CZECH TECHNICAL UNIVERSITY IN PRAGUE
Faculty of Electrical Engineering
Department of Control Engineering

V2V communication-enabled collision avoidance for railroad vehicles

Master's thesis

Bc. Loi Do

Supervisor
doc. Ing. Zdeněk Hurák, Ph.D.

2019

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Do Loi** Personal ID number: **434929**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

Branch of study: **Systems and Control**

II. Master’s thesis details

Master’s thesis title in English:

V2V communication-enabled collision avoidance for railroad vehicles

Master’s thesis title in Czech:

Systém pro předejití srážce kolejových vozidel založený na komunikaci V2V

Guidelines:

1) Design an algorithm that could possibly run onboard a tram (or a train) and give a warning to the driver that a collision with the vehicle ahead is imminent unless the driver starts breaking immediately.
2) The algorithm contains as one of its components an estimator of its own position and velocity fusing the measurements from GNSS/GPS, accelerometers, tachometers and a digital map.
3) Estimated position and velocity of the tram (train) ahead are communicated wirelessly (vehicle-to-vehicle, V2V).
4) The proposed algorithm should be tested extensively using simulations, hence reasonably accurate models of tram (train) longitudinal dynamics (including traction and breaking) must be developed. Real experiments are anticipated in collaboration with project partners.

Bibliography / sources:

[1] N. An, J. Mittag, and H. Hartenstein, “Designing fail-safe and traffic efficient 802.11p-based rear-end collision avoidance,” in 2014 IEEE Vehicular Networking Conference (VNC), 2014, pp. 9–16.
[2] S. Iwnicki, Handbook of Railway Vehicle Dynamics. CRC Press, 2006.
[3] K. Knothe and S. Stichel, Rail Vehicle Dynamics. Springer International Publishing, 2017.
[4] A. Steimel, Electric Traction: Motive Power and Energy Supply, 2 edition. München: Deutscher Industrieverlag GmbH, 2014.
[5] S. Biswas, R. Tatchikou, and F. Dion, “Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety,” IEEE Communications Magazine, vol. 44, no. 1, pp. 74–82, Jan. 2006.

Name and workplace of master’s thesis supervisor:

doc. Ing. Zdeněk Hurák, Ph.D., Department of Control Engineering, FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **13.02.2019** Deadline for master's thesis submission: **24.05.2019**

Assignment valid until: **20.09.2020**

doc. Ing. Zdeněk Hurák, Ph.D.
Supervisor’s signature

prof. Ing. Michael Šebek, DrSc.
Head of department’s signature

prof. Ing. Pavel Ripka, CSc.
Dean’s signature

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Bc. Loi Do
May 2019

Acknowledgments

First and foremost, I would like to thank my supervisor, *Zdeněk Hurák*, for inviting me into his research group and giving me an opportunity to work on this project. This thesis would not have been possible without his supervision and guidance. My thanks also go to our project partner *Ivo Herman* from *Herman Systems* for organizing experiments and providing real data on which big part of this thesis is built upon.

I can not forget to mention and to thank my friends and colleagues from the lab. The occasional procrastination with them may have prolonged writing of this thesis but made my stay in the lab throughout the past year more enjoyable. I especially thank *Adam*$_1$ *Kollarčík* and *Martin Gurtner* for answering my typography-related questions. I must also thank *Vít Obrusník*, a hardcore hacker, for his contribution to the project. I also take this opportunity to thank all members of *AA4CC* for listening to my awkward control-engineering jokes.

I also wish to thank my family. With their support, I could fully concentrate on writing this thesis. Lastly, my immense gratitude belongs to my girlfriend *Anna*, with whom I could share both good and hard times of my studies.

Abstract

This thesis deals with the design of a collision avoidance system for trams utilizing V2V (vehicle-to-vehicle) communication between trams. V2V communication falls under the category of dedicated short-range communication technology, allowing to establish wireless communication between two vehicles. The thesis presents an algorithm which could run onboard a tram and in case of an imminent collision with tram ahead warn a driver that immediate action is required to prevent the collision. One part of the algorithm is an estimator of tram longitudinal motion based on Kalman filter which uses for estimation measurements from inertial sensors and GNSS/GPS. The functionality of this algorithm is then extensively tested using simulations with a different set of parameters of both the algorithm and simulations. For this purpose, the thesis also presents a mathematical model of longitudinal dynamics of a tram, including a model of a driver. The model generates data with acceptable accuracy close to real measurements.

Keywords: anti-collision system, digital map, kalman filter, V2X-communication.

Abstrakt

Tato práce se zabývá návrhem systému pro předejití srážce tramvají s využitím V2V (vehicle-to-vehicle) komunikace mezi tramvajemi. V2V komunikace spadá pod kategorii vyhrazené komunikace krátkého dosahu umožňující bezdrátovou komunikaci mezi vozidly. V práci je popsán návrh algoritmu, který by mohl běžet na palubě tramvaje a v případě hrozící srážky s tramvají jedoucí vpřed upozornit řidiče na nutnost neprodlené akce pro zabránění srážky. Jednou ze součástí algoritmu je pozorovatel stavu systému založený na Kalmanovu filtru využívající k odhadu stavu systému měření z inerciálních senzorů a GNSS/GPS. Funkčnost navrženého algoritmu je následně rozsáhle otestována na simulacích s různým nastavením parametrů algoritmu i simulace. K tomuto účelu práce také předkládá odvození matematického modelu podélné dynamiky tramvaje společně s modelem řidiče. Pomocí tohoto modelu lze generovat data věrná reálným měřením ze senzorů s přijatelnou přesností.

Klíčová slova: antikolizní systém, digitální mapa, kalmanův filtr, V2X-komunikace.

Contents

Nomenclature

Acronyms

CAM	Cooperative awareness message.
CAN	Controller Area Network (bus).
DEM	Digital elevation model.
DOF	Degree of freedom.
DSRC	Dedicated short-range communication.
ECEF	Earth-centered, earth-fixed.
FVCWS	Forward vehicle collision warning system.
GIS	Geographic information systems.
GNSS	Global Navigation Satellite System.
GPS	Global Positioning System.
IMU	Inertial measurement unit.
KF	Kalman filter.
LKF	Linear Kalman filter.
LMD	Longitudinal motion dynamics.
MEMS	Micro-electro-mechanical systems.
OSM	Open-street maps.
PDF	Probability density function.
PDM	Power drive mechatronics.
RMSE	Root-mean-square error.
SV	Subject vehicle/tram.
TV	Target vehicle/tram.
UCU	Universal communication unit.
UKF	Unscented Kalman filter.
V2V	Vehicle-to-vehicle.
V2X	Vehicle-to-everything.

Greek letters

Δt	Sampling period of the system.
$\omega_x, \omega_y, \omega_z$	Measured angular rate around an axis x, y, z.
ψ, θ, ϕ	Heading, slope and bank angles.
$\sigma(x)$	Standard deviation of an estimation of variable x.

Uppercase latin letters

$A^\intercal$	Transposition of a matrix A.
D	Time delay.

F_k	System (state) matrix of the system in time k.
$H(s), H(z)$	Transfer function for continuous-time and discrete-time system.
H_k	Output matrix of the system in time k.
K_k	Kalman gain at time k.
P_k	Covariance of the estimation error at time k.
T	Total time duration of a simulation.

Lowercase latin letters

a_x, a_y, a_z	Measured acceleration along an axis x, y, z.
c_h, c_v	Horizontal and vertical curvature of a track.
$d_t(P, Q)$	Distance along the track between points P and Q.
$d_{\mathrm{br},k}$	Braking distance at time k.
$f(.)$	Non-linear state equation of a system.
$f_{\mathrm{PDF}}(x)$	Probability density function of a variable x.
$h(.)$	Non-linear output equation of a system.
$s, \dot{s}, \ddot{s}$	Longitudinal position, speed and acceleration.
$v_k \sim (\mu_k, R_k)$	Measurement noise with Gaussian distribution with mean value μ_k and covariance matrix R_k at time k.
$w_k \sim (\mu_k, Q_k)$	Process noise with Gaussian distribution with mean value μ_k and covariance matrix Q_k at time k.
$x_{c,k}$	Clearance at time k.
x_k	State vector at time k.
y_k	Measurement vector at time k.
$y_{s,k}$	Distance along the track measured by GPS at time k.
$y_{v,k}$	Longitudinal (linear) speed measured by GPS at time k.

Other Symbols

$\approx$	Approximation.
$:=$	Assignment.
$\triangleq$	Definition.

1 | Introduction

In this thesis, I deal with a problem of collision avoidance for railroad vehicles, especially trams in an urban area. I focus on the utilization of V2V (vehicle-to-vehicle) communication in combination with absolute vehicle position estimation. V2V communication falls under the category of dedicated short-range communication (DSRC) technologies, allowing to establish wireless communication between two vehicles.

1.1 Motivation

This thesis deals with one part of a collaboration project between the Czech technical university in Prague and private company *Herman systems*. The shared interest is in the utilization of V2X (vehicle-to-everything) communication for Intelligent public transport. In particular, in the city of Ostrava, *Herman systems* have installed V2X-universal communication units (UCU) into public transport vehicles. To this date, nearly all public transport vehicles have V2X communication unit[1]. These communication units are capable of communicating with other vehicles or the surrounding infrastructure. Also every communication unit contains inertial measurement unit (IMU), GNSS module and access to Controller Area Network bus (CAN) of the vehicle (if applicable). Availability of such technology in almost every public transport vehicle gives the primal motivation of the project: how to create more intelligent public transport with the use of V2X communication?

In this thesis, I deal with one of the emerged use cases: collision avoidance for trams. This might seems to be a minor problem but for cities with dense tram infrastructure (such as Prague, Brno or Ostrava in the Czech Republic), a considerable number of collisions involving trams occur. For instance, in Prague, there are around 1400 tram collisions each year of which around 200 collisions are caused by driver of a tram, see statistics[2] in Tab. 1.1. These collisions may cause injuries (or even deaths) of passengers, damage to the tram with high repair cost and delay of the public transport. I give an overview of several more severe tram collisions in Tab. 1.2 (gathered from press reports and the Train office of the Czech Republic). Note that most of these collisions were rear-end.

[1] [In Czech] zdopravy.cz/v-ostrave-spolu-zacinaji-mluvit-tramvaje-osazeno-uz-je-95-procent-vozidel-24706/

[2] [In Czech] Taken from: novinky.cz/ekonomika/476795-srazkam-tramvaji-by-mohl-zabranit-autopilot-dpp-vyviji-system-s-cvut.html

Year	Total number of collisions	Caused by a driver of a tram	Caused by a driver of other vehicle	Collision with pedestrians
2008	1441	219	1055	76
2009	1421	225	1021	91
2010	1432	242	1040	76
2011	1279	177	959	69
2012	1282	204	943	62
2013	1302	204	938	82
2014	1394	226	999	90
2015	1359	194	984	99
2016	1353	183	1026	75
2017	1572	231	1164	111

Table 1.1: Statistics of tram collisions in Prague between years 2008 and 2017.
Source: novinky.cz

Year	Location	Type of a collision	Injuries	Damage Cost in [CZK]	Cause: human factor
2008	Ostrava	Frontal	3 fatalities, 11 serious, 55 other	2.6 millions	Yes
2009	Prague	Rear-end	1 fatality	58 millions	Partly
2014	Ostrava	Rear-end	Non	0.3 millions	No
2017	Brno	Frontal	10 light	0.5 millions	Partly
2017	Brno	Read-end	1 fatality	0.2 millions	Yes
2017	Ostrava	Rear-end	14 light	3 millions	Yes
2018	Prague	Rear-end	25 light	44 millions	Yes
2018	Prague	Rear-end	7 light	N/A	N/A
2018	Prague	Rear-end	1 half-serious	N/A	Yes
2018	Prague	Rear-end	12 light	N/A	N/A

Table 1.2: Overview of available data from tram collisions. I gathered the data from press reports and the Train office of Czech Republic.

1.2 Problem definition

To more accurately describe the problem, I use some terms defined by ISO 15623 [13]. For the sake of completeness, I list used terms with their definition below. Quotation starts here:

- "Collision warning: information that the system gives to the driver indicating the need for urgent action to avoid the severity of a potential rear-end collision with another forward vehicle."
- (modified) "Clearance $x_{c,k}$: distance $x_{c,k}$ (at time k) from the target vehicle trailing

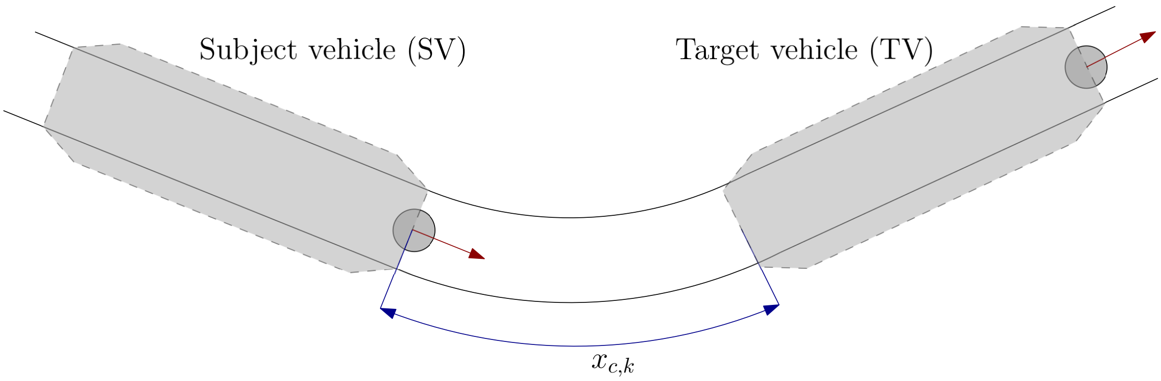


Figure 1.1: Terms defined by ISO 15623, namely Subject vehicle (SV), Target vehicle (TV) and clearance $x_{c,k}$.

surface to the subject vehicle leading surface."

- (modified) "Subject vehicle or Subject tram (SV): vehicle equipped with FVCWS as defined herein."
- "Forward vehicle: vehicle in front of and moving in the same direction and travelling on the same roadway as the subject vehicle."
- (modified) "Target vehicle or Target tram (TV): forward vehicle that is closest in the forward path of the subject vehicle; forward vehicle that the FVCWS operates on."
- "Forward vehicle collision warning system (FVCWS): system capable of warning the driver of a potential collision with another forward vehicle in the forward path of the subject vehicle."
- "Rear-end collision: forward vehicle collision in which the front of the subject vehicle strikes the rear of the forward vehicle."

Quotation ends here (ISO 15623 [13], p. 1-3). See Fig. 1.1 for illustration of the $x_{c,k}$, SV and TV. Since in this thesis, vehicles represent only trams, I will also use terms *subject tram* and *target tram* with the same abbreviations SV and TV. With the use of defined terms, this thesis aims to propose an implementation of FVCWS utilizing communication units placed on board of trams.

Collision warning/avoidance systems are heavily studied for cars, mainly for application such as platooning or autonomous vehicles [22, 23]. Although trams share many similar attributes with cars, there are specific attributes of trams which complicate the FVCWS design but also which simplify the design. One of the main attributes of trams (or railroad vehicles in general) is surely the restriction of their motion on rail tracks. Second important attribute of trams is their relatively high weight, for instance, 16 tonnes Tatra T3 tram) to 42 tonnes (Škoda 15T tram). Higher weight implies slower dynamics and thus for example, higher braking distance. In addition, this weight vary due to a number of passengers in the tram. For Tatra T3 and Škoda 15T trams, passenger capacity is 100 and 180 respectively. Assuming the average weight of one passenger to be 70 kg, this increases the total weight of Tatra T3 tram by ≈ 40 % and the total weight of Škoda 15T tram by ≈ 30 %. Such a change in weight can dramatically influence the dynamics of a vehicle.

1.3 Thesis outline

Let me now give a brief description of the proposed solution of the defined problem. I first focus on one of the vital functions of FVCWS: estimation of clearance $x_{c,k}$. Estimation of the clearance is typically done using vision-based approaches [15]. Most trams in Ostrava, however, do not have any sensor capable of measuring the clearance $x_{c,k}$ directly. I solve this in a distributed manner. Every tram estimates its state of motion using onboard sensors. Utilizing V2V communication, a tram then broadcasts its motion state to other trams in the area. When subject tram receives information of target tram's absolute position, clearance $x_{c,k}$ can then be computed for FVCWS. Another important functionality of FVCWS is an estimation of the braking/stopping distance. In general, the braking distance is hard to predict since it depends on many things such as adhesion conditions of the rail track or already discussed the weight of the tram. Lastly, a human-machine interface between the FVCWS and a driver of the tram is important, I do no cover this in the thesis.

The thesis is structured as follows. I first focus on position estimation of a single tram using sensory measurements. This part starts with Chapter 2 in which I give a description and an analysis of measurements from sensors placed on board of the tram. In Chapter 3, I examine available geographic information of the tram track, which plays an important role in position estimation. In Chapter 4, I focus on the development of dynamic models of tram motion. I then utilize results from the previous chapter in Chapter 5 in which I solve the state estimation using Kalman filter. I then focus on collision avoidance in Chapter 6. Finally, I conclude and discuss the achieved results in Chapter 7.

1.4 Collaboration

This project falls under the collaboration project *"Intelligent public transport with the use of V2X communication"* between Ing. Ivo Herman, CSc and Czech technical university in Prague and is financed by Technology Agency of the Czech Republic. Work on this project started in January 2018, firstly only by me. Part of the project was then done within course at CTU: FEE as a team project, lasting to June 2018. Unless otherwise stated, it is supposed that I did the described work in the thesis.

2 | Sensory measurements

This chapter serves as an introduction to modelling and identification of a railroad vehicle model, in particular, a tram. In this chapter, I will focus mainly on introducing sensory data from the inertial measurement unit (IMU) placed onboard a tram during its motion. IMU consist of triaxial digital, micro-electro-mechanical systems (MEMS) based accelerometer and gyroscope sensors. Data from IMU are supplemented by measurements from the GNSS module which measures the absolute geographic position and possibly a longitudinal velocity.

