is displaying similar conditions, but for filtered inputs without SP. The 'best' parameters of the simulation were $c = 100kg^{-1}s^2$ and $k = 700kg^{-1}s$ and the worst are $c = 100kg^{-1}s^2$, $k = 0kg^{-1}s$, the table 5.1 shows summary of the parameters. On the figures you can see the nature of the results: the 'best' it is clearly visible, that the results with the worst parameters are undamped oscillations.

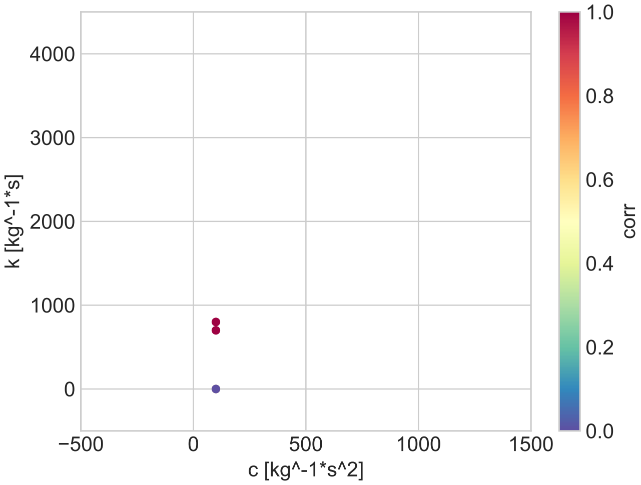


Figure 5.3: Simulation of the V1-V2 - results representation

simulation	c	k	corr
-	$kg^{-1}s^2$	$kg^{-1}s$	-
with SP- best	100	800	0.999997106
with SP- worst	100	0	0.998875717
withouth SP- best	100	700	0.999994051
withouth SP- worst	100	0	0.996791925

Table 5.1: Finding c-k simulation parameters

On the figure 5.1 you can see the results of the simulation demonstrations which are based on the results listed in the table 5.1. I have to mention that I am aware of combining results from two different data set is not a correct method, but the figure 5.1 is for demonstrative purposes only. You can see one of the problems being overlapping of the displayed points - $c = 0$ and $k = 0$ should be displayed twice as there are two results

[1] Referring to the results as the 'best' the local maximum reached by the method introduced in the section 4.4.1 not the best as global maximum

for two different simulations. The results does not have this this problem, because there are not two results in the same point for one simulation method. Even if there would be they would have the same results.

5.2 With Stationary Points

Basic and advanced simulations were the first simulations I have performed with purpose to make the first assumptions about the nature of the simulations. For vehicle-pairs V1-V2, V2-V3 and V3-V4 (figures 5.2 and 5.2 you can see approximately linear dependencies of the maximum range. Starting with V1-V2 linear approximation is $k = 1.5395c + 1053.4$ with $R^2 = 0.9504$ the range $c = 0 - 17500 kg^{-1}s^2$ (see complete list of linear approximation on the table 5.2) In these three situations are the maximums around the point [100 1000]. This means that the both 'strings' are functioning in the model and are influencing the movement of the second vehicle and the model works as intended.

The last row of the table is displaying if the simulation results tends to have oscillations according to the chapter 3 ($A^2 < B^2$: simple harmonic oscillations; $A^2 > B^2$: over-damped harmonic oscillations; and $A^2 = B^2$: critically-damped harmonic oscillations)

		basic		advanced		
Vehicle-pair	correlation	c	k	dependency	r^2	$A^2?B^2$
-	-	$kg^{-1}s^2$	$kg^{-1}s^2$	-	-	-
V1-V2	0.999996736	100	1100	$k = 1.5395c + 1053.4$	0.9504	$\approx$
V2-V3	0.999997808	0	1100	$k = 1.448c + 1246.3$	0.9278	$\geq$
V3-V4	0.999989519	0	800	$k = 2.1548c + 828.89$	0.9705	$\approx$
V4-V5	0.999999216	0	1800	-	-	$\gg$
V5-V6	0.999994549	0	1100	-	-	$\gg$
V6-V7	0.999996882	0	1100	-	-	$\gg$
V1-V5	0.999997808	0	300	-	-	$\gg$

Table 5.2: Estimations of the c and k dependency

On the other hand for the vehicle-pairs V4-V5, V5-V6 and V6-V7 you can see the best results grouping around $c = 0kg^{-1}s^2$ this can be from two reasons:

- Search wants to reach negative coefficients; or
- With current setting of the simulation it is impossible to use one of the strings.

If the solution would be in using negative coefficient it would break the model, but it didn't happened in the previous three vehicle-pairs. Therefore I assume that the setting of the vehicle mass and safety distance is incorrect (spring described by c is trying to reach zero point of distanced between the vehicles + safety distance). This conclusion is obvious when you compare the basic and *advanced* results.

To support this assumption I have created model of vehicle V1 - V5 (5.2 where the safety distance is a sum of all the previous distances and safety time is a sum of the safety times. The simulation has the same problem as the three previous ones.

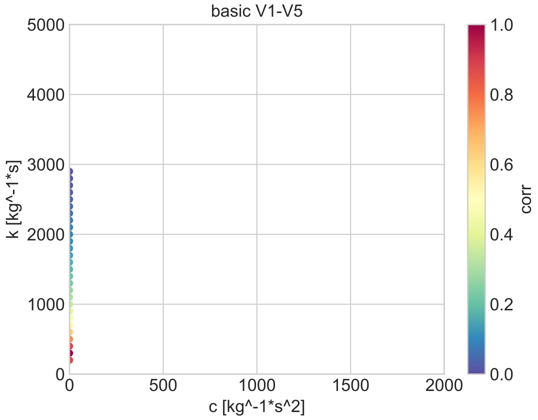


Figure 5.4: Estimating of the c and k constants - basic model b)

Heuristic was another approach how to reach the same points of the maximum as in the previous method. I have placed it next to the c-k search to show how is the algorithm reaching the maximum. Every time it starts from point [100 100] usually doesn't take the shortest way. This means there is a local minimum in between those places which is the heuristic trying to avoid.

There is as summary in the table 5.2, where diff a) is displaying differences between the heuristic and the basic search and diff b) difference between heuristic and advanced search. Positive number means better result, negative worse.

	basic			advanced			
Vehicle-pair	c	k	corr	c	k	corr	diff
-	$kg^{-1}s^2$	$kg^{-1}s^2$	-	$kg^{-1}s^2$	$kg^{-1}s$	-	-
V1-V2	100	1100	0.999996736	100	800	0.999997106	$4{\cdot}10^{-7}$
V2-V3	0	1100	0.999997808	0	500	0.999998094	$3{\cdot}10^{-7}$
V3-V4	0	800	0.999989519	0	300	0.999985503	$-4{\cdot}10^{-6}$
V4-V5	0	1800	0.999999216	0	1100	0.999999280	$6{\cdot}10^{-8}$
V5-V6	0	1100	0.999994549	0	800	0.999995147	$6{\cdot}10^{-7}$
V6-V7	0	1100	0.999996882	0	700	0.999996907	$2{\cdot}10^{-8}$

Table 5.3: Summary of the results with SP a)

Vehicle-pair	c	k	corr	diff a	diff b
-	$kg^{-1}s^2$	$kg^{-1}s$	-	-	-
V1-V2	38.15	729.00	0.999997239	$5{\cdot}10^{-7}$	$1{\cdot}10^{-7}$
V2-V3	283.11	1109.29	0.999998030	$2{\cdot}10^{-7}$	$-6{\cdot}10^{-8}$
V3-V4	0.05	460.24	0.999983871	$-6{\cdot}10^{-6}$	$-2{\cdot}10^{-6}$
V4-V5	2.31	1301.92	0.999999195	$-2{\cdot}10^{-8}$	$-9{\cdot}10^{-8}$
V5-V6	10.13	866.36	0.999995130	$6{\cdot}10^{-7}$	$-2{\cdot}10^{-8}$
V6-V7	4.15	825.19	0.999996748	$-1{\cdot}10^{-7}$	$-2{\cdot}10^{-7}$

Table 5.4: Summary of the results with SP b)