It is intuitive that the position obtained from GNSS is in the long term more reliable than position obtained by dead reckoning using inertial measurement. Inertial measurement, however, still plays an important role in state estimation of a tram. In particular, for short GPS signal outage, dead reckoning is the only way, how to determine the position of the tram. Therefore, it is desirable to gain as much information from inertial sensors as possible.

In this chapter, I will first give a list of sensors used for measurements. I will describe the motion of a tram on railtrack and how is this motion reflected in inertial sensors. Lastly, I will describe and show measured data.

2.1 Used sensors

Herman's UCU contains several sensors. I identified only the following sensors to be relevant: accelerometer, gyroscope and GNSS module. Also these sensors are only low-cost. For our analysis, it is thus appropriate to use sensors similar to which are in *Herman* UCU so proposed implementation of FVCWS is designed accordingly. We used a smartphone with an Android operating system which usually contains IMU and module for GNSS navigation. For completeness, the type of IMU in the smartphone used for measuring is BMI160[1]; GNSS module in the smartphone is unknown. To simultaneously acquire data from IMU and GNSS, my colleague Vít Obrusník created an Android application. A sampling frequency of accelerometer and gyroscope is set to around 500 Hz; sampling frequency of GNSS is set approximately to 0.1 Hz (fastest achievable frequency in smartphones when using only GNSS). From now on, I will refer to these sensors as 'low-cost'.

We, however, seek also to have sensors which give more precise measurements. This helps mainly to gain better knowledge about the motion of the tram and also helps in developing a model of the system in the following chapters. We select IMU from Analog devices:

[1] Datasheet available at www.mouser.com/ds/2/783/BST-BMI160-DS000-07-786474.pdf

ADIS16465-1[2] and GNSS module from u-blox: C94-M8P application board[3] with integrated GNSS module and active antenna. A sampling frequency of accelerometer and gyroscope is 2000 Hz; sampling frequency of GNSS is set to 1 Hz. From now on, I will refer to these sensors as 'precise'.

2.2 Inertial measurement of tram motion

In this section, I examine the motion of a tram and how this motion is reflected in measurements in triaxial accelerometers and gyroscopes. For all measurements, axes of IMU are aligned with the tram right-handed set of axes. That is, axes x, y and z are aligned respectively with longitudinal (pointing in driving direction), lateral (pointing left) and vertical axis (pointing up) of a tram. Note that discussion of inertial measurement in this section does not include other influences on measurement such as bias, sensor noise or vibrations from surrounding of the sensor. These influences will be discussed later in Sec. 2.3. This section is based on work done in [10].

2.2.1 Constrains on motion

In general, considering vehicle as a rigid body, six values (three for a position and three for an orientation) fully describe its configuration and therefore are needed to describe its motion accurately. This is for instance a case of unmanned aerial vehicles or underwater vessels where no holonomic constraints are imposed on the vehicle/vessel motion.

I can, however, describe the motion of a tram (both position and orientation) using only one parameter, thus having only one DOF. Intuitively, this can be seen if a railtrack is represented as a curve (center line of two parallel rails) see Fig. 2.1a. Only a single value s, a distance from starting point of the curve, parametrize the position of any point on the curve.

For describing the orientation of a vehicle, I use Tait–Bryan angles[4]: heading ψ, slope θ (longitudinal inclination) and bank ϕ (lateral inclination) see Fig. 2.1b. I chose North, East and Down (NED) frame orientation convention. For a given position, heading and slope angles can be uniquely determined assuming that tram is travelling only tangent to the track. From perspective of both horizontal and vertical movement, tram needs to follow the rails unless cases of a derail. Bank of the tram is given by the difference in height of rail (compensation of the lateral acceleration during the motion in turns). Angles describing the orientation are therefore functions of distance, denoted as: $\psi(s)$, $\theta(s)$ and $\phi(s)$. If not ambiguous, I will omit the dependence of angles on distance s.

[2] Datasheet available at www.analog.com/media/en/technical-documentation/data-sheets/adis16465.pdf

[3] Specification available at www.u-blox.com/en/product/c94-m8p

[4] In flight dynamics, these angles are called yaw, pitch and roll.

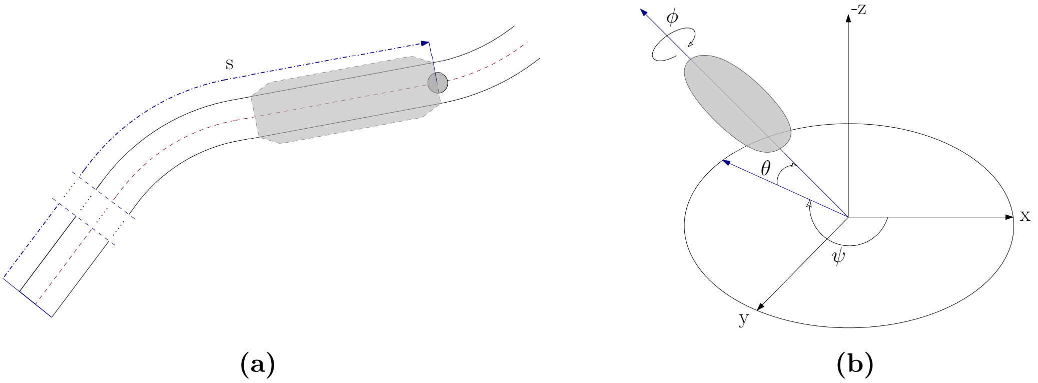


Figure 2.1: (a) Railroad track (two black parallel lines) represented by a center curve (red). Distance and thus a position on the track of a tram is given by a parameter s. **(b)** Orientation of a rigid body represented using Tait-Bryan angles with NED frame orientation convention. Angles heading, slope and bank are respectively denoted as ψ, θ and ϕ.

2.2.2 Accelerometer measurements

I denote three components of an inertial acceleration as a_x, a_y and a_z. For standing tram, these values are affected only by a gravity g:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} -g\sin\theta \\ g\sin\phi\cos\theta \\ g\cos\phi\cos\theta \end{bmatrix} . \tag{2.1}$$

For moving tram, longitudinal speed $\dot{s}$ and orientation can change throughout its motion. Change of speed $\ddot{s}$ is added directly to a_x whereas describing influence of change of orientation is more laborious. It is convenient to define horizontal and vertical curvature c_h and c_v:

$$c_h = \frac{d\psi(s)}{ds},\ c_v = \frac{d\theta(s)}{ds} . \tag{2.2}$$

If motion on curved track (both horizontally and vertically) is locally approximated as circular motion on arc, curvature is reciprocal of the radius r of this arc:

$$c_h = \frac{1}{r_h},\ c_v = \frac{1}{r_v} . \tag{2.3}$$

For change of heading c_h during motion, acceleration is therefore given as:

$$a_{\text{heading}} = \frac{\dot{s}^2}{r_h} = \frac{d\psi(s)}{ds}\dot{s}^2 = c_h\dot{s}^2 . \tag{2.4}$$

Taking into account also non-zero bank ϕ, acceleration (2.4) influence both a_y and a_z:

$$\begin{aligned} a_{y,\text{heading}} &= c_h \dot{s}^2 \cos\phi \ , \\ a_{z,\text{heading}} &= c_h \dot{s}^2 \sin\phi \ . \end{aligned} \tag{2.5}$$

Change of slope c_v during motion can be incorporated into accelerometer measurement in similar way as change of heading:

$$\begin{aligned} a_{y,\text{slope}} &= -c_v \dot{s}^2 \sin\phi \ , \\ a_{z,\text{slope}} &= -c_v \dot{s}^2 \cos\phi \ . \end{aligned} \tag{2.6}$$

Note that minus signs are added in (2.6) to be consistent with selected positive orientation of slope angle and selected orientation of IMU in a tram. Accelerometer thus measures:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \ddot{s} - g\sin\theta \\ g\sin\phi\cos\theta + c_h\dot{s}^2\cos\phi - c_v\dot{s}^2\sin\phi \\ g\cos\phi\cos\theta + c_h\dot{s}^2\sin\phi - c_v\dot{s}^2\cos\phi \end{bmatrix} \ . \tag{2.7}$$

For a tram dynamics, bank angle is usually not significant and thus $\cos\phi \approx 1$ and $\sin\phi \approx 0$. Formula (2.7) is simplified to:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \ddot{s} - g\sin\theta \\ c_h\dot{s}^2 \\ g\cos\theta - c_v\dot{s}^2 \end{bmatrix} \ . \tag{2.8}$$

2.2.3 Gyroscope measurements

I denote three components of an inertial angular rate vector measured by gyroscope as ω_{bank}, ω_{slope} and ω_{heading}. Gyroscope measures mainly change of the orientation of vehicle. Using rotation matrices C_ϕ and C_θ defining rotation around x axis by ϕ and rotation around y axis by angle θ respectively:

$$C_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -sin\phi & \cos\phi \end{bmatrix} , C_\theta = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & \cos\theta \end{bmatrix} \ . \tag{2.9}$$

gyroscope measurement is given by equation [21]:

$$\begin{bmatrix} \omega_{\text{bank}} \\ \omega_{\text{slope}} \\ \omega_{\text{heading}} \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + C_\phi \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + C_\phi C_\theta \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi}\sin\theta \\ \dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta \\ \dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi \end{bmatrix} , \tag{2.10}$$

For small bank angle and restricted tram dynamics, I can simplify the equation (2.10) to:

$$\begin{bmatrix} \omega_{\text{bank}} \\ \omega_{\text{slope}} \\ \omega_{\text{heading}} \end{bmatrix} \triangleq \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} . \tag{2.11}$$

2.3 Data analysis

Here, I explain and show how I use data from inertial measurements for state estimation of a tram. In previous Sec. 2.2, I have defined equations (2.8) and (2.11) which describe how a motion of the tram is sensed by inertial measurements. From these measurements, it should be possible to determine position and orientation on the tram at every time instance. These equations, however, do not include other influences which cause errors to motion measurement and make it impossible to estimate the motion using only IMU measurements. I therefore show samples from several datasets where these influences are analysed and then I select features which are relevant for state estimation of the tram.

For this analysis, I use data, which I acquired as a regular passenger in the tram, using both low-cost and precise IMUs. I placed both IMUs on a horizontal surface in the tram and fixed them onto the surface by a tape. I intentionally selected a tramway where trams can reach higher speed and contains both right and left turns.

2.3.1 Filtration

First, see Fig. 2.2a which displays a_x measured by precise accelerometer and its spectrogram in Fig. 2.2b. It can be seen from both figures that there are several sources of noise in the data. For standing tram (around $t \approx 0\,\text{s}$ or $t \approx 80\,\text{s}$), mainly white-noise affect the measurement (similar power on all frequencies); this is caused by nature of MEMS sensor. Another significant source creates oscillations with constant frequency, around 290 Hz, and is probably caused by power drive electronics in the tram.

When the tram started moving, measurement is heavily influenced by vibrations with high power which makes difficult to get some insight into the data. The motion of high weight vehicle (such as a tram), however, has slow dynamics. I can therefore use a low-pass filter to separate only useful signal from the measurement. I use discrete infinite impulse response (IIR) low-pass filter, third order, with passband frequency 2 Hz. The transfer function of the used filter is:

$$H_{\text{filt}}(z) = \frac{0.0736z^3 + 0.2208z^2 + 0.2208z + 0.0736}{z^3 - 0.9761z^2 + 0.8568z - 0.2919} . \tag{2.12}$$

Figs. 2.3a and 2.3b show filtered acceleration a_x and angle rate ω_y using precise IMU. Other measurement can be filtered in the same manner. After filtration, features of the motion contained in the data are more visible which helps in further analysis.

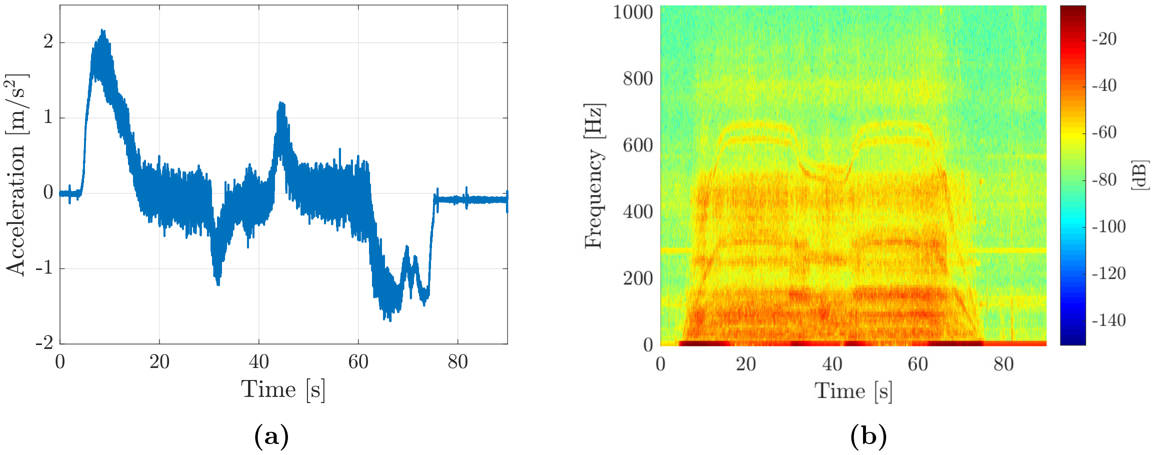


Figure 2.2: Acceleration measured by precise IMU on board of Škoda 15T tram. **(a)** Measurement from accelerometer a_x. **(b)** Spectrogram of a_x. Measurement contains white-noise and is heavily influenced by vibrations during the motion.

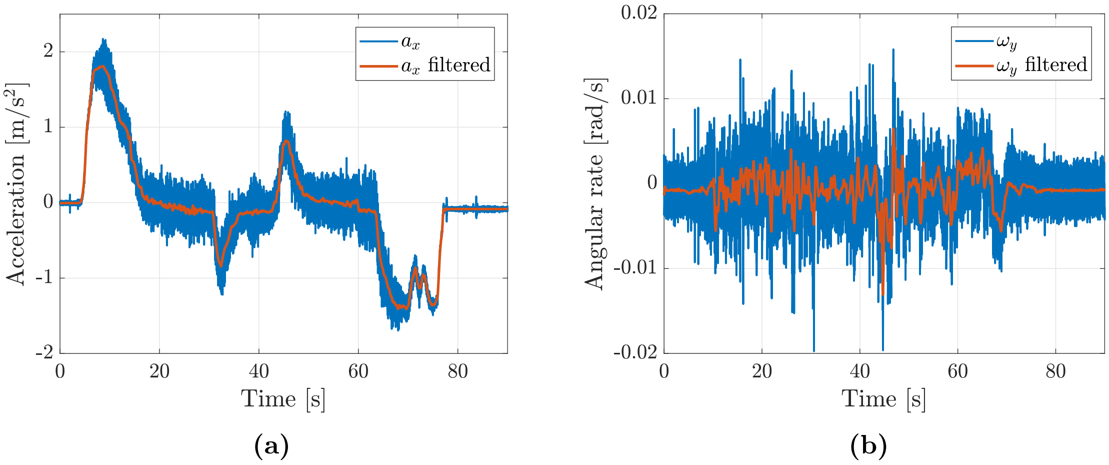


Figure 2.3: Measurement a_x and ω_y filtered by low-pass filter. After filtration, features of the motion contained in the data are more visible.

2.3.2 Features of the motion

Now, having measurements filtered from vibrations and partly from noise, it is easier to extract useful information from them. I first focus on longitudinal acceleration $\ddot{s}$ contained in a_x. When estimating the state of a tram, $\ddot{s}$ can be directly used for acceleration estimation. As can be seen from Fig. 2.3a, however, a_x is also affected by a gravity g causing changing bias (compare values for $t = 0\,\mathrm{s}$ or $t = 80\,\mathrm{s}$). Since this bias is dependent on the slope θ at the current position, it can be eliminated when information about the θ is available. Theoretically, it should be possible to obtain θ from ω_y by numerical integration. Measurement ω_y, however,

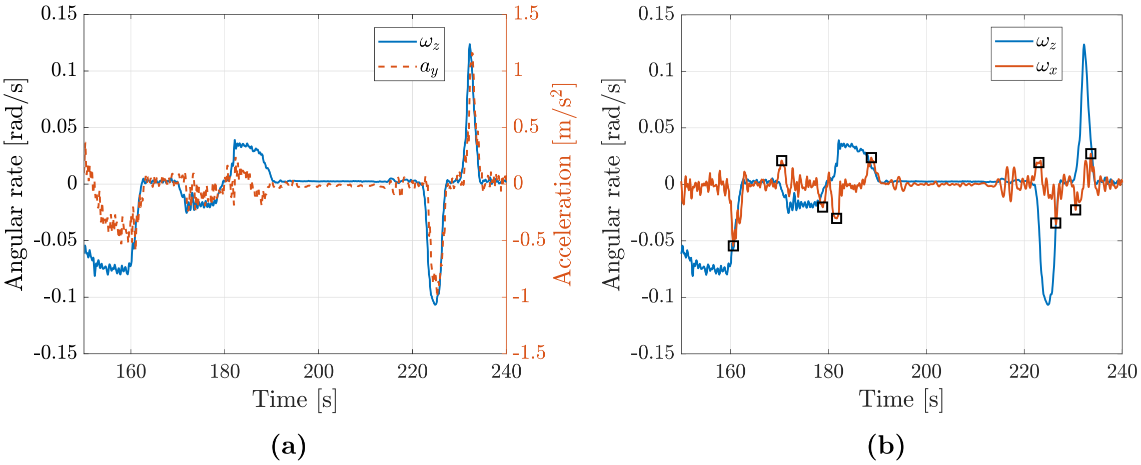


Figure 2.4: **(a)** Horizontal curvature c_h visible in ω_z and a_y. **(b)** Measurement from gyroscope: ω_x and ω_z. Change of bank due to compensation of lateral acceleration is visible (highlighted by black squares).

contains other low-frequency components which were not filtered out by low-pass filter. As can be seen in Fig. 2.3b, measurement has constant bias and is affected by swinging of the tram body due to longitudinal acceleration/deceleration (visible in Fig. 2.3a). For trams, change of the slope is usually small and therefore it is hard to distinguish it in ω_y from other influences. To de-bias the measurement a_x I will use geographic information of a tram track since orientation of a vehicle is fully given by its position on a track, This idea is explained in detail in the next chapter.