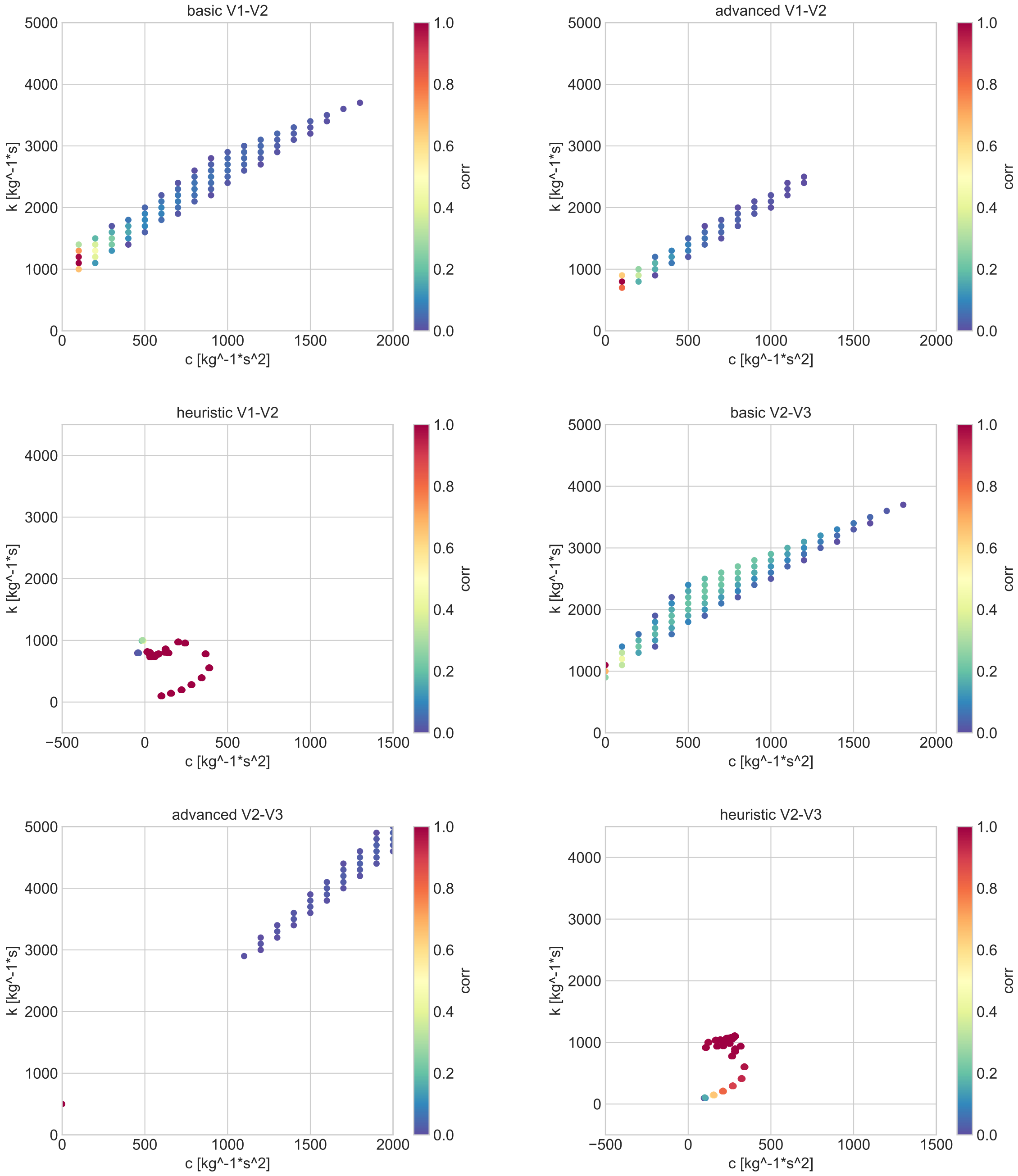


Figure 5.5: Searching the optimal $c-k$ V1-V2, V2-V3

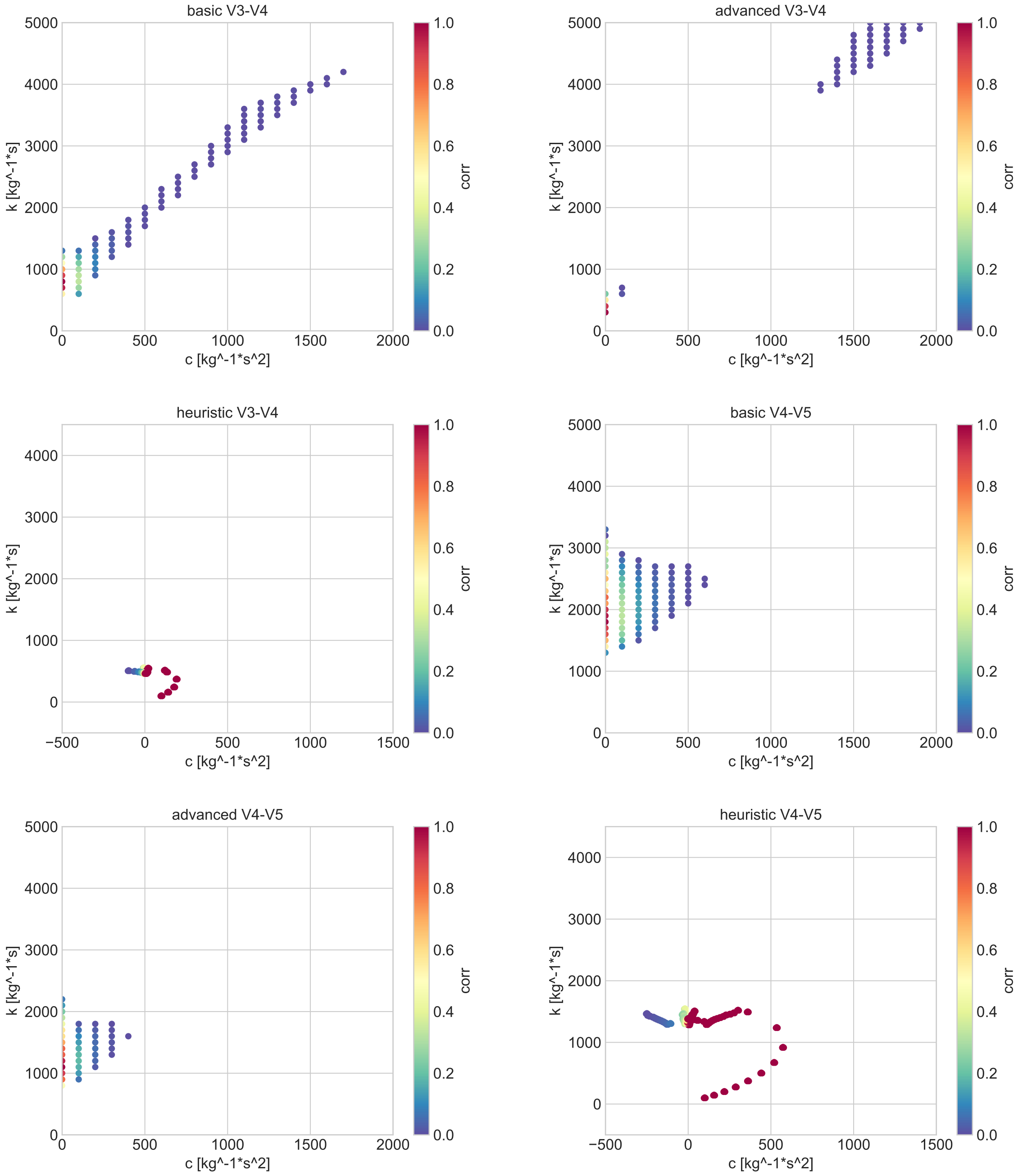


Figure 5.6: Searching the optimal $c - k$ V3-V4, V4-V5

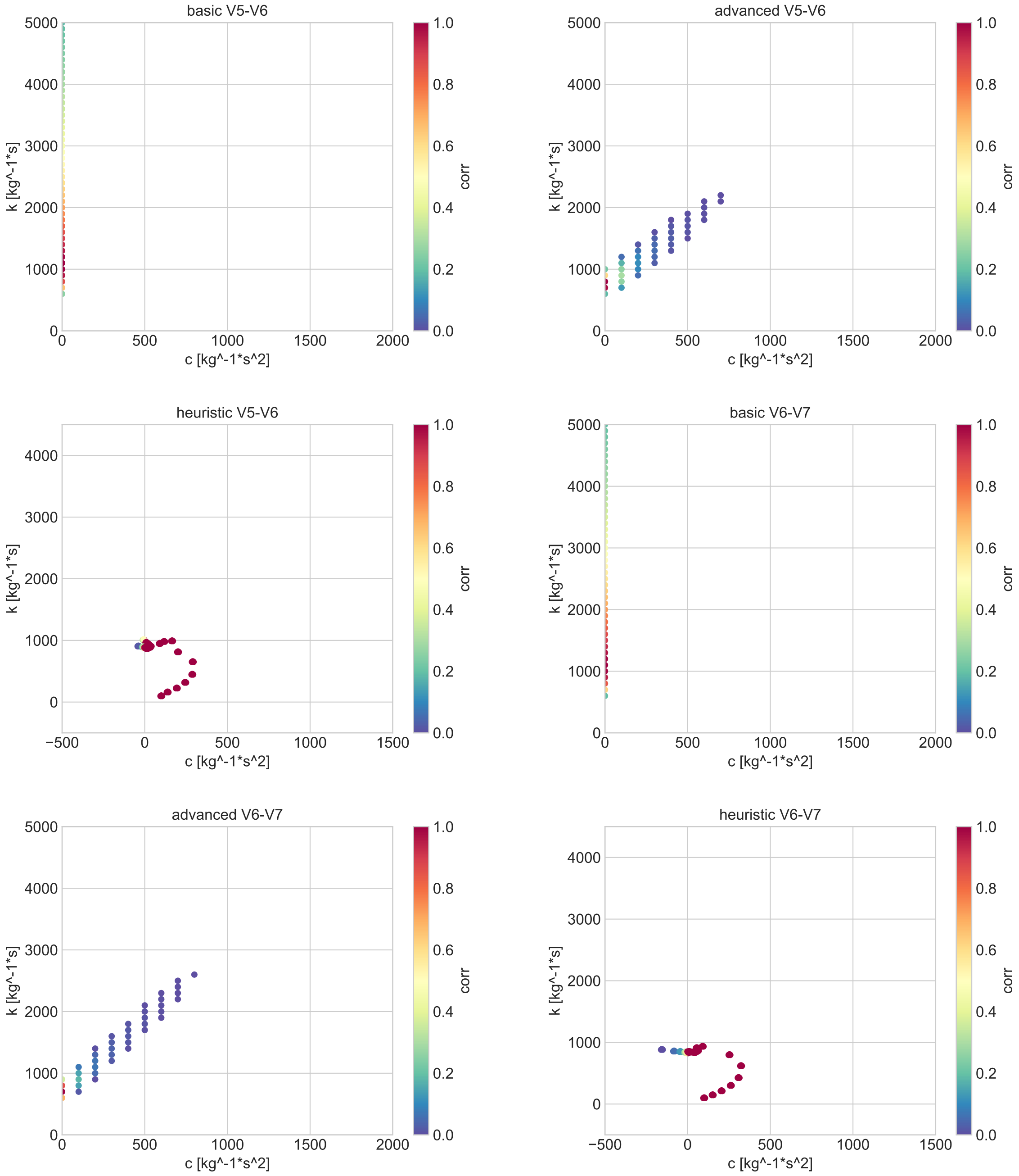


Figure 5.7: Searching the optimal $c - k$ V5-V6, V6-V7

5.3 Without Stationary Points

As I have mentioned earlier I have created this data set to compare the influence of the SP / stops on the quality of the simulations. Chapter 4.3 is describing the whole process of creating of this data set.

You can also see local maximum which is disconnected from another maximum - vehicle pair V3-V4 (fig. 5.3 bottom left). This fact contradicts my original assumptions for usage of the heuristic ($c - k$ plane has one global maximum; there distribution of the correlation is mostly linear). Because it means that the heuristic can finishing the search in a local maximum instead of a global one without possibility to assure the results.

Differences between dataset without SP and with SP. You can see the biggest differences on the example of vehicle pair V5-V6 and V6-V7 (fig. 5.3 left). The coefficients moved from $c = 0$ (and ignoring the distance between the vehicles) towards using this coefficient with good results. This means that I have reduced the error of the evaluation based on the different safety distances when the vehicles were standing still.

Heuristic in this dataset shows also some unexpected behavior on vehicle-pair V6-V7 (fig. 5.3 bottom right). Instead of stopping around the point [0 1000] it keeps going on towards the maximum which was found by advanced search - but instead of finding a direct way it keeps jumping back and forward in an angle around 0.8π which was unexpected. Because I have set the maximum number of steps to 100 the heuristic ended in the point This was a surprising result because I was expecting heuristic going more directly to the maximum. To complete the heuristic for V5-V6 I have made additional testing and found out that the maximum was reached in the point $c = 3080kg^{-1}s^2, k = 27716kg^{-1}s$ with $corr = 0.999993531$ (fig. 5.3) which is better by $2 \cdot 10^{-6}$ from the result of limited heuristic and better by $7 \cdot 10^{-7}$ in compare with advanced search. This figure is also in smaller scale so you can see the individual triangles generated during every step of the heuristic.

There is as summary in the table 5.3, where diff is displaying differences between the heuristic and the advanced search. Positive number means better result, negative worse.

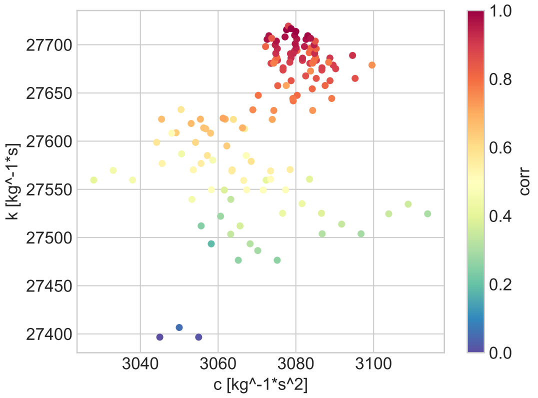


Figure 5.8: Searching the optimal $c-k$ V6-V7; heuristic - completed

	Simple Search			Heuristic			
Vehicle-pair	c	k	corr	c	k	corr	diff
-	$kg^{-1}s^2$	$kg^{-1}s$	-	$kg^{-1}s^2$	$kg^{-1}s$	-	-
V1-V2	100	700	0.999994051	90.12	702.24	0.999994057	$6 \cdot 10^{-9}$
V2-V3	0	500	0.999996443	35.27	584.46	0.999996499	$6 \cdot 10^{-8}$
V3-V4	0	400	0.999976504	30.57	331.48	0.999978598	$2 \cdot 10^{-6}$
V4-V5	0	1400	0.999997579	-1.10	1142.79	0.999997592	$1 \cdot 10^{-8}$
V5-V6	600	9000	0.999988503	-0.12	1005.47	0.999980505	$8 \cdot 10^{-6}$
V6-V7	1000	9000	0.999992858	195.32	2307.77	0.999991490	$1 \cdot 10^{-6}$