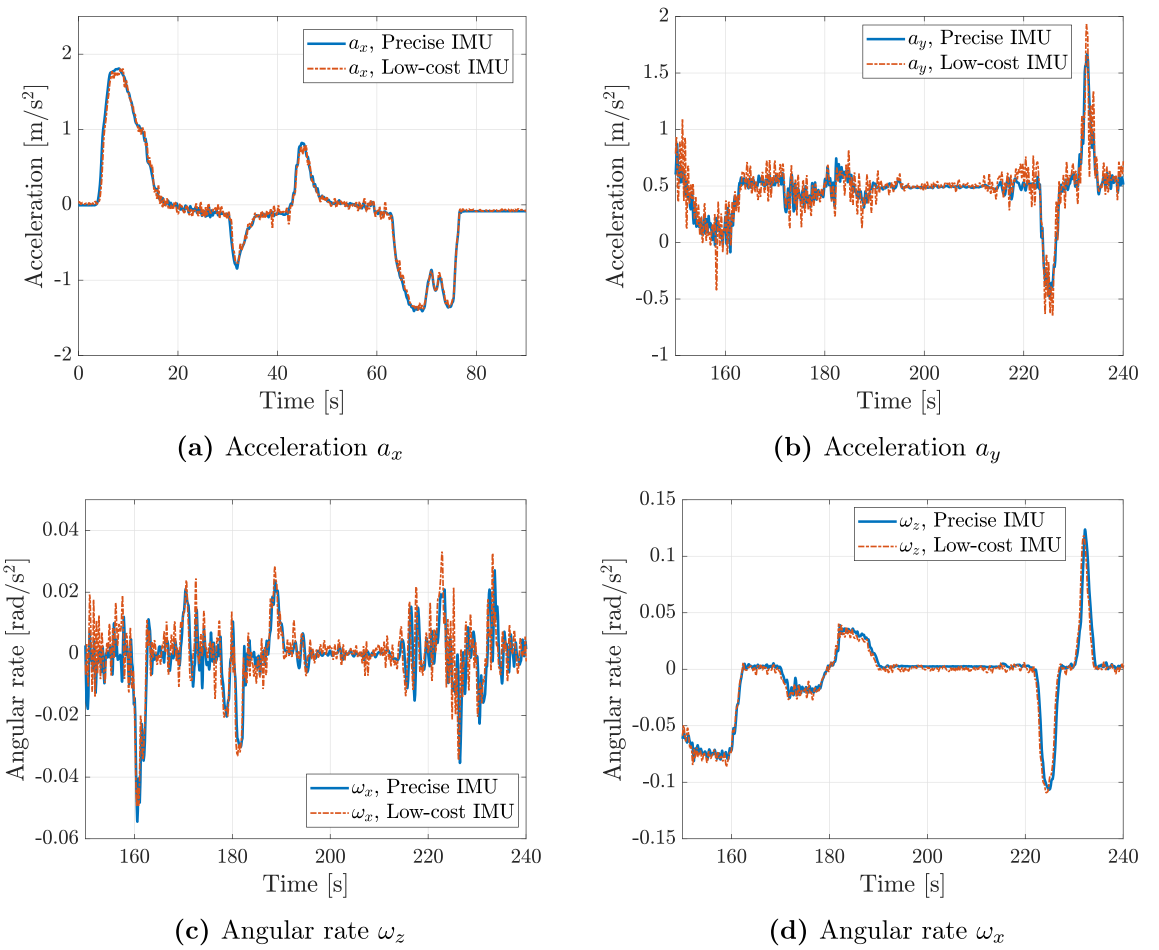
Another feature of the motion which is contained in the measurement is horizontal curvature c_h. Curvature c_h is reflected in ω_z and also in a_y, see Fig. 2.4a. Again, using geographic information of a track, I can compare measured curvature c_h with the curvature of the track at a particular position and thus refine position estimation.

Lastly, see Fig. 2.4b which contains measurements ω_x and ω_z. From this figure, it can be seen, that there is a change in bank angle accompanying a change of heading to lateral acceleration (highlighted by black squares). Information of current bank, regarding its small amplitude, has no use in estimation.

2.3.3 Comparison of data from low-cost and precise IMUs

In the previous subsection, I have shown only datasets obtained by a precise IMU. For a low-cost IMU, it is intuitive, that several features could be lost in the data due to smaller sampling frequency or overall lower sensitivity.

First, see the comparison of acceleration a_x in Fig. 2.5a. Longitudinal acceleration $\ddot{s}$ of the tram is visible in the data even from low-cost IMU. Next, see the comparsion of acceleration



(a) Acceleration a_x

(b) Acceleration a_y

(c) Angular rate ω_z

(d) Angular rate ω_x

Figure 2.5: Comparison of measurements from Precise and Low-cost IMUs.

a_y in Fig. 2.5b. Lateral acceleration caused by horizontal curvature c_h is visible only for higher speed of the tram (between 220 s and 240 s). For lower speed of the tram, lateral acceleration is less visible due to unfiltered noise. Lastly, see Fig. 2.5c and 2.5d which show measurements of angular rates ω_x and ω_z. In ω_x from low-cost IMU, the bank change is even less legible than from precise IMU. Measurement ω_z from Low-cost IMU is similar to the measurement from precise IMU and thus can be used for estimation.

3 | Digital map

In this chapter, I will examine available geographic information related to a railroad track on which subject and target vehicles are travelling. As I have shown in Chapter 2, using the geographic information, which I refer to as the *digital map*, it is sufficient to describe the motion of a tram as a point mass moving on railtrack with one degree of freedom (DOF). The exact geographic position of the vehicle, including its orientation can then be represented using the digital map. Such representation gives a simpler description of the motion of the vehicle and thus enables to create a more efficient dynamic model of a tram. Retaining information about the vehicle orientation in the digital map is helpful in state estimation.

Further in this chapter, I will describe how the digital map can be constructed for a particular railroad track using data from geographic information systems (GISs). Note that in this thesis, I use WGS84 [12] map coordinate system to express the geographic position.

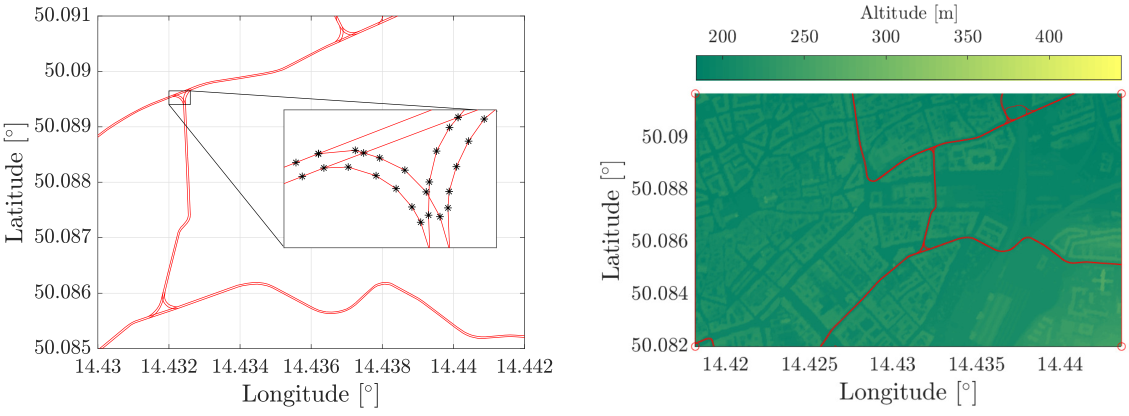
3.1 Representation of the railroad track

One railroad track consists of two parallel rails which are constructed as a smooth curves. In GISs, however, railroad track is usually stored/represented only as one curve (same representation used in Sec. 2.2.1). This curve is approximated as a set of points connected by line segments. As an example, see Fig. 3.1a. Note that these points, which I will refer to as *track nodes*, are not selected equidistantly but to reasonably approximate real shape of the track. Information about the bank of the track is not contained in this representation, but as shown in Sec 2.3.2, this information is not important.

To get the digital map of a particular area (in this case, I chose an area in Prague), I merged information from OpenStreet map[1] (OSM) with digital elevation model (DEM) from Geoportal of Prague[2]. OSM contains only latitude and longitude of these points whereas DEM contains altitude. See the output of the merge in Fig. 3.1b. There are many other GISs with needed data but I chose those two for their accuracy.

[1] www.openstreetmaps.org

[2] www.geoportalpraha.cz



(a) **(b)**

Figure 3.1: **(a)** Railroad track (red curves) given by set of points (black dots) connected by a line. Note that each read curve represents one pair of rails. Area captures Prague, track near Jindřišská street. **(b)** Merge of data from OSM and DEM. Red curves represents railroad tracks whereas color at particular point gives the altitude.

3.1.1 Parsing OSM data

OSM data consists of hierarchically structured elements, see documentation on OSM wiki[3]. To get relevant data from OSM, I proceeded in the following way. First, I investigated the structure of data which can be exported from OSM. I found the XML format as the most convenient since there are several MATLAB® community libraries dedicated to parsing data in this format. Data for smaller areas like in Fig. 3.1a can be exported directly from OSM site[4]. Data for larger areas (e.g. whole city) can be exported for instance from Humanitarian OSM Team site[5]. For creating the digital map of tram track, important elements are: node, way and relation. Each element can also have a tag which further characterizes the element. Tag is defined by its name and value.

- Node: represents geographic position of a point (black dots in Fig. 3.1a). Nodes can also define tram stops by having tag "railway" with value "tram_stop".
- Way: represents connection of nodes and thus creates (a segment of) tram track. Ways representing only tram track has a tag "railway" with value "tram".
- Relation: represents a whole tram line. It consists of set of members. Each member can be a node (tram stops) or a way (segment of a whole tram line). It has a tag "name" with value defining the exact tram line, for instance: "Tram 18: Vozovna

[3] https://wiki.openstreetmap.org/wiki/Beginners_Guide_1.3

[4] https://www.openstreetmap.org/export

[5] https://export.hotosm.org

Pankrác ⇒ Nádraží Podbaba".

To get the digital map for a particular tram line, I find a corresponding *relation* (using a tag "name"). I then iterated through all *ways* in this *relation* and for each *node* in each *way*, I stored its geographic location. Information from OSM of a tram line can be gathered to a structure with following fields:

- Coordinates: list of size n, each i-th entry contains latitude and longitude of a i-th track node.
- Lines: list of size $(n-1)$, each i-th entry contains an analytic description of a line segment connecting i-th and $(i+1)$-th track nodes.

3.1.2 Parsing DEM data

DEMs are usually stored as raster image of an area. The geographic location of the area is given by coordinates of corner points of the image. The captured area is divided into a grid of square areas, each square corresponds to one pixel of the image. Therefore, each pixel contains information about the longitude, latitude and altitude. Key attribute of DEM is resolution of the grid—the size of an area covered by one pixel. In Fig. 3.1b, DEM has 1 m grid.

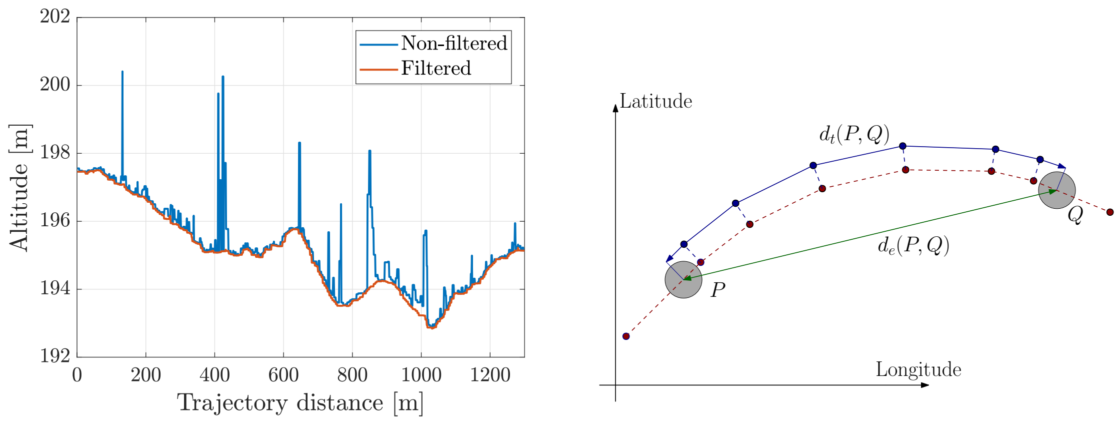
To verify data from DEM, I created a trajectory consisted of equidistantly sampled points on tram track and directly read an altitude from DEM. Resulted altitude profile is displayed in Fig. 3.2a (blue line). Altitude profile contains clearly unrealistic change of altitude. This inaccuracy can be caused by noise in the data but also by an incorrect alignment of OSM data with DEM data. To solve this problem, I tried several 2-D order-statistic filtering of DEM image. The best result was achieved by using first order (minimum) filter from 3×3 vicinity of a pixel. See the filtered altitude profile of the track in Fig. 3.2a (red line). Information of altitude is added to the structure described above as third coordinates of track nodes.

3.2 Use of the digital map

In this section, I describe, how I use the digital map for state estimation. For particular tram, it is not necessary to store the digital map for a whole area. Unlike the other road vehicles which can move freely on any road in an area, trams are usually constrained by their current schedule. Therefore, only the digital map of its scheduled path needs to be stored.

3.2.1 Position on the track

Main use of the digital map is to restrict motion of a tram to track and thus use only one parameter (distance s) instead of two (or three) geographic coordinates.



(a)

(b)

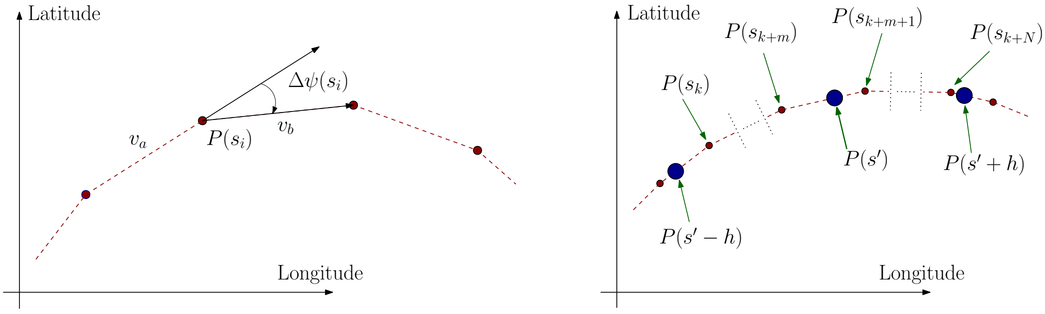
Figure 3.2: (a) Comparison of non-filtered and filtered altitude data obtained from DEM. **(b)** Definition of distance along the track $d_t(P,Q)$ and comparison with euclidean distance $d_e(P,Q)$. Red dashed line represents tram track and red dots track nodes.

It is convenient to define *distance along the track* $d_t(P,Q)$ (or *travelled distance*) of two points P and Q, see Fig. 3.2b. It is intuitive that distance along the track better reflects travelled distance of a tram then Euclidean distance $d_e(P,Q)$. If not ambiguous, I will refer to distance along the track only as distance. When one point on the track is selected as a reference (with $s=0$), distance s uniquely gives a position on the track (forward points from reference points has positive distance, backwards points has negative distance). Coordinates (latitude and longitude) are then given by iterating from current position through all track nodes in path to reach distance s. For illustration, see again Fig. 3.2b for $d_t(P,Q)=s$ where P is current point and point in distance s is Q. To get coordinates between two consecutive track points, analytical description of line connecting two points is used.

3.2.2 Horizontal curvature and slope angle

Another values which I use from the digital map are horizontal curvature and slope angle computed from track nodes coordinates. I use horizontal curvature directly for position estimation (comparison with gyroscope measurements ω_z) and slope angle to eliminate bias $g\sin\theta(s)$ caused by gravity in accelerometer measurement $a_x = \ddot{s} - g\sin\theta(s)$. Note that because WGS84 coordinates (of track nodes) are not Cartesian, I converted them into ECEF (earth-centered, earth-fixed) coordinates which are Cartesian. This is important for computation given below. For further explanation, I introduce following notation:

- $P(s)$: ECEF coordinates of points on track in distance s,
- s_i: distance of i-th track node,
- $P(s_i)$: ECEF coordinates of i-th track node.



(a) **(b)**

Figure 3.3: **(a)** Change of heading computed for track nodes P_i as angle between segments v_a and v_b. **(b)** Computation of curvature c_h using $\Delta\psi(s_i)$ of track nodes between points $P(s-h)$ and $P(s+h)$.

Horizontal curvature

Horizontal curvature c_h is given as derivation of $\psi(s)$, see Eq. (2.2). Because the track is given only by a set of track nodes, a derivation is replaced by numerical differentiation. Since heading $\psi(s)$ changes only on track nodes (heading is constant on the line between them), I first compute only changes of heading $\Delta\psi(s_i)$. Let denote the segment which ends by i-th track node as v_a and a segment which starts with i-th track node as v_b, see Fig. 3.3a. Change of heading $\Delta\psi(s_i)$ is equal to angle between vectors v_a and v_b. To compute this angle and retaining information of the sign of orientation (positive to for left turn and negative to the right turn), I use the following equation:

$$\Delta\psi(s) = \mathrm{sign}[(n \times v_a) \cdot v_b] \arccos\left(\frac{v_a \cdot v_b}{|v_a||v_b|}\right) . \tag{3.1}$$

where n is normalized vector pointing from the origin to point $P(s_i)$. To compute numerical differentiation, I use symmetric difference:

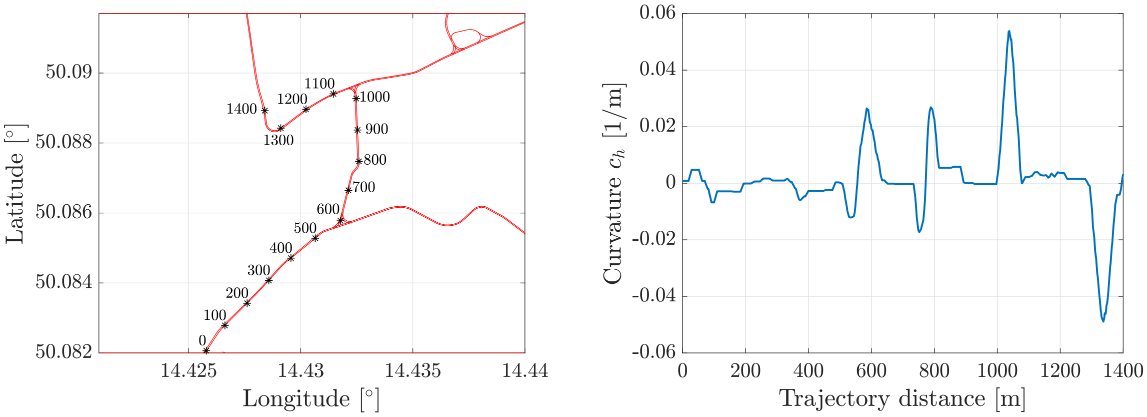
$$c_h = \frac{\Delta\psi(s)}{\Delta s} = \frac{\psi(s+h) - \psi(s-h)}{2h} . \tag{3.2}$$

I further change this equation so I can directly use $\Delta\psi(s_i)$ instead of absolute values of heading $\phi(s+h)$ and $\phi(s-h)$. Curvature c_h at point s' is then given by equation:

$$\frac{\Delta\psi(s')}{\Delta s'} = (2h)^{-1} \sum_{i=k}^{k+N} \Delta\psi(s_i) , \tag{3.3}$$

where $k, k+1, ..., k+N$ are indexes of starting track nodes between points $P(s-h)$ and $P(s+h)$, see Fig. 3.3b.

To test the procedure explained above, I computed the $c_h(k)$ for 1400 m long trajectory



(a) (b)

Figure 3.4: **(a)** Trajectory generated by sampling railroad track with 1 m spacing. Markers indicates 100 m segments of the trajectory for better pairing with computed curvature. **(b)** Curvature of the generated trajectory.

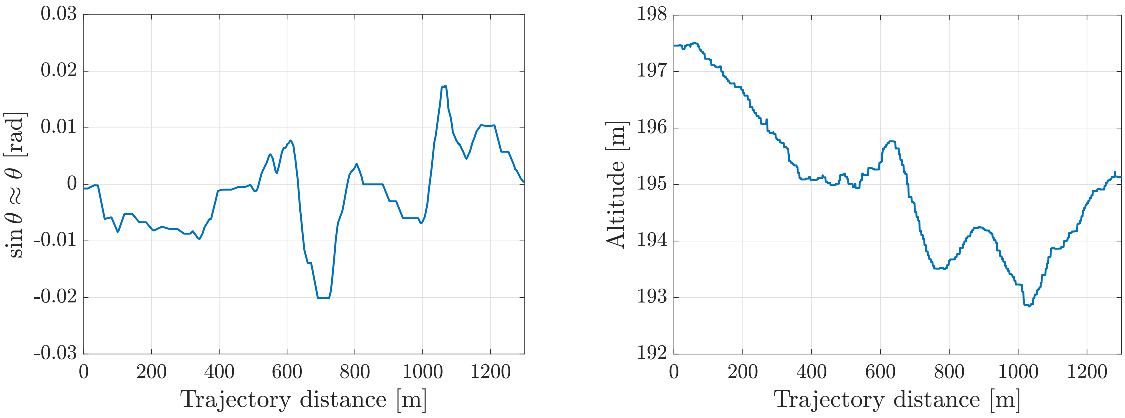
generated along the particular tram line. Points of trajectory are sampled every 1 m, thus $k \in \{1, 2, \ldots, 1399, 1400\}$. See the generated trajectory in Fig. 3.4a, markers on the figure indicates 100 m long segments. Markers on the figure indicates 100 m long segments. Resulted horizontal curvature of this trajectory is shown in Fig. 3.4b. In this case, I found the best result with $h = 15\,\text{m}$. In addition, I used a moving average filter with window size of 10 samples to lower influence of inaccuracy in the data.

Slope angle

Computation of slope angle $\theta(s)$ is similar to computation of heading $\psi(s)$ in case of horizontal curvature. I already explained how I obtained altitude profile of the track nodes in Sec. 3.1.2. Unlike horizontal curvature, I am interested in absolute value of $\theta(s)$ or more precisely $\sin\theta(s)$. To get $\sin\theta(s)$, I compute slope $\theta(s_i)$ for track nodes P_i. For P_i, slope $\theta(s_i)$ is given by equation:

$$\sin\theta(s_i) = \frac{A(s_{i+1}) - A(s_i)}{s_{i+1} - s_i}\ . \tag{3.4}$$

where $A(s_i)$ and $A(s_{i+1})$ are altitude at point $P(s_i)$ and $P(s_{i+1})$ respectively. To get value of $\sin\theta(s')$ for arbitrary point s' on track and robust to inaccuracy in the data, I proceed in similar manner as for computation of horizontal curvature. First, for $\sin\theta(s')$, I select points $P(s' - h)$ and $P(s' + h)$. Then I iterate through all track nodes between these points and compute (3.4) for all track nodes in the path to get $[\sin\theta(s_k), ..., \sin\theta(s_{k+N})]$. From all these values, I make a mean which gives a $\sin\theta(s')$. Lastly, to again lower the influence of inaccuracy in the data, I used moving average filter with window size of 20 samples. See the resulted $\sin\theta(s')$ in Fig. 3.5a for $s' \in \{1, 2, \ldots 1300\}$. I tested this procedure for the



(a)

(b)

Figure 3.5: (a) Slope θ obtained from altitude profile of the generated trajectory. **(b)** Altitude profile of the generated trajectory.

same trajectory which I generated for altitude profile in Fig. 3.2a. For easier comparison of altitude profile and slope profile, the same altitude profile is shown in Fig. 3.5b.

4 | Dynamic model of a tram

With this chapter, I am stepping into the modelling of dynamic system of a tram. In Chapter 3, I have shown that it is sufficient to describe only the longitudinal dynamics of the vehicle. Exact geographic position and track characteristics for position estimation are given by the digital map. In general, it is convenient to have as precise model of a plant as possible. On the other hand, it is intuitive that white box approach (complete description of electronics and mechanics of a tram) is unnecessary or even unrealisable.

With available measurements from sensors and insight into the system, I will create two grey-box models. First, I will develop a simple model which will reflect a motion of the tram, incorporating the available measurements and use the digital map. I will use this model in estimation algorithm. Then, I will develop second, high-fidelity model which will be used mainly to generate a testing data (similar to real data). This enables to evaluate the proposed algorithm with extensive (Monte Carlo) simulations. Similar evaluation of the algorithms using real data would be infeasible because of many reasons (missing information of ground-truth position of the tram, time consumption of real experiments, ...).

4.1 Simple model

For collision avoidance, main motion-dependent values are: position, velocity and acceleration of a tram, Since there is no information of input, I use constant acceleration model:

$$x_{k+1} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_k \tag{4.1}$$

with state vector consists of longitudinal (along the track) position, speed and acceleration: $x^\intercal = [s, \dot{s}, \ddot{s}]$ and Δt is sampling period of the system. White noise $w_k \sim (0, Q_k)$ is added to model the uncertainty in change of acceleration caused by the input. Covariance Q_w is [18]:

$$Q_k = \begin{bmatrix} \Delta t^5/20 & \Delta t^4/8 & \Delta t^3/6 \\ \Delta t^4/8 & \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^3/6 & \Delta t^2/2 & \Delta t \end{bmatrix} \tag{4.2}$$

4.1.1 Output model

Creating output model is more laborious. In the Chap. 2, I have shown that only following measurements from sensors are relevant:

- Position from GPS y_s,
- Velocity y_v,
- Accelerations a_x and a_y from accelerometer,
- Angular rate ω_z from gyroscope.

Since model (4.1) uses parameter s to represent position whereas GPS gives coordinates in latitude and longitude, I convert GPS coordinates to representation using s. GPS points which are on track can be directly parametrized by value of s. Other points (which are not on track) can be projected onto track as closest point on the line from the GPS point. Therefore, I handle GPS measurements as direct measurement of current distance s. I denote GPS position measurement as y_s.

Measurements y_v and a_x are directly given by states $\dot{s}$ and $\ddot{s}$ respectively. For remaining measurements a_y and ω_z, I use the digital map which gives a mapping $c_h(s)$. For a_y, I then use Eq. (2.8) and for ω_z I use Eq. (2.11). In addition, all measurements contains noise v_k. I model this noise as white, zero mean with covariance matrix R_k. To sum up, output $y_k = [y_s, y_v, a_x, a_y, \omega_z]^\intercal$ at time k is given two sets of equations, linear:

$$\begin{bmatrix} y_{s,k} \\ y_{v,k} \\ a_{x,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} v_k \,, \tag{4.3}$$

and non-linear:

$$\begin{aligned} a_{y,k} &= c_h(s_k)\dot{s}_k^2 + \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} v_k \,, \\ \omega_{z,k} &= c_h(s_k) + \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} v_k \,, \end{aligned} \tag{4.4}$$

where $v_k \sim (0, R_k)$. Furthermore, for GPS outage, only available measurement is a_x:

$$a_{x,k} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} v_k. \tag{4.5}$$

This can be described as *hybrid* system where with available GPS, output is given by Eq. (4.3) and (4.4) and with GPS outage by Eq. (4.4) and (4.5).

4.2 High fidelity model

In this section, I will create a model of the system which better reflects data measured on-board of a tram. An important note is that this model is not supposed to be an accurate model for all trams. Rather, the goal is to have a decent model of a particular tram with driver based on available data for generating data which would be similar to real data. These data will be used for evaluation of the estimation algorithm. Most of the data which will be used for identification are from experiments in Poruba depot. Measurements were acquired on-board of Tatra T3 tram. Details of these experiments are in the Appendix B.

With insight into the tram mechatronics, I divide the model into five subsystems:

- Driver: controller of a tram motion.
- Power mechatronics: this subsection includes all electronics and mechanics between the driver's input and torque created by a motor which is applied to the wheel.
- Longitudinal motion dynamics (LMD): represents the longitudinal motion of a tram.
- Digital map: incorporates track characteristics, already defined in the Chap. 3.
- Model of the sensors (GNSS module and IMU): adding noise to signals.

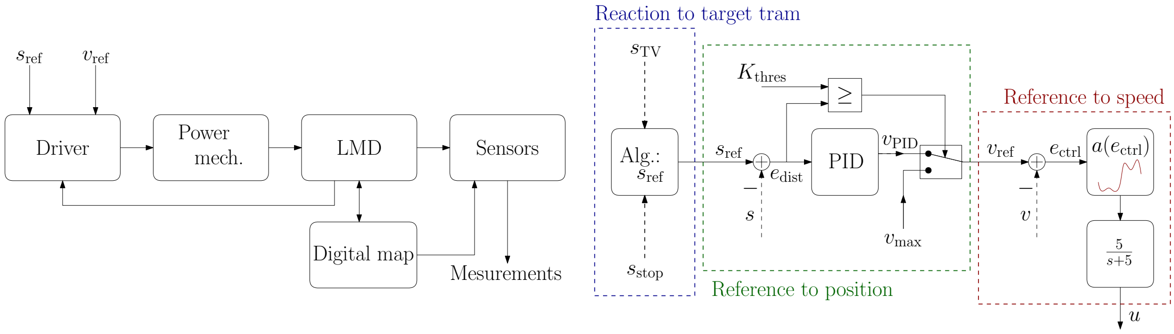
See their interconnection in Fig. 4.1a. I created a complete model in Simulink.

4.2.1 Driver model

Looking closely at the role of the driver, his task is to drive the tram from point A to point B (tram stops). Throughout the travel, the driver is adjusting acceleration and speed to react to the environment. This is, for instance, avoiding collisions with other vehicles on the track or satisfying soft and hard constraints such as maximal allowed speed, traffic rules or passenger comfort. Role of the human driver in the tram is thus the same as the role of a controller in closed-loop dynamic system. However, unlike a regular (deterministic) controllers such as PID controller or state-feedback [9], driver's decisions are mainly set by the experience of the driver and thus not always deterministic. For the purpose of developing the model, I seek to parametrize how the driver is deciding about the control input: the position of a lever or a pedal which controls value of torque generated by a motor and thus the motion of the vehicle. The block diagram of this subsystem is shown in Fig. 4.1b, below is a description of its development.

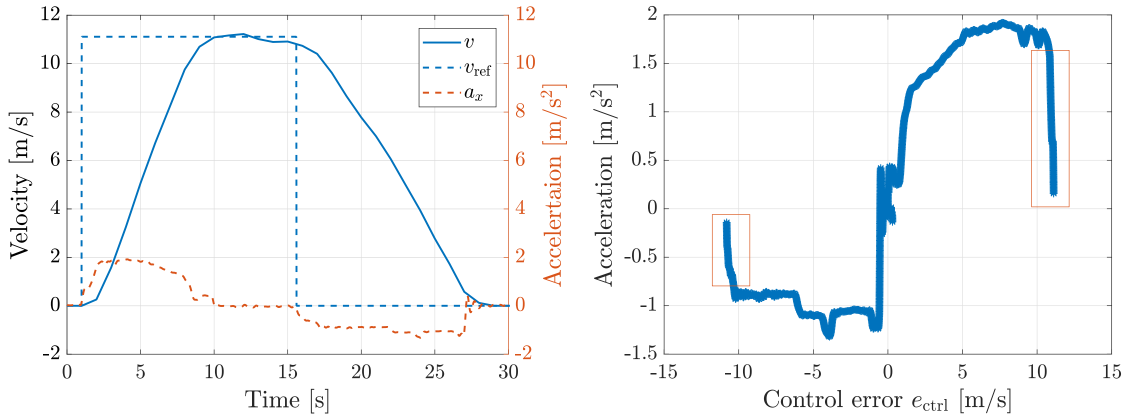
Reference on speed

The only available data for parametrizing driver's decisions are data from experiments in Poruba depot where we instructed driver to follow speed reference v_{ref} which vary from $5\,\mathrm{km\,h^{-1}}$ to $40\,\mathrm{km\,h^{-1}}$ with $5\,\mathrm{km\,h^{-1}}$ spacing. We also instructed driver both to speed up and to stop as if there were passengers in the tram (satisfying their comfort during acceleration



(a) (b)

Figure 4.1: **(a)** Interconnection of subsystems in high fidelity model. **(b)** Block diagram of the Driver subsystem.

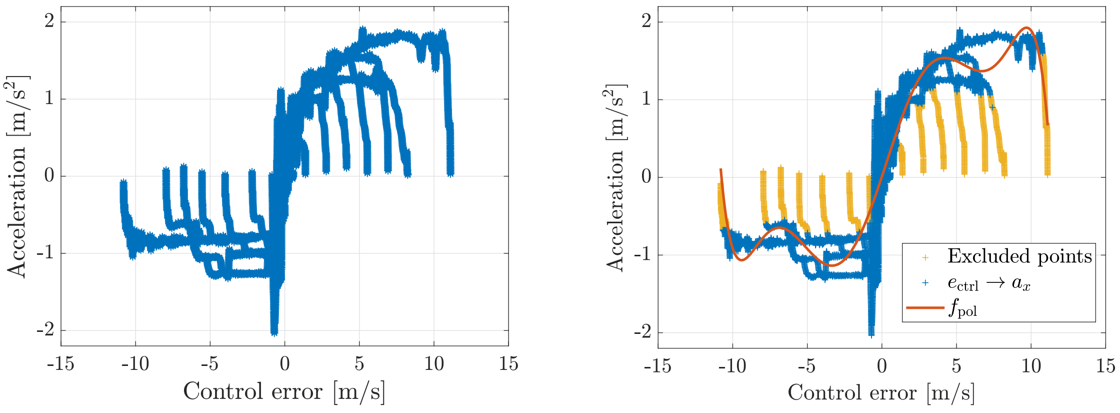


(a) (b)

Figure 4.2: **(a)** Experiment in which driver followed reference speed with pulse to $v_{\text{ref}} = 40\,\text{km}\,\text{h}^{-1}$. Displayed is speed v measured by GNSS module and filtered acceleration a_x. **(b)** Dependence of a_x on control error e_{ctrl}. Areas in red rectangles caused by step changes of v_{ref}.

and deceleration). In Fig. 4.2a, see output from one of such experiment. The best approach would be to compute control error $e_{\text{ctrl}} = v_{\text{ref}} - v$ and create mapping from values of e_{ctrl} to values of input u, for instance position of the control lever. The problem is that there is no direct information of the chosen input u. I address this problem in the Subsection 4.2.2.