Table 5.5: Summary of the results without SP

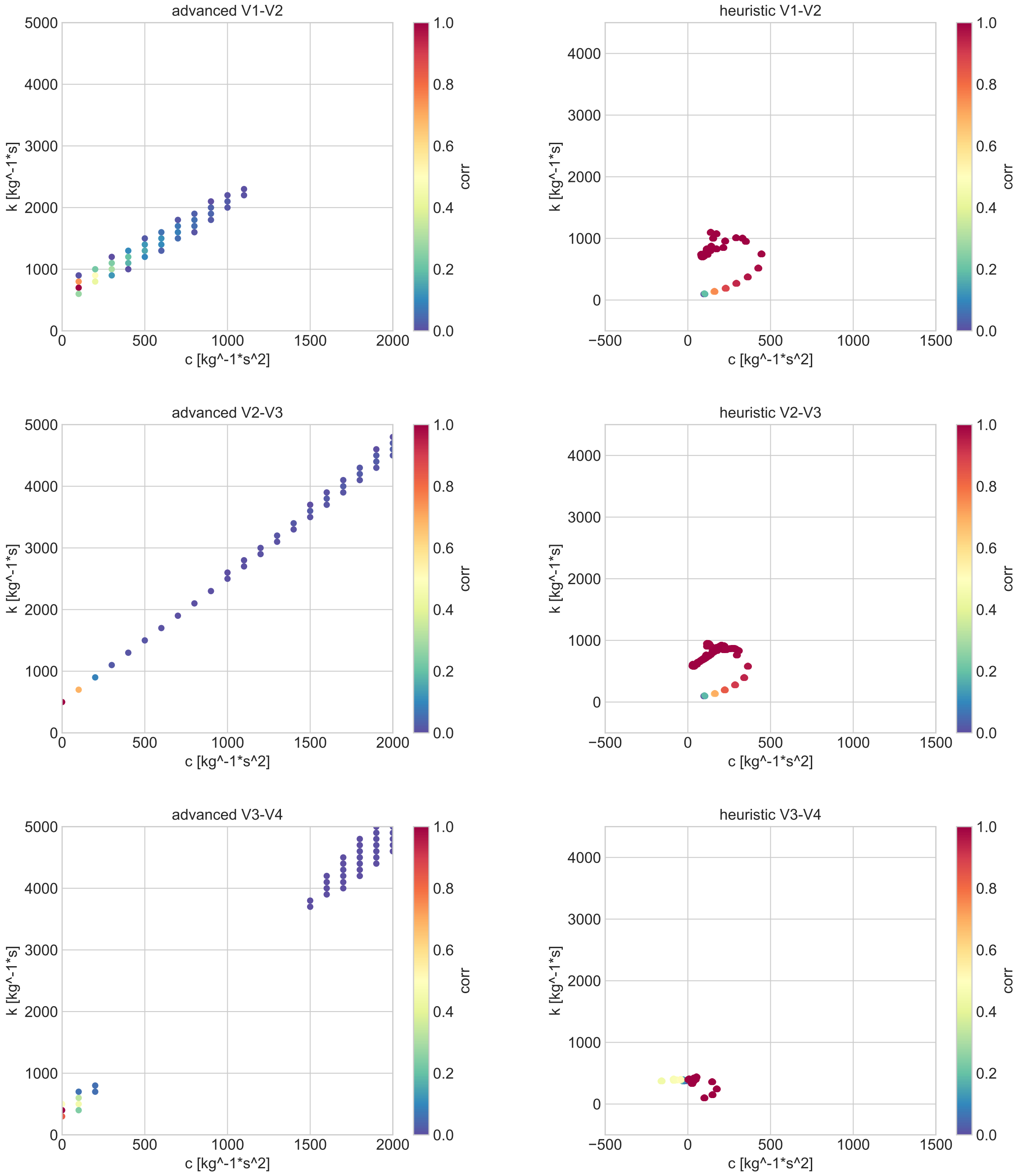


Figure 5.9: Searching the optimal $c - k$ V1-V2, V2-V3, V3-V4

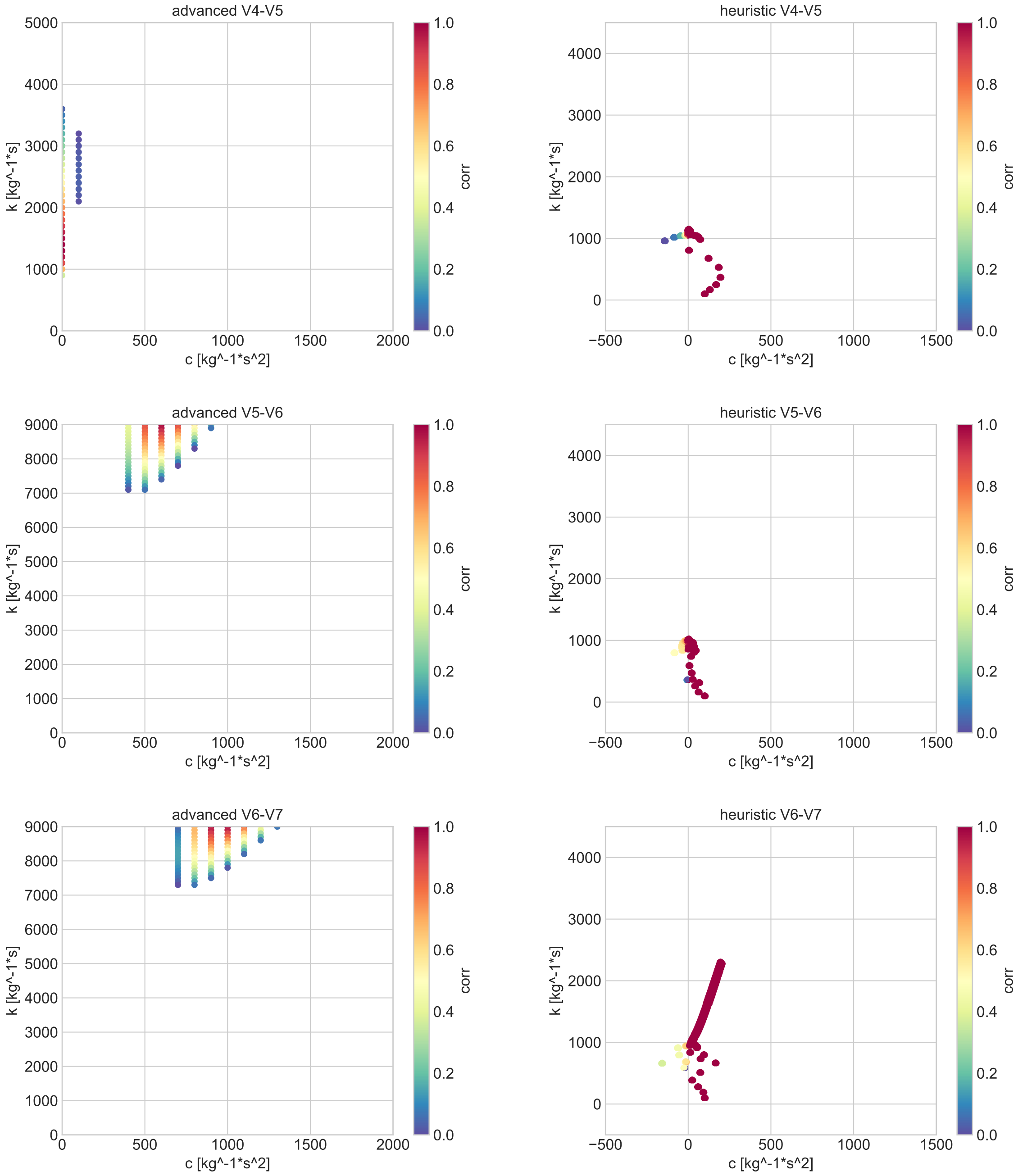


Figure 5.10: Searching the optimal $c - k$ V4-V5, V5-V6, V6-V7

Chapter 6

Conclusion

6.1 Summary of thesis

To sum up the whole thesis I would like to start with the summary of the work I have done in the individual chapters.

In the chapter 2 I have summarized the course and the output of the Coflow proof of the concept experiment which took place on 9th December 2016. Then I have analyzed the outputs of the experiment and chosen one data set for further usage in this thesis.

In the chapter 3 I have summed up the traffic modeling techniques, especially variety of the CFM. Then I have described the model developed by Klára Menglerová - the former PhD student of our faculty - FD CTU and I have described how am I going to use the model in this master thesis as well.

In the chapter 4 I have described the Modelica simulation language - the core of the simulations in this thesis. Then I have describe how I have implemented the mathematical model from the chapter 3. Furthermore I have presented three ways how to discover the qualities of the driver driving in the vehicle-pair. The last one - Reversed c and k coefficients search (chapter 4.4.3) is a quick, but does not provide easy insight into the drivers behavior. The problem is the variety of the results ad change of the behavior through the time. Although this method deserves more attention in the future work.

In the chapter 5 I have shown the results are dependent on the properties of the driver and the vehicle. Also I have shown that the model does not seem to be suitable for stop and go waves or speeds close to zero, because it is trying to put drive in the reverse - which is not reflecting the reality - and it also struggles with stationary moments. Span search is not an efficient method but it provided good starting point for the understanding of the behavior. Furthermore I have shown that suggested heuristic was decreasing the time needed for finding of the optimum, but it was also behaving in unexpected ways and has to be improved before further use.