For now, I create a mapping from values of e_{ctrl} to value of a_x, see Fig. 4.2b. This should represent which levels of acceleration is driver choosing depending on current value of e_{ctrl}. Points in red rectangles represent the situation when there is a step change of v_{ref}. For a brief moment, there is already high e_{ctrl} but driver did not change the control input yet or the change of the input did not propagate to sensor yet. Except points in red rectangles, the mapping $e_{\text{ctrl}} \to a_x$ agrees with intuition. With higher control error, driver tends to choose



(a) **(b)**

Figure 4.3: **(a)** Dependence of a_x on control error e_{ctrl} for eight experiments with v_{ref} varying from $5\,\mathrm{km\,h^{-1}}$ to $40\,\mathrm{km} \perp \mathrm{h}$. **(b)** Seventh order polynomial function f_{pol} (red) which represents driver's decisions. Function was found using polynomial regression where yellow points were excluded from regression.

n	7	6	5	4
p_n	-1.405×10^{-6}	1.540×10^{-6}	2.973×10^{-4}	-2.768×10^{-4}
n	3	2	1	0
p_n	-2.007×10^{-2}	1.705×10^{-2}	5.798×10^{-1}	1.1673×10^{-2}

Table 4.1: Coefficients of seventh order polynomial representing driver's decision strategy.

higher acceleration (similarly for negative values). It is worth mentioning that values of acceleration are not symmetric for positive and negative control error. Recreating the same procedure for all v_{ref} tracking experiments, the result is shown in Fig. 4.3a. It can be seen from the figure, that driver is not always consistent. To get consistent and also injective mapping $e_{\text{ctrl}} \to a_x$, I approximate the data using polynomial regression, see Fig. 4.3b. Points displayed in yellow are excluded from regression, since these do not represent the driver decision as I already discussed. Obtained is seventh-order polynomial function:

$$a(e_{\text{ctrl}}) = p_7\, e_{\text{ctrl}}^7 + p_6\, e_{\text{ctrl}}^6 + \ldots + p_1\, e_{\text{ctrl}} + p_0 \;, \tag{4.6}$$

see coefficients p_n in the Table 4.1. The function $a(e_{\text{ctrl}})$ is used only for $e_{\text{ref}} \in \langle -10; 10 \rangle$; extrapolation for values outside the interval are done using values at the end-points of the interval. To model dynamics of driver's decisions, I use first order system with time constant $T_s = 0.2\,\mathrm{s}$ and unity DC-gain [9]:

$$H_{\text{driver}}(s) = \frac{5}{s+5} \;, \tag{4.7}$$

Reference on position

So far, the model of the Driver has v_{ref} or more precisely the e_{ctrl} as an input. For further experiments, I have found necessary to have a model of driver reacting to distance control error $e_{\text{dist}} = s_{\text{ref}} - s$ instead of e_{ctrl}. Distance reference s_{ref} can represent a distance to next tram stop or to rear-end of forward vehicle. I propose a cascade structure where reference on position is transferred to reference on speed so I can use the strategy (4.6).

For transferring s_{ref} to v_{ref}, I design a following control law: if e_{dist} is higher than threshold K_{thres}, v_{ref} is set to maximal allowed speed $v_{\max}$. On the other hand, if $e_{\text{ref}} \leq K_{\text{thres}}$, v_{ref} is given as output of PID controller. Following equation characterize the control law:

$$v_{\text{ref}} = \begin{cases} k_p e_{\text{dist}} + k_i \int e_{\text{dist}} + k_d \dfrac{de_{\text{dist}}}{dt} & \text{if } e_{\text{dist}} \leq K_{\text{thres}}\ , \\ v_{\max} & \text{otherwise}\ . \end{cases} \tag{4.8}$$

Reaction to target tram

The last block which I include to the Drive is how s_{ref} of subject tram changes in reaction to target tram. This will be important for experiments in which two trams are moving on the same track and position reference of subject tram changes due to target tram. I propose an Algorithm 4.1 which should simulate the behaviour of human driver.

4.2.2 Power drive mechatronics

Structure of power drive mechatronics (PDM) subsystem is shown in Fig. 4.4a. I first assume that output from Driver subsystem is a control action u, giving a position of the control lever. The first block in this subsystems models the mechanics of the lever. In T3-type trams, this lever can only be in discrete positions which in consequence quantizes a power driving the tram motor. This can be seen from acceleration a_x in Fig. 4.4b where acceleration is decreasing in stairstep manner. For forward motion, the lever has seven positions (levels). From experiments in Poruba depot, I identify maximal acceleration to be $\approx 1.75\,\text{m}\,\text{s}^{-2}$. Input u is thus quantized with 0.25 quantization interval.

Second block, first-order system, is black-box representation of the transfer of lever position to torque generated by a motor. This first order system is again characterized by a time constant T_{pdm} and a gain K_{pdm}. I chose the T_{pdm} and K_{pdm} in order to create such torque, that in consequence gives desired longitudinal acceleration of the tram (to be equal to $a(e_{\text{ctrl}})$).

4.2.3 Longitudinal motion dynamics

This subsystem is responsible for transferring torque generated by a motor T_{mot} to longitudinal motion of a tram. The model of this subsystem is based on the model developed in [19].

Algorithm 4.1: Change of SV's position reference s_{ref} in reaction to TV's position.

```
Input  : Position of TV (rear-end) $s_{\text{TV}}$, position of SV $s_{\text{SV}}$, velocity of SV $v_{\text{SV}}$,
         position of the destination (tram stop) $s_{\text{stop}}$, safety margin $d_{\text{safe}}$, relative
         distance to start $d_{\text{start}}$.
Output : Position reference $s_{\text{ref}}$.

if $s_{\text{TV}} \leq s_{\text{stop}}$ then
    if $v_{\text{SV}} = 0$ then
        $\Delta s \leftarrow s_{\text{TV}} - s_{\text{SV}}$                    // Relative distance of trams
        if $\Delta s \leq d_{\text{start}}$ then
            $s_{\text{ref}} \leftarrow s_{\text{SV}}$                           // Stay at current position
        else
            $s_{\text{ref}} \leftarrow s_{\text{TV}} - d_{\text{safe}}$
        end
    else
        $s_{\text{ref}} \leftarrow s_{\text{TV}} - d_{\text{safe}}$
    end
else
    $s_{\text{ref}} \leftarrow s_{\text{stop}}$                  // TV is not in the path to the destination
end
```

To model this subsystem, I use framework of bond graphs. Extensive description of system modelling using bond graphs is given for instance in [5, 14]. To create the bond graph, it is important to track the power flow in the subsystem. Main power-blocks, which I identified in the subsystem, are shown in Fig. 4.5a.

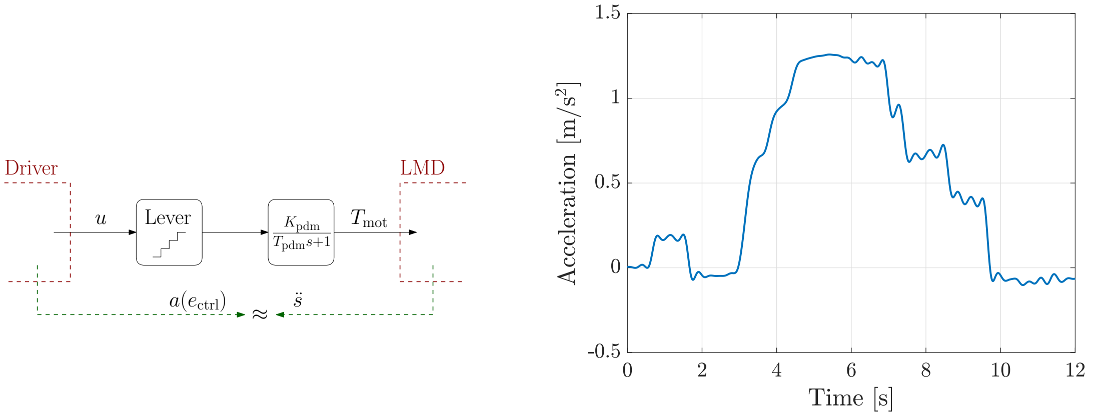
Motor (PDM) is the main source of the power entering the subsystem. I assume that motor creates torque T_{mot} independent of its speed ω_{mot} and thus I model the motor as a effort source given by equation:

$$T_{\text{mot}} = T(u); . \tag{4.9}$$

Energy from the motor is transferred into an angular motion of the wheel which I model using an inertance defined by an equation:

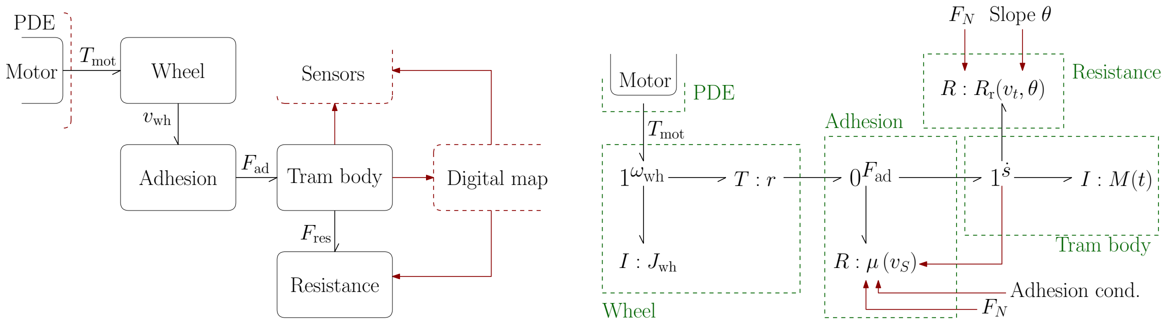
$$\omega_{\text{wh}}(t) = \frac{1}{J_{\text{wh}}} \int T_{\text{tot}}(t) dt \ , \tag{4.10}$$

where J_{wh} is moment of inertia of a wheel and $T_{\text{tot}}(t)$ is a total torque action on the wheel.



(a) **(b)**

Figure 4.4: **(a)** Block diagram of power drive mechatronics subsystem. Input u is used as lever position. **(b)** Acceleration a_x changes in stairsteps manner due to quantization of the control lever.



(a) **(b)**

Figure 4.5: **(a)** Block diagram of power flow in LDM subsystem depicted by half arrow (power bond) and interaction (signal based) with other subsystems (Sensors and Digital map). **(b)** Bond graph of the longitudinal dynamics motion subsystem.

Angular velocity $\omega_{\mathrm{wh}}(t)$ is then transferred to tangential speed $v_{\mathrm{wh}}(t) = r\omega_{\mathrm{wh}}(t)$ where r is radius of the wheel. Another block in the path of the power flow corresponds to adhesion of a rail and a wheel. Model of adhesion is important to be able determine maximal tractive force (under certain conditions) which can be used for braking. I model the adhesion as non-linear resistance (dissipation of the energy) which creates adhesion force $F_{\mathrm{ad}}(t)$:

$$F_{\mathrm{ad}}(t) = \mu(v_S)F_N(t)\ , \tag{4.11}$$

where $F_N(t) = M(t)g$ is a normal force, $\mu(v_S)$ is adhesion coefficient dependent on slip

Rail surface conditions	$\mu_{\max}$	K_S
Dry	0.3	0.72
Medium	0.2	0.72
Wet	0.1	0.72

Table 4.2: Coefficients defining adhesion under certain rail surface conditions. For further explanation, see [17].

velocity $v_S = v_{\mathrm{wh}} - \dot{s}$. To compute adhesion coeficient, I use formula from [17]:

$$\mu(v_S) = \frac{2K_S\mu_{\max}^2 v_S(t)}{\mu_{\max}^2 v_S^2(t) + K_S^2}\ , \tag{4.12}$$

where K_S and $\mu_{\max}$ are coefficients given by adhesion condition of the track, see Table. 4.2. Due to the adhesion, energy accumulated in the wheel velocity is then transferred to longitudinal velocity of the tram. I model this energy accumulation in motion of the tram again as inertance:

$$\dot{s}(t) = \int \frac{1}{M(t)} F_{\mathrm{tot}}(t)dt\ , \tag{4.13}$$

where $M(t)$ is a time varying mass of the tram body and $F_{\mathrm{tot}}(t)$ is total force acting on the tram body. Since tram usually cannot travel backwards, I added a constrain on speed: $\dot{s} \geq 0$. Lastly, energy contained in the tram motion is dissipated due to the resistive force F_{res} (rolling and grade resistance). For the railroad vehicles, I found that this can be modelled using Davis formula [8]:

$$F_{\mathrm{res}}(t) = 1000F_N(t)\left(0.1\frac{\theta\pi}{180} + 2.5 + \frac{(3.6\dot{s}(t))^2}{850}\right)\ . \tag{4.14}$$

Formula (4.14) was developed by *AnsaldoBreda* company for the tram *Sirio* but I assume that it should be approximately valid for other types of trams. I altered Eq. (4.14) from [8] so M is in kilograms, θ in radians and $\dot{s}$ in meters per second. Final bond graph of this subsystem is shown in Fig. 4.5b. It is now straightforward to get state-space representation of the model or model in Simulink from of this bond graph. For completeness, I give the resulted equation of LMD in Appendix C.

4.2.4 Parameters of the model

Before adding noise to the measured signals, let me first identify (to set) all remaining parameters in the model to have matching generated signals with filtered measured signals. All remaining parameters are listed in the Table 4.3. Here, I explain in detail chose of

Parameter	Value	Unit	Subsystem
v_{max}	65	$\mathrm{km\,h^{-1}}$	Driver
K_{thres}	100	-	Driver
d_{safe}	2	m	Driver
d_{start}	10	m	Driver
PID: k_p	0.15	-	Driver
PID: k_i	0	-	Driver
PID k_d	0.01	-	Driver
T_{pdm}	1	s	PDM
K_{pdm}	6100	-	PDM
J_{wh}	30	$\mathrm{kg\,m^2}$	LMD
M	16000	kg	LMD

Table 4.3: All remaining parameters of high fidelity model.

parameters. I took value of M from online source[1]. Value of J_{wh} is only an approximation based on available parameters (radius of the wheel r). Even thought the maximal speed is $v_{\mathrm{max}} = 65\,\mathrm{km\,h^{-1}}$, I set maximal speed only to $v_{\mathrm{max}} = 50\,\mathrm{km\,h^{-1}}$ which is approximately usual speed in urban areas. I set values of K_{thres}, d_{safe} and d_{start} to approximately reflect the reality. Constants of PID controller needed to be set to get control on position without overshoot (non-negative impulse response control). In case of position reference given by a target tram, overshoot might lead to collision. I identified parameters T_{pdm} and K_{pdm} from experiments in Poruba depicted on Fig. 4.2a by setting the same speed reference to the model. See comparison of acceleration from experiments and the model on Fig. 4.6a and see comparison of measured speed y_v and $\dot{s}$ on Fig. 4.6b. There are clearly some inaccuracies, mainly in acceleration, for instance the transient event around $\approx 27\,\mathrm{s}$ in Fig. 4.6a.

4.2.5 Model of sensory measurement

Last thing which is needed to add to the model is model of the sensors. To get measurements y_s, y_v and a_x, I only add noise to s, $\dot{s}$ and $\ddot{s}$, respectively. Other measurements a_y and ω_z need to be generated using the digital map.

Position measurement model

Accuracy of GPS position is not contained in the data from Herman's UCU. I, therefore, assume the accuracy to be 5 m. To model GPS position inaccuracy, I first take ground-truth position s generated from the model. Using the digital map, I then find coordinates (latitude and longitude) of point $P(s)$. Then, the point P' is selected in distance $d \sim (0, 25)$ with center in $P(s)$ and uniformly selected heading. This newly selected point P' is then again

[1] https://cs.wikipedia.org/wiki/Tatra_T3

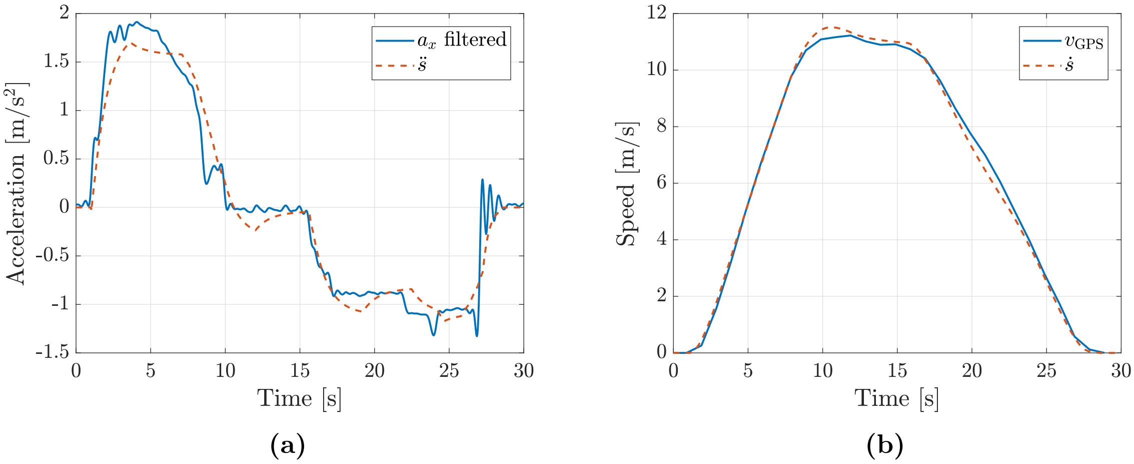


Figure 4.6: **(a)** Comparison of filtered acceleration a_x obtained from real experiments and acceleration $\ddot{s}$ generated from the high fidelity model. **(b)** Comparison of speed v_{GPS} obtained from real experiments and speed $\dot{s}$ generated from the high fidelity model.

projected onto the track to obtain $P'(s)$ and its position s on track.

In addition, I also model GPS dropouts/outage. To get an idea of the real typical frequency of GPS dropouts, I use data gathered by Herman's UCU. In UCU, GPS is sampled (together with all other measurements) with sampling period $T_s = 100\,\mathrm{ms}$ and every sample contains a timestamp. In every discrete time instance, I compute the difference of GPS timestamps using most recent GPS data and GPS data used in previous time instance. See the results in Fig. 4.7a displaying time difference of GPS over time and histogram 4.7b of differences. GPS dropout occurs when the time difference is higher than 100 ms. It can be seen that there are mainly dropouts of one GPS measurement. From the histogram 4.7b, I estimate the probability of such one-step GPS dropout to be $p_{\mathrm{out}} = 0.2549$. It can also be seen that GPS dropouts with higher duration (in Fig. 4.7a between time 4000 s and 6000 s) usually exhibit in a burst. Difference equal to 0 ms represents the situation, when in current time instance, GPS data from previous time instance is used.

Speed measurement model

I assume that both speed and position are measured by GNSS module. Typical accuracy of measured speed in data obtained by Herman's UCU is $0.5\,\mathrm{m\,s^{-1}}$. I model this inaccuracy as a random walk with values bounded in an interval $\langle -0.5; 0.5\rangle$. See an example of noised measurement y_v of a speed in Fig. A.1.