Having this said the thesis embraces the problem from the orientation in the problematic in the beginning to the listing of the results in the end.

6.2 Fulfillments of targets

When I have started with this master thesis I have set five following goals:

1. Review of the existing modeling approaches;
2. Design the model and the evaluation criteria;
3. Evaluate experimental traffic flow data;
4. create the simulation of the traffic flow; and
5. Compare the results of the simulation and the experiment, discuss the results.

From the summary in the previous section you can see that I have fulfilled all of them. Besides these goals I have made great progress in the evaluation of the drivers behavior which was the task I have underestimated in the beginning of the thesis. During the time I have discovered this problem really interesting and opening many new possibilities in the field of the traffic simulations. The goal of this thesis was reached.

6.3 Further extensibility and recommendations

During the course of the work I have encountered many interesting challenges which were close to the scope of this master thesis, although they were out of the reach of my knowledge or my time possibilities. Therefore, there are many ways how to extend this master thesis, namely:

- obtain different traffic flow data with more fluent characteristics and confirm or refute the hypothesis that the model is going to perform with better results;

- Improve the heuristic for even faster and more reliable search of the optimum.
- Explore the behavior of the model in further detail and the phenomena of the local/global maximums and minimums.
- Extend the testing area and search for the patterns in the drivers behavior, using advanced methods such as artificial intelligence, neural networks etc.
- Apply this model for a longer convoy of vehicles.
- Introduce variable springs changing its stiffness of the springs based on the speed of the vehicle or any other variables.

All these tasks are awaiting for another student or a researcher and my hope is that one day all these questions are going to be answered.