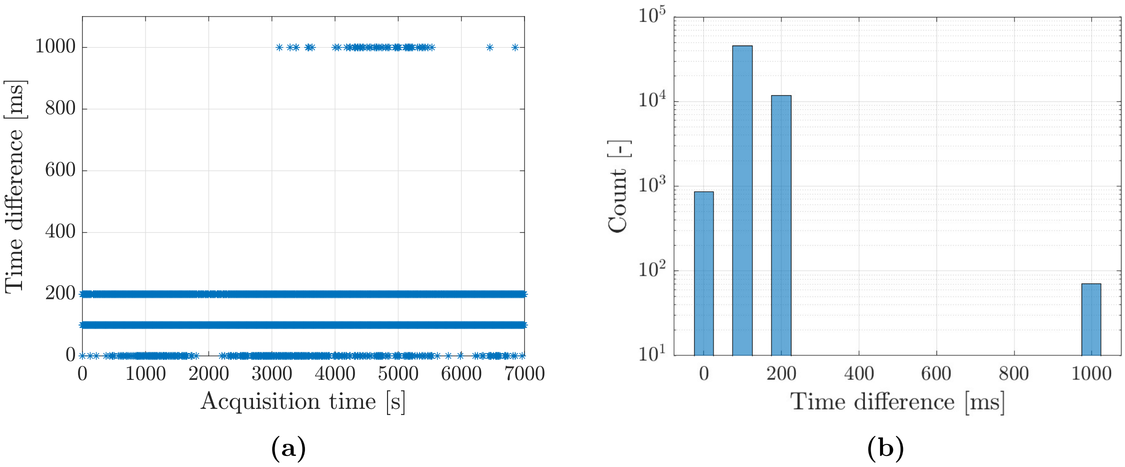


Figure 4.7: **(a)** Time difference of GPS data over time. **(b)** Histogram of time differences. Note that y-axis has logarithmic scale.

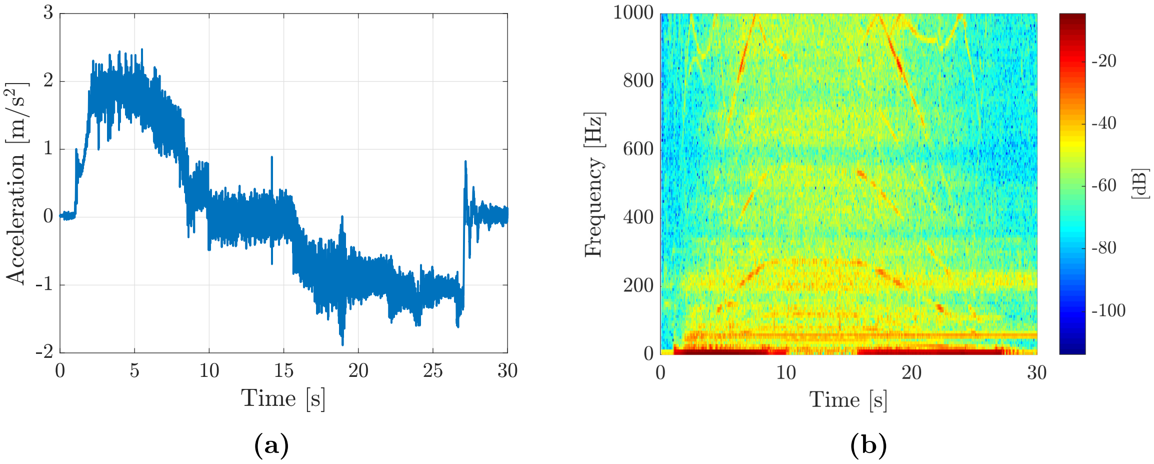


Figure 4.8: Data from experiment in Poruba depot, tram Tatra T3. **(a)** Acceleration a_x. **(b)** Spectrogram of a_x.

Acceleration measurement model

To model noise contained in acceleration, I use data from experiments in Poruba. First, see measured acceleration a_x and its spectrogram in Fig. 4.8a and 4.8b. From spectrogram of the a_x signal, I identified and divide the main contained noise into three components (I denote constants corresponding to particular component using subscript a, b and c respectively).

First component represents vibrations with constant amplitude on frequency 56 Hz. Source of these vibrations might be some mechatronic part in the tram (e.g. cooling). I model this component as sinus signal with constant amplitude and frequency. Even thought this is not exactly true, I assume that this component is active only when a tram is moving $\dot{s} > 0$. From power spectral density, I identified the amplitude of these vibrations to be $A_\mathrm{a} = 0.117$.

Second component is white noise with variance depending on the speed of the tram. This component includes noise of the MEMS sensor but also models the vibration of body of the tram due to higher speed. From power spectral density, I identified standard deviation of this noise as

$$\sigma_\mathrm{b} = \sigma_\mathrm{const} + \sigma_\mathrm{var} = 0.015 + 0.009\dot{s} \ . \tag{4.15}$$

Last component represents high-power vibrations with linearly changing frequency (emerged red curves in the spectrogram). Obviously, all emerged curves reflect only one source of vibration created as multiplication of fundamental frequency. When comparing the spectrogram with corresponding speed in Fig. 4.6b, it can be clearly seen, that the frequency is dependent on the speed $\dot{s}$ (the same phenomenon can be seen in Fig. 2.2b). My assumption is, based on values of frequencies and the power, that vibrations are caused by rotation of the wheels. Therefore, I model this component using sinus signal with varying frequency:

$$x_\mathrm{sin} = A_c \sin\left[\Phi_0 + 2\pi \int_0^t f(\tau) d\tau\right] \ , \tag{4.16}$$

where A_c is an amplitude, Φ_0 is initial phase and $f(t)$ is a function describing dependence of frequency on time. I model $f(t)$ as linear dependence on speed:

$$f(t) = k_\mathrm{c}\dot{s}(t) \ , \tag{4.17}$$

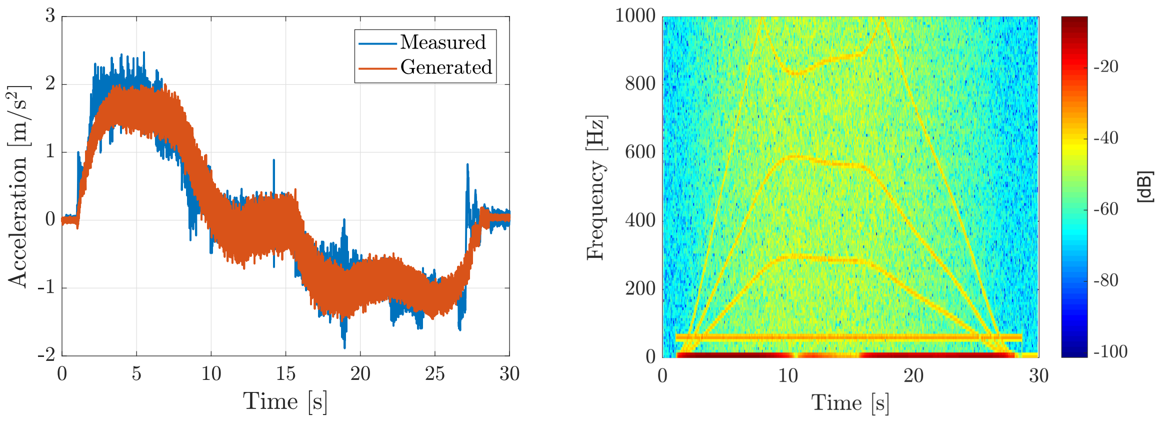
where $k_c = \frac{288}{11}i, \ i = \{1, 2, 3\}$ is slope of speed-frequency dependence which I identified from Fig. 4.6b and 4.8b. Combining Eq. (4.16) and (4.17) and setting $\Phi_0 = 0$:

$$x_\mathrm{sin} = A_c \sin\left[2\pi \int_0^t k_c \dot{s}(\tau)\right] = A_c \sin\left[2\pi k_c s(t)\right] \ . \tag{4.18}$$

I identified amplitude $A_c = 0.06$ again from power spectral density. Finally, see comparison of generated and measurement of a_x in Fig. 4.9a. To check that generated signal has similar spectrogram as measured a_x, see Fig. 4.9b.

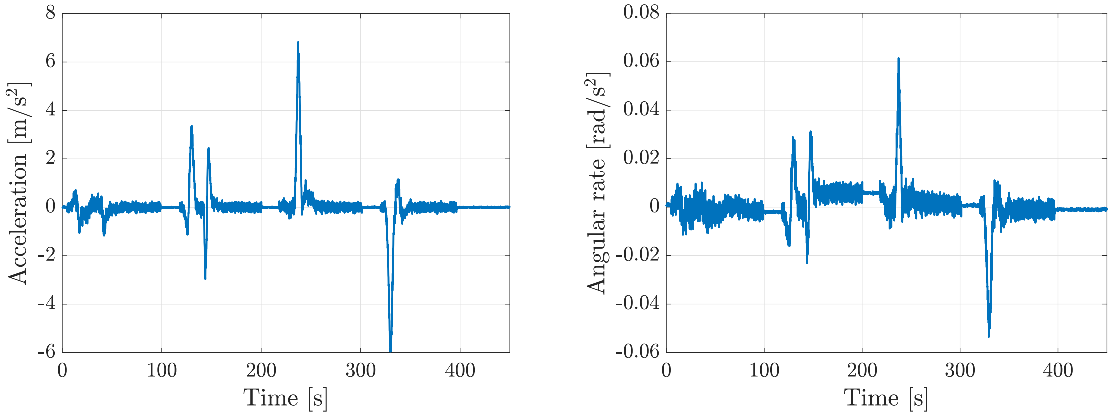
Curvature measurement model

Creating a measurement model of a_y and ω_z is now quite straightforward. With identified horizontal curvature of a track c_h, measurements are given as: $a_y = c_h\dot{s}^2$ and $\omega_z = c_h$. To add noise into the a_y, I use the same noise signals as in case of a_x. Noise added to ω_z also has the same components as a_x but the amplitude of deterministic component divided by 50 and a standard deviation of stochastic components also divided by 50. See examples of generated noise signals in Fig. 4.10. I do not have a real measurement for comparison. Note that in turns, simulated lateral acceleration is quite high in the amplitude. This is caused by



(a) **(b)**

Figure 4.9: **(a)** Comparison of generated a_x and generated a_x from the model. **(b)** Spectrogram of generated a_x.



(a) Acceleration a_y

(b) Angular rate ω_z

Figure 4.10: Generated measurements from simulated motion of a tram on track with several right and left turns.

the fact that I do not include a restriction on speed in turns and thus a tram does not drive through the turn with realistic speed.

5 | State estimation

In this chapter, I will describe and use Kalman filter (KF) for state estimation of a tram motion and evaluate its performance using extensive (Monte Carlo) simulations.

Kalman filter is a well-known mechanism used in many applications related to state estimation of the stochastic system. Numerous monographs deal with KF, for instance [2] or [20] thus I will state only key equations to introduce the notation. I will use two types of discrete-time KF: linear and unscented. The former is a fundamental form of KF for state estimation. The problem with linear KF is, however, that the model used for estimation must be linear. This is not satisfied for output Eq. (4.4). Nevertheless, measurement ω_z does not contain crucial information of the position and thus can be removed with only minor loss of information. Output equation then becomes linear and linear KF can be used which has simpler implementation. The latter, Unscented KF (UKF), is dedicated for estimation using the non-linear model. The reason, why I chose UKF instead of Extended KF (also for non-linear models), is that extended KF requires the model as a symbolic function whereas output model (4.4) is in the form of a look-up table. In addition, UKF should theoretically be able to handle non-linearity in the output model better than Extended KF.

After a brief introduction to KF algorithms, I will evaluate the performance of state estimation using (Monte Carlo) simulations. Data for the simulations are generated from the high fidelity model created in Sec. 4.2.

5.1 Estimation algorithms

5.1.1 Linear discrete time KF

Linear discrete time KF estimates state of a stochastic, linear and discrete time system:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \ , \\ y_k &= H_k x_k + v_k \ . \end{aligned} \tag{5.1}$$

Noises w_k and v_k are assumed to be white, zero mean and uncorrelated with covariance matrices Q_k and R_k:

$$\begin{aligned} E[w_k w_j^T] &= Q_k \delta_{k-j} \ , \\ E[v_k v_j^T] &= R_k \delta_{k-j} \ , \\ E[w_k v_j^T] &= 0 \ , \end{aligned} \tag{5.2}$$

where $E[X]$ is an expectation of random variable X and δ_{k-j} is Kronecker delta function. Kalman filter for the system (5.6) with assumptions (5.2) is given by following equations:

- Initialization:

$$\begin{aligned} \hat{x}_0 &= E[x_0] \ , \\ P_0 &= E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^\mathsf{T}] \ . \end{aligned} \tag{5.3}$$

- Each time instance $k = 1, 2, \ldots$ contains two steps. Former is called *Time* (prediction) step and contains time propagation of estimated state $\hat{x}_{k-1}$ and covariance of the estimation error P_{k-1} using state equations:

$$\begin{aligned} \hat{x}_{k|k-1} &= F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1} \ , \\ P_{k|k-1} &= F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1} \ , \end{aligned} \tag{5.4}$$

  and latter is *Data* (correction) step which contains a least-squares estimation using measured data (if no measurement is available in step k, Data step is skipped.):

$$\begin{aligned} K_k &= P_{k|k-1}H_k^T(H_kP_{k|k-1}H_k^T + R_k)^{-1} \ , \\ \hat{x}_k &= \hat{x}_{k|k-1} + K_k(y_k - H_k\hat{x}_{k|k-1}) \ , \\ P_k &= P_{k|k-1} - K_kH_kP_{k|k-1} \ . \end{aligned} \tag{5.5}$$

5.1.2 Unscented KF

Unscented KF is an algorithm for state estimation of stochastic, non-linear discrete time system:

$$\begin{aligned} x_k &= f(x_k, u_k, t_k) + w_k \ , \\ y_k &= h(x_k, t_k) + v_k \ . \end{aligned} \tag{5.6}$$

with the same restriction on the noises w_k and v_k as in the case of Linear KF. Unscented KF algorithm is given by following equations [7]:

- Initialization: same as in case of Linear KF (5.3).

- In each time step $k = 1, 2, \ldots$, time step starts with generating the *sigma* points $\hat{x}_{k-1}^{(i)}$ and weights W_i:

$$\begin{aligned} \hat{x}_{k-1}^{(0)} &= \hat{x}_{k-1} \ , \quad W_0 = \kappa(n+\kappa)^{-1} \ , \\ \hat{x}_{k-1}^{(i)} &= \hat{x}_{k-1} + \tilde{x}^{(i)} \ , \quad W_i = (2n + 2\kappa)^{-1} \ , \qquad i = 1, \ldots, 2n \ , \\ \tilde{x}^{(i)} &= \left(\sqrt{(n+\kappa)P_{k-1}}\right)_i^\mathsf{T} \ , \qquad i = 1, \ldots, n \ , \\ \tilde{x}^{(n+i)} &= -\left(\sqrt{(n+\kappa)P_{k-1}}\right)_i^\mathsf{T} \ , \qquad i = 1, \ldots, n \ , \end{aligned} \tag{5.7}$$

where n is an order of the system, $\left(\sqrt{(n+\kappa)P}\right)_i$ is i-th row of the matrix and $\kappa = 3-n$ (theoretical optimum for normal distribution). Here, square root of a matrix $\sqrt{P}$ is defined as $\sqrt{P}^{\mathsf{T}}\sqrt{P} = P$. Time propagation of $\hat{x}_{k-1}$ and P_{k-1} is defined as:

$$\begin{aligned}
\hat{x}_k^{(i)} &= f\left(\hat{x}_{k-1}^{(i)}, u_k, t_k\right) \ , \\
\hat{x}_{k|k-1} &= \sum_{i=0}^{2n} W_i \hat{x}_k^{(i)} \ , \\
P_{k|k-1} &= \sum_{i=0}^{2n} W_i \left(\hat{x}_k^{(i)} - \hat{x}_{k|k-1}\right)\left(\hat{x}_k^{(i)} - \hat{x}_{k|k-1}\right)^{\mathsf{T}} + Q_{k-1} \ .
\end{aligned} \tag{5.8}$$

Data step again starts with generation of the sigma points $\hat{x}_{k|k-1}^{(i)}$ using Eq. (5.7) with latest estimation $\hat{x}_{k|k-1}$ and $P_{k|k-1}$ instead of $\hat{x}_{k-1}$ and P_{k-1}. Least-squares estimation using measured data is defined as:

$$\begin{aligned}
\hat{y}_k^{(i)} &= h\left(\hat{x}_{k|k-1}^{(i)}, t_k\right) \ , \\
\hat{y}_k &= \sum_{i=0}^{2n} W_i \hat{y}_k^{(i)} \ , \\
P_y &= \sum_{i=0}^{2n} W_i \left(\hat{y}_k^{(i)} - \hat{y}_k\right)\left(\hat{y}_k^{(i)} - \hat{y}_k\right)^{\mathsf{T}} + R_k \ , \\
P_{xy} &= \sum_{i=0}^{2n} W_i \left(\hat{x}_{k|k-1}^{(i)} - \hat{x}_{k|k-1}\right)\left(\hat{y}_k^{(i)} - \hat{y}_k\right)^{\mathsf{T}} \ .
\end{aligned} \tag{5.9}$$

Finally:

$$\begin{aligned}
K_k &= P_{xy} P_y^{-1} \ , \\
\hat{x}_k &= \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_k) \ , \\
P_k &= P_{k|k-1} - K_k P_y K_k^{\mathsf{T}} \ .
\end{aligned} \tag{5.10}$$

5.2 Simulation scenario

To evaluate the state estimation, I will use Monte Carlo simulations to generate the data from high fidelity model developed in Sec. 4.2. I will create three datasets. All dataset will consist of $N = 100$ simulation runs each with a different seed of random number generator to generate different noise vectors of measurements. Each run lasts for $T = 450\,\mathrm{s}$. In the first dataset, I will assume no GPS outage whereas in the second and third dataset, I will define several intervals of GPS outages. The third dataset will differ from the second dataset in added constant bias in acceleration measurement.

The scenario (deterministic part) of the simulation is as follows: I place the generated

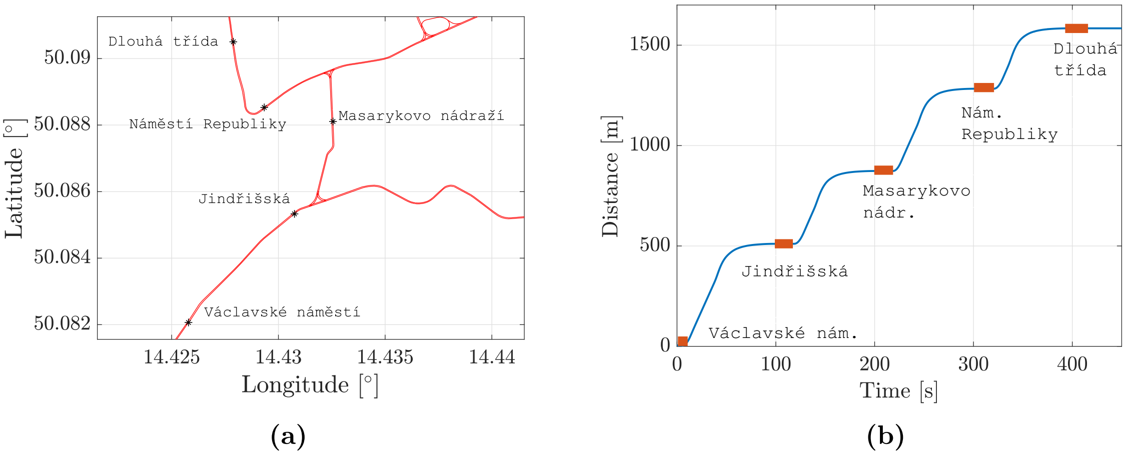


Figure 5.1: **(a)** Map of Prague, near Jindřišská street with displayed tram track and stops. **(b)** Generated travelled distance s corresponding to described scenario. Red rectangles represents stops at tram stops.

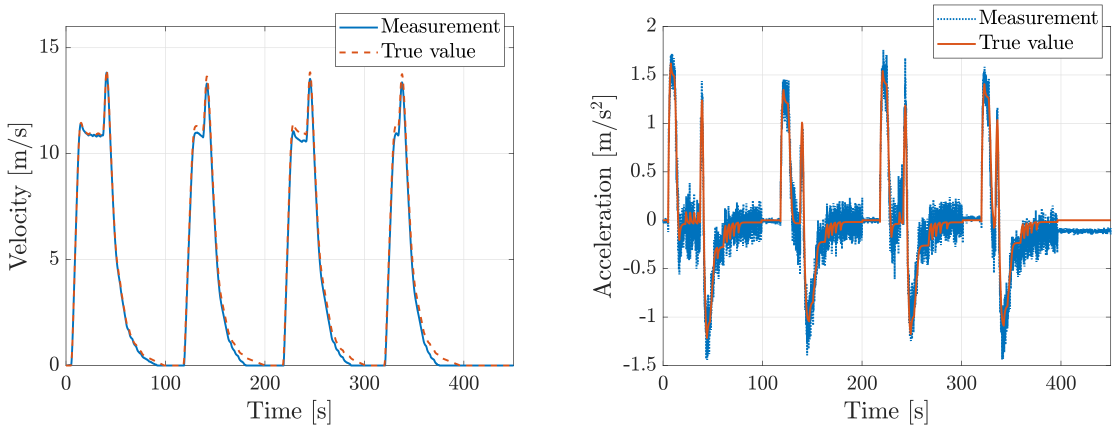
trajectory of a tram on tram track in Prague, near Jindřišská street. The tram starts at "Václavské náměstí" and travels through "Jindřišská", "Masarykovo nádraží" and "Náměstí Republiky" to "Dlouhá třída", see Fig. 5.1a. Total travelled distance is ≈ 1.6 km. At each tram stop, the tram stops for approximately 20 s. Also, total weight of the tram changes at tram stops due to varying number of passengers in the tram:

- "Václavské náměstí": 17 t,
- "Jindřišská": 21 t,
- "Masarykovo nádraží": 18 t,
- "Náměstí Republiky": 20 t.

I assume constant adhesion conditions (dry conditions) of the track. See the generated signals (true values) of position s, $\dot{s}$ and $\ddot{s}$, in Fig. 5.1b, 5.2a and 5.2b respectively. Note the changes of maximal acceleration and deceleration due to change in total weight of the tram in Fig. 5.2b.

All measurements are noised according to output model created in Sec. 4.2.5. See an example of noised data in Fig. 5.2a and 5.2b (distance s is transformed into GPS points and thus is not displayed). Note the bias caused by non-zero slope ≈ 400 s in Fig. 5.2b. Output sampling time is 100 ms in accordance with a sampling time of Herman's UCU.

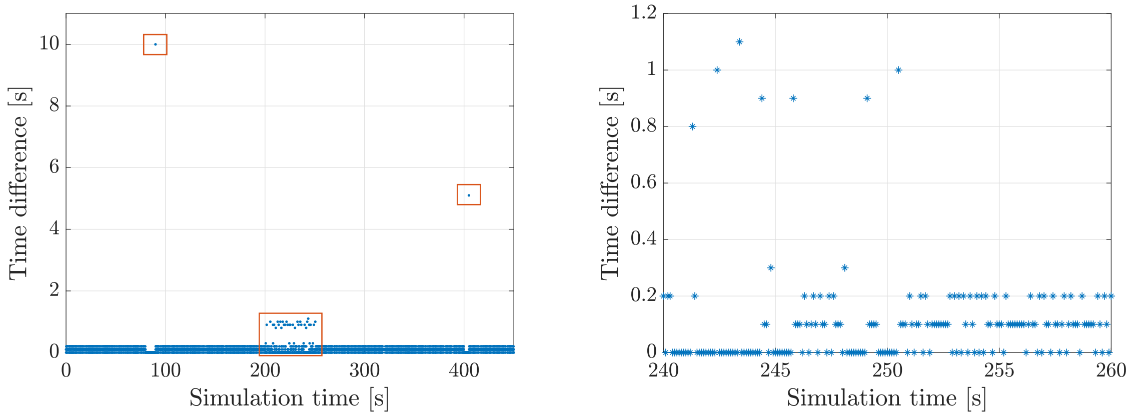
GPS outages for second and third dataset are displayed in Fig. 5.3a There are continuous one-step outages with probability ≈ 0.25, two high-duration outages and one burst of outages.



(a) Velocity

(b) Acceleration

Figure 5.2: Generated velocity $\dot{s}$ and acceleration $\ddot{s}$ corresponding to described scenario as true values and measurements (with added noise and bias).



(a)

(b)

Figure 5.3: **(a)** Generated GPS outages for the simulation scenario. Red rectangles mark significant outages. There are two high-duration outages: 10 s outage around $t = 80$ and 5 s outage around $t = 410$ and burst of ≈ 1 s outages around $t = 200$. **(b)** Close look at generated GPS outages.

5.3 Evaluation of estimation algorithms

In this section, I evaluate estimation using LKF and UKF on datasets generated in previous Sec. 5.2. For every simulation run and every state variable, I compute root-mean-square-error (RMSE) of true value y and estimated value $\hat{y}$:

$$e_{\mathrm{RMSE}}(y) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(y_t - \hat{y}_t\right)^2}\,. \tag{5.11}$$

		$e_{\text{RMSE}}(s)$	$e_{\text{RMSE}}(\dot{s})$	$e_{\text{RMSE}}(\ddot{s})$
LKF	Without outages	1.9763	0.1864	0.0638
	With outages, without bias	2.0339	0.1906	0.0641
	With outages, with bias	2.3498	0.4591	0.1979
UKF	Without outages	2.0949	0.1813	0.0648
	With outages, without bias	2.1753	0.1842	0.0652
	With outages, with bias	2.3672	0.4432	0.1973

Table 5.1: Evaluation of estimation algorithms on different datasets.

Then, I compute mean from all RMSEs across all simulation runs in the dataset. RMSE for all cases are summarized in Tab. 5.1.

5.3.1 Linear KF

For time-step, I use constant acceleration model (4.1) with $\Delta t = 0.1$. As output model, I use only linear part of model (4.3). System matrices are:

$$F_k = \begin{bmatrix} 1 & 0.1 & 0.005 \\ 0 & 1 & 0.1 \\ 0 & 0 & 1 \end{bmatrix}, G_k = 0 \, , H_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.12}$$

and covariance matrices are:

$$Q_k \approx \begin{bmatrix} 0 & 0 & 0.0002 \\ 0 & 0.0003 & 0.005 \\ 0.0002 & 0.005 & 0.1 \end{bmatrix}, R_k = \begin{bmatrix} 25 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}. \tag{5.13}$$

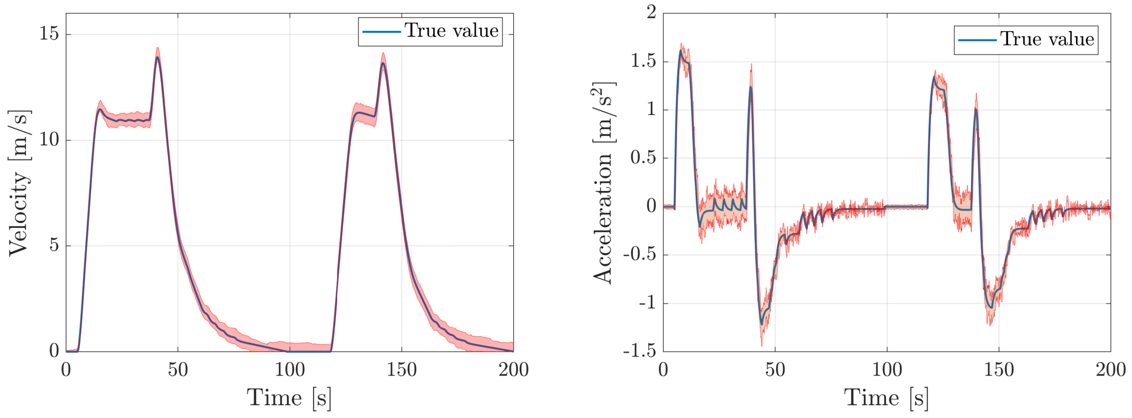
I initialize the LKF algorithm with:

$$P_0 = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \hat{x}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\intercal}. \tag{5.14}$$

In addition, to refine the estimation, I use the digital map to obtain slope $\theta(\hat{s}_k)$ at estimated position $\hat{s}_k$ which is then used to de-bias acceleration measurement.

Without GPS outages

First, I evaluate performance of LKF on dataset with no GPS outages. Resulted state estimation of tram motion gives following RMSEs (across all simulation runs in the dataset): $e_{\text{RMSE}}(s) = 1.9763$, $e_{\text{RMSE}}(\dot{s}) = 0.1864$ and $e_{\text{RMSE}}(\ddot{s}) = 0.0638$. See Fig. 5.4a and 5.4b where true values are displayed with maximum and minimum bound of estimated values (red area).



(a) Velocity

(b) Acceleration

Figure 5.4: True values of acceleration and velocity with displayed bounds (maximum and minimum) of estimated values (red area) using LKF with no GPS outages. Shown are only values for $t \in \langle 0; 200\rangle$ for better legibility.

I do not display similar figure for distance since the difference in estimated values and real values is very subtle and thus would not be legible in a figure.

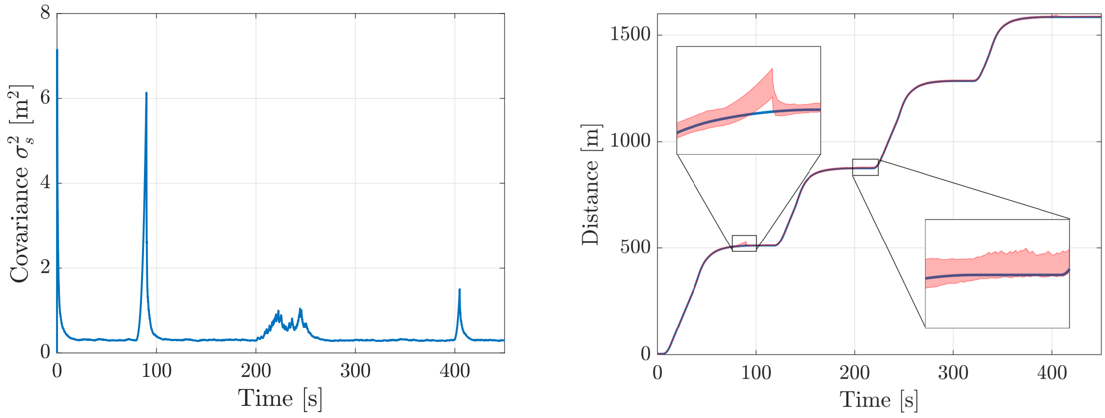
With GPS outages, without bias in acceleration

With GPS outages, the only remaining measurement is acceleration a_x. In linear case, output is given by Eq. (4.3) for available GPS and for GPS dropouts only by Eq. (4.5). Resulted state estimation of tram motion gives following RMSEs: $e_{\mathrm{RMSE}}(s) = 2.0339$, $e_{\mathrm{RMSE}}(\dot{s}) = 0.1906$ and $e_{\mathrm{RMSE}}(\ddot{s}) = 0.0641$.

Comparing the RMSEs with the previous case, RMSEs of estimation do not change dramatically. This is thanks to the quite accurate measurement of a_x without bias (removed by use of the digital map) and after filtration with the low-pass filter. I do not show the figures with estimation bound as in the previous case since results are similar (differences are not legible). One thing is, however, worth mentioning and showing. With $H = [0\ , 0\ , 1]$, system (F, H) is unobservable [3]. During the GPS outage, this causes an increase in the covariance of estimated states, see for instance covariance of estimated position in Fig. 5.5a. This is of course undesirable but reflects the nature of estimation of the unobservable system.

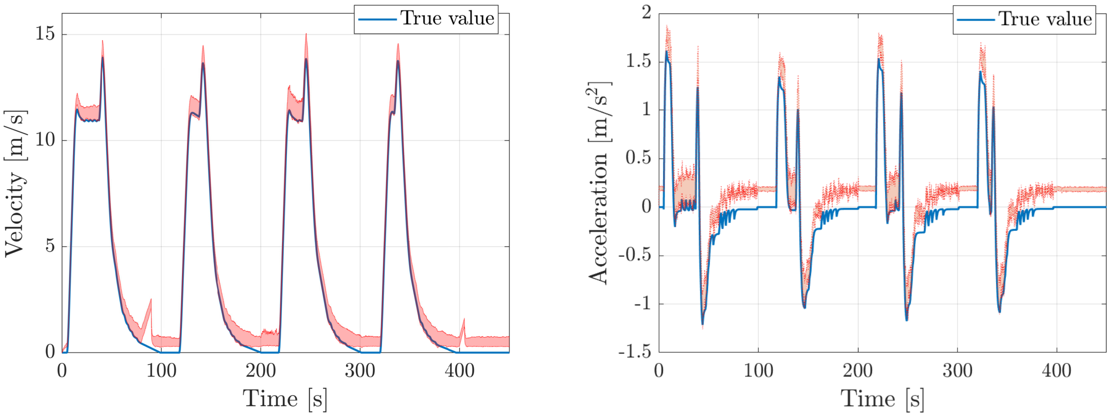
With GPS outages, with bias in acceleration

In this case, I use the same approach as in the case without bias in acceleration. Results are, however, significantly different. For bias $b_a = 0.2\,\mathrm{m\,s^{-2}}$ added to acceleration measurement a_y, I obtain following results: $e_{\mathrm{RMSE}}(s) = 2.3498$, $e_{\mathrm{RMSE}}(\dot{s}) = 0.4591$ and $e_{\mathrm{RMSE}}(\ddot{s}) = 0.1979$. In GPS outages, estimation of tram position in data step of KF is given only as double-



(a) **(b)**

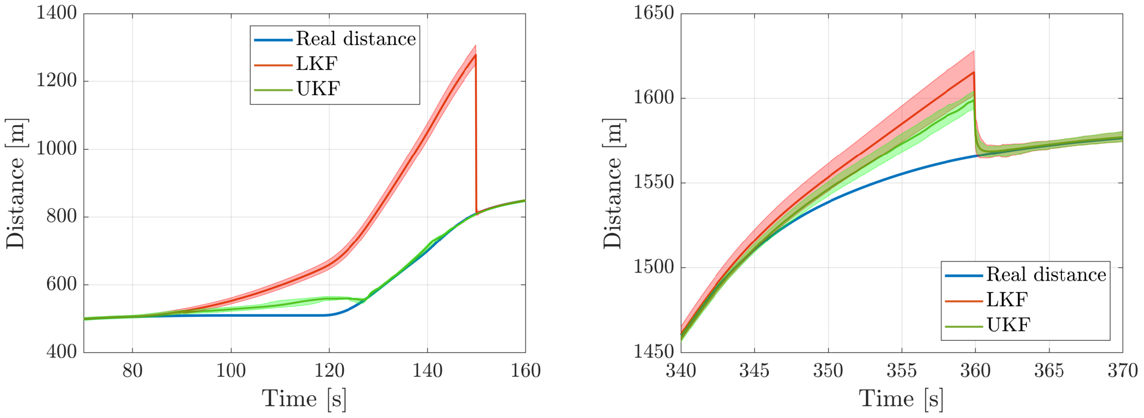
Figure 5.5: (a) Covariance error σ_s of position estimation influenced by GPS outages. **(b)** True value of travelled distance with displayed maximum and minimum bound of estimated values (red area) using LKF with GPS outages and with bias in acceleration measurement. Displayed are also close-looks at time interval of GPS outages.



(a) Velocity **(b)** Acceleration

Figure 5.6: True values of acceleration and velocity with displayed bounds (maximum and minimum) of estimated values (red area) using LKF with GPS outages and bias in accelerometer measurement.

integration of acceleration. Even small bias in acceleration is significantly propagated into the position estimation (in GPS outage intervals), see Fig. 5.5b. To be precise, in zoom-in plot near time $t = 90\,\mathrm{s}$, the estimated value differs from real value up to 20 m and for zoom-in plot around time $t = 200\,\mathrm{s}$, estimated value differs from real value up to 6 m. From Fig. 5.6a and 5.6b, it can be seen that estimation is biased which is caused by the bias b_a in acceleration measurement.



(a) Close look at first GPS outage **(b)** Close look at second GPS outage

Figure 5.7: Comparison of estimated travelled distance of a tram using LKF and UKF. Displayed are mean estimated values from 100 runs and min/max bounds of the estimation.