Appendix A

Modelica Code Sample

Listing A.1: Modelica Class BoundVehicle

```
model BoundVehicle
  //---------------forces on vehicle
  Real G(quantity = "Force", unit = "N");
  Real O(quantity = "Force", unit = "N");
  Real Ae(quantity = "Force", unit = "N") = 0 ;
  Real T(quantity = "Force", unit = "N") = 0;
  parameter Real M(quantity = "Force", unit = "N") = 0;
  parameter Real aerodynamicFriction = 0;
  parameter Real faceArea(quantity = "Area", unit = "m2") = 0;
  parameter Real airDensity(quantity = "Density", unit = "kg/m3") = 0;
  parameter Real vehicleLength(quantity = "Length", unit = "m") = 4;
  parameter Real frictionCoeficient = 0;
  parameter Real vehicleMass(quantity = "mass", unit = "kg") = 30;
  Real deltaPosition(quantity = "Distance", unit = "m");
  Real deltaSpeed(quantity = "Distance", unit = "m", start = 0);
  SafetyDistance safetyDistance(vehicleLength = vehicleLength,
    safetyTime = safetyTime,
    previousVehicle = previousVehiclePosition,
    useVehicleLength = false);

//---------------inputs
  input Real safetyTime(quantity = "Time", unit = "s");
  input Real previousVehiclePosition(quantity = "Distance", unit = "m");
  input Real gravity(quantity = "Acceleration", unit = "m/s2") = 9.81;
  input Real vehiclePosition(quantity = "Distance", unit = "m");
  input Real coefficient;

//---------------outputs
  output Real c;
  output Real k;
  output Real A2;
  output Real B2;
```

```

//----------------derivations
  Modelica.Blocks.Continuous.Derivative vehicleSpeed;
  Modelica.Blocks.Continuous.Derivative vehicleAcceleration;
  Modelica.Blocks.Continuous.Derivative previousVehicleSpeed;

equation
  //----------------space variables
  deltaPosition = previousVehiclePosition - vehiclePosition - safetyDistance.distance;
  connect(vehicleSpeed.u, vehiclePosition);
  connect(previousVehicleSpeed.u, previousVehiclePosition);
  connect(vehicleAcceleration.u, vehicleSpeed.y);
  deltaSpeed = previousVehicleSpeed.y - vehicleSpeed.y;

  //----------------forces
  G = -c * deltaPosition;
  O = -k * deltaSpeed;
  Ae = aerodynamicFriction * faceArea * airDensity / 2 * vehicleSpeed ^ 2;
  T = vehicleMass * gravity * frictionCoeficient;
  vehicleMass * vehicleAcceleration.y  = -G - O - Ae - T + M;

  //----------------for simulation
  c = coefficient*k;
  A2 = (k/(4*vehicleMass))^2;
  B2 = c/vehicleMass;

end BoundVehicle;
```

Listing A.2: Modelica Class SafetyDistance

```
class SafetyDistance
//----------------inputs
  input Real vehicleLength(quantity = ”Length”, unit = ”m”) = 4;
  input Real safetyTime(quantity = ”Time”, unit = ”s”);
  input Boolean useVehicleLength = false;
  input Real previousVehicle;

//----------------outputs
  output Real distance(quantity = ”Distance”, unit = ”m”);

equation
  distance = if useVehicleLength then vehicleLength
  else if der(previousVehicle) * safetyTime > vehicleLength
  then der(previousVehicle) * safetyTime else vehicleLength;

end SafetyDistance;
```

Listing A.3: Modelica Class SafetyDistance

```
model TestBoundVehicle
```

```

//——————————inputs
  parameter Real coeficient = 1;
  parameter Real vehicleMass = 2000;
  parameter String vehicle1file = ”data/vehicle1.txt”;
  parameter String vehicle2file = ”data/vehicle2.txt”;
  BoundVehicle vehicle(vehicleLength = 5);
  Modelica.Blocks.Sources.CombiTimeTable vehicle1pos(
    fileName = vehicle1file,
    tableName = ”tab1”,
    tableOnFile = true,
    smoothness = Modelica.Blocks.Types. Smoothness.MonotoneContinuousDerivative2);
  Modelica.Blocks.Sources.CombiTimeTable vehicle2pos(
    fileName = vehicle2file,
    tableName = ”tab1”,
    tableOnFile = true,
    smoothness = Modelica.Blocks.Types. Smoothness.MonotoneContinuousDerivative2);

//——————————outputs
  output Real A2;
  output Real B2;
  output Real vehicle1c;
  output Real vehicle1k;

equation
  connect(vehicle1pos.y[1], vehicle.previousVehiclePosition);
  connect(vehicle2pos.y[1], vehicle.vehiclePosition);
  connect(A2, vehicle.A2);
  connect(B2, vehicle.B2);
  connect(coeficient, vehicle.coeficient);
  connect(vehicle1c, vehicle.c);
  connect(vehicle1k, vehicle.k);
  connect(vehicleMass, vehicle.vehicleMass);

end TestBoundedVehicle;
```

Appendix B

Python Code Sample

Listing B.1: Python adjustments of the input data

```python
def form_access(csv_file='..\\..\\..\\04-data\\TestQuery1_2.csv',
                output_file_path='..\\..\\..\\04-data\\test_output_3',
                acc_range=0.3,
                drop_acc = True):
    df_in = pd.read_csv(filepath_or_buffer=csv_file, header=0, names=['timestamp',
                                                                      'car1id',
                                                                      'lat1',
                                                                      'lon1',
                                                                      'acc1',
                                                                      'car2id',
                                                                      'lat2',
                                                                      'lon2',
                                                                      'acc2'])
    if drop_acc:
        max_diff = find_max_diff(df_in)
        df_in = df_in[acc_range < abs(df_in.acc1)]
        df_in.reset_index(inplace=True)
        df_in = shift_values(df_in, max_diff)

    df_out_1 = pd.DataFrame(columns=['timestamp', 'cumulative_distance'], dtype=float)
    df_out_2 = pd.DataFrame(columns=['timestamp', 'cumulative_distance'], dtype=float)

    df_in['distance1'] = two_point_dist(df_in.lat1,
                                        df_in.lat1.shift(1),
                                        df_in.lon1,
                                        df_in.lon1.shift(1))
    df_in['distance2'] = two_point_dist(df_in.lat1,
                                        df_in.lat2,
                                        df_in.lon1,
                                        df_in.lon2)

    df_out_1['cumulative_distance'] = df_in.distance1.cumsum()
    df_out_2['cumulative_distance'] = df_out_1.cumulative_distance - df_in.distance2

    df_out_1.timestamp = df_in.timestamp - df_in.timestamp[0]
    df_out_2.timestamp = df_in.timestamp - df_in.timestamp[0]

    df_out_1.interpolate(method='linear', inplace=True)
    df_out_1.dropna(inplace=True)
    df_out_2.interpolate(method='linear', inplace=True)
    df_out_2.dropna(inplace=True)

    dfToModelicaTxt(output_file_path + '_0', df_out_1)
    dfToModelicaTxt(output_file_path + '_1', df_out_2)


def find_max_diff(df_in):
    df_in['difference'] = df_in.timestamp - df_in.timestamp.shift()
    return df_in.difference.max()


def shift_values(df_in, max_diff):
    df_temp = [pd.DataFrame(columns=df_in.columns.values)]
    i = 0
    start_append = True
    for index, row in df_in.iterrows():
        if 0 < index < df_in.__len__() - 1:
            if (row.timestamp - df_in.loc[index-1].timestamp) > max_diff:
                start_append = True
                df_temp.append(pd.DataFrame(columns=df_in.columns.values))
                i += 1
            if start_append:
                df_temp[i].loc[index] = row
            if (df_in.loc[index+1].timestamp - row.timestamp) > max_diff:
                start_append = False

    df_out = df_temp[0]
    for index, component in enumerate(df_temp):
        if index > 0:
            diff = component.timestamp.min() - df_temp[index-1].timestamp.max()
            component.timestamp = component.timestamp - diff + 0.2
            df_out = df_out.append(component, ignore_index=True)
    return df_out


def two_point_dist(lat1=0, lat2=0, lon1=0, lon2=0):
    lat1 = lat1 / 10000000
    lat2 = lat2 / 10000000
    lon1 = lon1 / 10000000
    lon2 = lon2 / 10000000

    ret = np.arccos(
        np.cos(np.radians(90 - lat1)) * np.cos(np.radians(90 - lat2)) + np.sin(np.radians(90 - lat1)) * np.sin(
            np.radians(90 - lat2)) * np.cos(np.radians(lon1 - lon2))) * 6371000
    return ret


def sort_by_corr():
    data_input = 'E:\\Dropbox\\Hochschule\\Masterarbeit\\04-data\\02-heuristics\\02-without_stationary_points'
    for filename in os.listdir(data_input):
        if filename.split('.')[len(filename.split('.'))-1] == 'csv':
            res_file_name = os.path.join(data_input, filename)
            print res_file_name
            with open(res_file_name, 'rb') as csv_file:
                df_output = pandas.read_csv(csv_file)
                df_output.sort_values('corr', inplace=True, ascending=False)
                df_output.to_csv(data_input + '\\TOP\\' + filename)
```

Listing B.2: Python Change simulation Initiation

```
def changeSimulationInit(xmlName, variables, experimentAtributes):
    full_file = os.path.abspath(xmlName)
    xmlTree = ElementTree.parse(full_file)
    root = xmlTree.getroot()

    # change attributes of experiment
    settingAttribute = root.find('DefaultExperiment')
    for experimentAttribute in experimentAtributes:
        settingAttribute.set(
            experimentAttribute,
            str(experimentAtributes.get(experimentAttribute)))

    # change variables of experiment
    for scalarVariable in root.iter('ScalarVariable'):
        for real in scalarVariable.iter('Real'):
            try:
                newVar = variables.get(scalarVariable.get('name'))
                if newVar is not None:
                    real.set('start', str(newVar))
            except TypeError:
                pass

    xmlTree.write(full_file)
```

Bibliography

[1] P. Fritzson, *Principles of object oriented modeling and simulation with Modelica 3.3: a cyber-physical approach*, second. IEEE Press, 2015.

[2] R. Chase. [Online]. Available: `www.robinchase.org`.

[3] T. Tvrzský, "Využití kooperativních systémů pro ovlivňování dopravního toku", Technologická agentura České republiky, Tech. Rep., 2016.

[4] Tchořovice. (2017). Pronájem letiště, [Online]. Available: `www.tchorovice.cz/cs/pronajem-letiste/R46-A0/`.

[5] B. Wiley, *Gps geodetic reference system wgs 84*. [Online]. Available: `www.gps.gov/multimedia/presentations/2009/09/ICG/wiley.pdf`.

[6] B. Trammell, B. Tellenbach, D. Schatzmann, and M. Burkhart, "Peeling away timing error in netflow data", in *Passive and Active Measurement*, Springer, 2011, pp. 194–203.

[7] K. Menglerová, "Příspěvek k deteriministickému modelu dynamiky dopravního proudu", PhD thesis, CTU Prague - FD, 2015.

[8] L. Elefteriadou, *An Introduction to Traffic Flow Theory.* Springer, 2014.

[9] M. Mark Brackstone; Mike, "Car-following: A historical review", *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 4(2), pp. 181–196, 1999.

[10] A. Vasileia Papathanasopoulou; Constantinos, "Towards data-driven car-following models", *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 496–509, 2015.

[11] R. E. Chandler, R. Herman, and E. W. Montroll, "Traffic dynamics: Studies in car following", *Operations research*, vol. 6, no. 2, pp. 165–184, 1958.

[12] T. Kometani Eiji;Sasaki, "Dynamic behaviour of traffic with a nonlinear spacing-speed relationship", *Journal of Operational Research Japn*, vol. 2(1), 1958.

[13] W. Helly, *Simulation of Bottlenecks in Single Lane Traffic Flow.* Elsevier Publishing Co., pp. 207–238.

[14] R. Michaels, "Perceptual factors in car following", in *Proceedings of the 2nd International Symposium on the Theory of Road Traffic Flow (London, England), OECD*, 1963.

[15] S. Kikuchi and P. Chakroborty, "Car-following model based on fuzzy inference system", *Transportation Research Record*, pp. 82–82, 1992.

[16] R. Liu and X. Li, "Stability analysis of a multi-phase car-following model", *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 11, pp. 2660–2671, 2013.

[17] E. Chung and A.-G. Dumont, *Transport simulation: beyond traditional approaches.* EPFL press, 2009.

[18] New York Deparment of Motor Vehicles. (2017). Defensive driving, [Online]. Available: `www.dmv.ny.gov/about-dmv/chapter-8-defensive-driving`.

[19] Road Safety Authority of Ireland. (2015). Rules of the road, [Online]. Available: `www.rotr.ie/Rules_of_the_road.pdf`.

[20] T. D. Gillespie, *Fundamentals of Vehicle Dynamics. Warrendale.* Society of Automotive Engineers, 1992.

[21] M. Otter, *Modelica overview*, 2013. [Online]. Available: `www.modelica.org/education/educational-material/lecture-material/english/ModelicaOverview.pdf`.

[22] H. S. Tseng Wei-Kuo; Lee, "The Vector Function for Distance Travelled in Great Circle Navigation", *The Journal of Navigation*, vol. 60(1), 2007.