5.3.2 Unscented KF

For UKF, if applicable, I use the same setup as in case of LKF. The main difference is in output model where UKF uses both linear (4.3) and non-linear (4.4) parts. I set the covariance matrix R_k as diagonal matrix with following entries $[25; 0.25; 0.1; 0.01; 0.01]$ on the diagonal. I evaluate the UKF algorithm on the same scenarios as LKF. For the sake of brevity, I do not show the figures from the estimation since the differences with LKF are not legible. Results from the experiments are summarized in Tab. 5.1. When comparing the results with LFK, in terms of RMSE, UKF performs in all cases similarly or slightly worse. Slightly worse performance of UKF (based on RMSE) might be only a statistical error but also could mean that UKF is not able to sufficiently handle the non-linearity of output functions. Regardless off the cause, UKF is computationally more expensive than LKF and its performance is not significantly better and thus might not be appropriate for the real application. Additional research should be done in this area but in this thesis, I decided to use only LKF in further development for its simplicity and acceptable performance.

Lastly, to show that UKF can perform better in some scenarios than LKF, I create a simulation in which I add one 70 s duration GPS outage starting at simulation time 80 s and one 20 s outage starting at time 340 s. I then use both LKF and UKF for state estimation, see the comparison of distance estimation in Fig. 5.7 with displayed min/max bounds of the estimation from 100 simulation runs. Clearly, the UKF is able to track the motion of the tram (especially when the tram is moving) better than LKF during GPS outage. Such difference is, however visible only for high duration outages which are not usually present in reality. With only short GPS outages ≤ 10 s, performances of LKF and UKF are comparable.

6 | Collision avoidance

In this chapter, I will describe the design of collision avoidance/warning system (or more generally FVCWS). In particular, I will focus on solving two aspects of FVCWS: estimation of clearance $x_{c,k}$ and estimation of braking distance. I will first describe clearance estimation using V2V communication, utilizing absolute position estimation developed in Chapter 5. Regarding the clearance estimation, I will also deal with compensation of delay introduced by V2V communication.

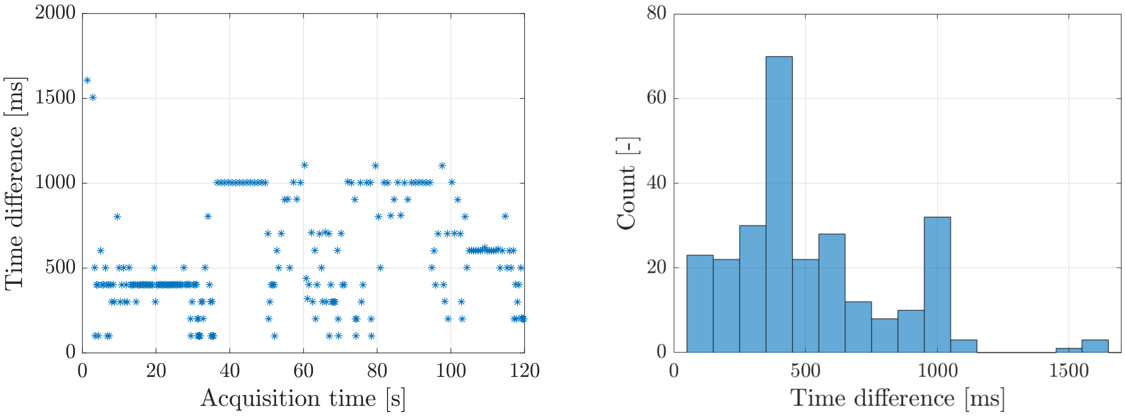
Then, I will describe the estimation of braking distance. Braking distance depends on various phenomena and thus is hard to estimate accurately. In particular, the collision avoidance system developed in this thesis serves only as an assistance for a driver. Thus, braking distance foremost depends on non-deterministic (unpredictable) behaviour of the driver. I address this problem by creating two models of driver's behaviour: simple, deterministic model and probabilistic model. Furthermore, (maximal) transfer from driver's control action to a deceleration of the tram heavily depends on adhesive conditions of the track and total weight of the tram (number of passengers in the tram).

Lastly, I will evaluate the performance of proposed design of collision avoidance/warning system using Monte Carlo simulations (similarly to the evaluation in Chap. 5). I will generate the data for simulation using high fidelity model created in Sec. 4.2.

6.1 V2V communication

V2V communication falls under the category of dedicated short-range communication technologies. It allows establishing a wireless (wifi-based) communication between devices. This communication is standardized as IEEE 802.11p. One of the components defined by the standard is cooperative awareness message (CAM) [11]. CAMs are used for transmitting relevant information to other vehicle and so they increase cooperative awareness in the network. Use case of such cooperative awareness is given for instance in [4] where authors deal with cooperative collision avoidance for vehicles on the highway. Defined by the standard, V2X communication device broadcasts CAM only if following condition is met:

- 100 ms passes from time of previously generated CAM and one of following is satisfied:
  - Change in heading from value in previous CAM is higher than 4 degrees,
  - Change in velocity from value in previous CAM is higher than $0.5\,\mathrm{m\,s^{-1}}$,
  - Change in position from value in previous CAM is higher than 4 meters,
  - More than 1 s passes from time of previously generated CAM.



(a)

(b)

Figure 6.1: (a) Time difference of consecutive CAM messages over time. **(b)** Histogram of time differences. Median of time differences is 400 ms.

6.1.1 CAM analysis from real data

To get an insight into a typical density of CAM based on generating conditions, I analyse real data gathered by Herman's UCU operated on-board of a tram in Ostrava. See Fig. 6.1a which displays time difference of consecutive CAM messages and in Fig 6.1b histogram of time differences. Time differences with values ≈ 1000 ms usually corresponds to the situation, when a vehicle (possibly a TV) is at rest (there is no such change in heading, velocity or position satisfy CAM generating condition). There are few time differences with value ≈ 1500 s which are probably caused by a packet loss or communication overhead. Clearly, receive of CAM is not periodic.

6.2 Method design

As already described at the beginning of this chapter, collision avoidance system consists of several sub-tasks. In addition, each task can be solved in several ways. Combining all possible solutions for all sub-task would give many possibilities. For sake of brevity, I design only two algorithms for collision warning system. First algorithm gathers 'average-values' solutions of the sub-tasks whereas second algorithm gathers conservative (in terms of collision avoidance) solutions.

6.2.1 Clearance estimation

First, I denote s_s to be position of leading surface of subject tram and s_t to be position of trailing surface of target tram. Clearance $x_{c,k}$ in time k is then defined as distance between these two surfaces:

$$x_{c,k} = s_{t,k} - s_{s,k} \; . \tag{6.1}$$

I assume that state estimation gives directly positions of s_s and s_t. This is reasonable assumption since estimated absolute position (depends on position of GPS antenna) differs from leading and trailing surfaces only by a constant (known) offset. I propose a following design of clearance estimation. Both subject and target trams estimate their state of motion using KF (developed in Chapter 5). Then, using V2V communication, target tram broadcasts its estimated state of motion (including timestamp) using CAM which is then received by a subject tram. Now, on-board of the subject tram, clearance can be computed.

The use of communication, however, cause two problems. First problem is communication (network) delay caused by limited bandwidth or communication overhead [16]. Thus, at current time k, the most recent available values are $s_{s,k}$ and $s_{t,k-D}$ where D is communication delay. Second problem is non-periodicity in receive of $s_{t,k}$ caused by rules of CAMs generation but also to possible packet loss. I propose two approaches to deal with these problems: state-based and conservative forward propagation of $s_{t,k-D}$ using state-space model.

State-based forward propagation

An intuitive approach of dealing with the delay is to use most recent received state of motion of target tram and propagate it into current time. Let $\hat{x}_{t,k-D}$ be estimated state of motion of target tram received by subject tram. For time propagation of $\hat{x}_{t,k-D}$ to current time k, I use deterministic part of the state-space model (4.1) with $\Delta t = D$. That is:

$$\hat{x}_{t,k} = \begin{bmatrix} s_{t,k} \\ \dot{s}_{t,k} \\ \ddot{s}_{t,k} \end{bmatrix} = \begin{bmatrix} 1 & D & \frac{1}{2}D^2 \\ 0 & 1 & D \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{t,k-D} \\ \dot{s}_{t,k-D} \\ \ddot{s}_{t,k-D} \end{bmatrix} . \tag{6.2}$$

This propagated state $\hat{x}_{t,k}$ then serves in estimation of the clearance in current time k. Furthermore, I use the time propagation to solve the problem with non-periodic CAMs receiving. When no CAM is received in current time, estimation from previous time $\hat{x}_{t,k-1}$ is used and propagated into current time using state-space model. Diagram of this approach is shown in Fig. 6.2.

Conservative forward propagation

Second approach, more conservative, is to use the same approach but assume worst-case scenario (in terms of collision avoidance application) as proposed in [1]. That is, I again use state-space model for forward propagation but in this case, I assume that target tram immediately starts braking after CAM is generated. That is:

$$\hat{x}_{t,k} = \begin{bmatrix} s_{t,k} \\ \dot{s}_{t,k} \end{bmatrix} = \begin{bmatrix} 1 & D \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_{t,k-D} \\ \dot{s}_{t,k-D} \end{bmatrix} - \begin{bmatrix} \frac{1}{2}D^2 \\ D \end{bmatrix} |a_{\text{targ}}| , \tag{6.3}$$

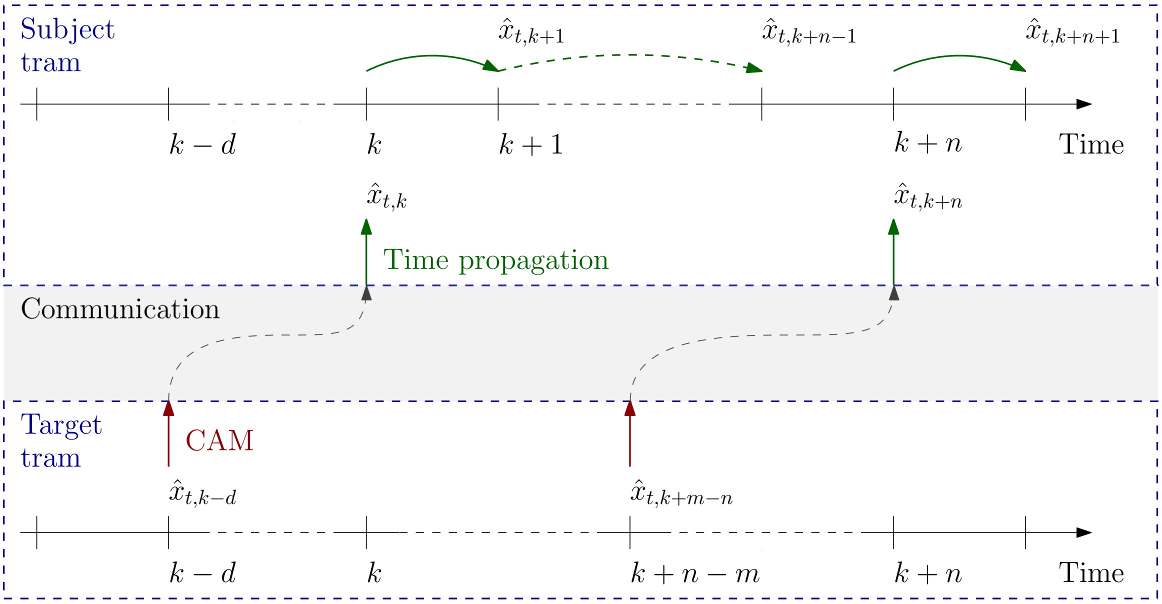


Figure 6.2: State-based forward propagation of delayed state of target tram $\hat{x}_{t,k-D}$. Red arrows represent generating of CAM. Green arrows represent time propagation usign state-space model.

where $|a_{\text{targ}}|$ is absolute value of maximal deceleration of target tram. As same as in the model, I assume the restriction $\dot{s}_{t,k} \geq 0$.

6.2.2 Braking distance estimation

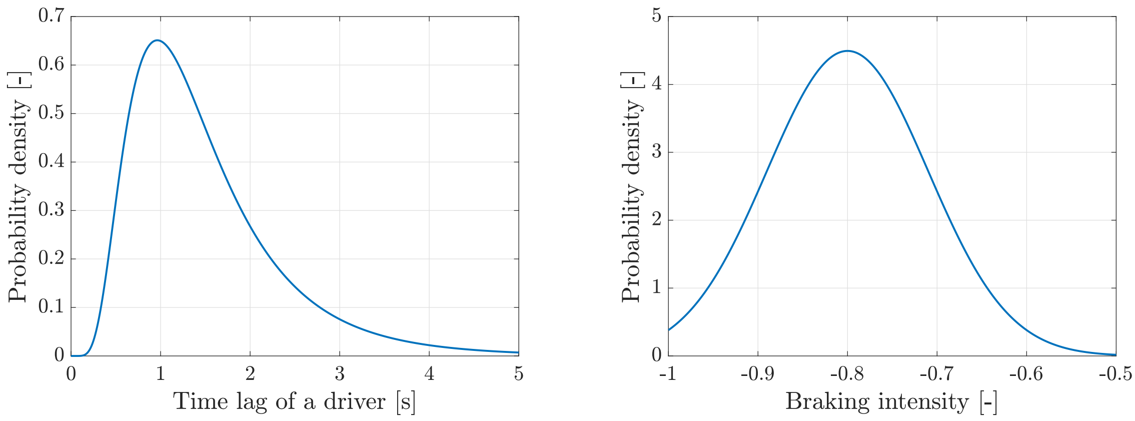
For collision avoidance application, I define braking distance d_{br} as a travelled distance from the time instance of collision warning (given by a FVCWS) to the time when tram stops. This can be written as:

$$d_{\text{br}} = \left(\dot{s}_k t_{\text{lag}} + \frac{1}{2}\ddot{s}_k t_{\text{lag}}^2\right) + \left(\frac{1}{2}(\dot{s}_k + \ddot{s}_k t_{\text{lag}})^2 a(u, a_{\text{subj}})^{-1}\right) , \tag{6.4}$$

where t_{lag} is reaction time of driver to collision warning and $a(u, a_{\text{subj}})$ is applied deceleration which depends on driver's chosen braking intensity u and maximal deceleration a_{subj} of subject tram. First term on right hand side (enclosed in brackets) in (6.4) represents travelled distance before driver reacts to the warning and second term represents distance to stop with velocity $(\dot{s}_k + \ddot{s}_k t_{\text{lag}})$ and applied constant deceleration $a(u, a_{\text{subj}})$.

Reaction time and braking intensity

It is intuitive that values of t_{lag} and u are not the same for all drivers. Less experienced drives might have slower reactions which increase t_{lag} or could underestimate the collision warning which affects the selection of u. Unfortunately, I do not have any real data from which I



(a) **(b)**

Figure 6.3: **(a)** Probability density function $f_{\text{pdf}}(t_{\text{lag}})$ of time lag t_{lag} of a driver. **(b)** Probability density function $f_{\text{pdf}}(u)$ of braking intensity u.

	Mean μ	Variance σ^2	Left bound	Right bound
$f_{\text{pdf}}(t_{\text{lag}})$	1.3	0.5476	N/A	N/A
$f_{\text{pdf}}(u)$	-0.8	0.09	-1	-0.5

Table 6.1: Parameters of probability density functions of time lag and braking intensity.

could identify values (or the distribution) of t_{lag} and u. To test the proposed algorithm of collision avoidance, I thus use models from [6]. Reaction time t_{lag} is modelled using log-normal distribution and braking intensity u is modelled by truncated normal distribution, see probability density function (PDF) in Fig. 6.3a and Fig. 6.3b respectively. Since braking intensity is modelled in [6] for cars, I use different parameters of PDF matching with trams, keeping the similar shape of PDF. Also, instead of direct values of deceleration, I create a PDF for braking intensity u (input set by a driver). The braking intensity $u = -1$ corresponds to deceleration equal to $-|a_{\text{subj}}|$ and $u = -0.5$ corresponds to $-0.5|a_{\text{subj}}|$. Parameters of the distributions are in Tab. 6.1.

Now, in order to use the information contained in PDFs for braking distance estimation, I propose two approaches. First is to use only mean values for braking intensity and reaction time. Second, again more conservative way, is to select values of $\tilde{t}$ and $\tilde{u}$ to cover the worst case with some chosen probability. I select to cover 95% of drivers. I find $\tilde{t}$ and $\tilde{u}$ for the chosen probability by integrating the PDFs $f_{\text{pdf}}(t_{\text{lag}})$ and $f_{\text{pdf}}(u)$ (getting distribution function) and finding values of t_{lag} and u for which the integrals equals to 0.95. That is, $\tilde{t}$

and $\tilde{u}$ satisfies following equation:

$$\int_{0}^{\tilde{t}} f_{\mathrm{pdf}}(t_{\mathrm{lag}}) dt_{\mathrm{lag}} = \int_{-1}^{\tilde{u}} f_{\mathrm{pdf}}(u) du = 0.95 \ . \tag{6.5}$$

Resulted values are $\tilde{t} = 3.2032$ and $\tilde{u} = -0.6517$.

Maximal deceleration estimation

I assume time-invariant value of a_{subj}. This has, of course, its caveat. When the estimated value of a_{subj} is in absolute value higher than real maximal deceleration, some collision warnings can be given too late. On the other hand, if a_{subj} is in absolute value lower than real maximal deceleration, it might give many false collision warnings. Even though such cases are undesirable, this approach is simple and computationally efficient.

I identify maximal deceleration for average based and conservative approaches using the high fidelity model. For average based approach, I set the total weight of the vehicle to 20 t and dry adhesive conditions whereas for conservative approach, I set total weight to 24 t and wet adhesive conditions. From the simulations, maximal deceleration for average based approach is $a_{\mathrm{subj,avg}} = -1.74\,\mathrm{m\,s^{-2}}$ and for conservative approach $a_{\mathrm{subj,cnv}} = -0.57\,\mathrm{m\,s^{-2}}$.

6.2.3 Collision warning

In its simplest form, collision warning is in current time k triggered if braking distance is higher than clearance (with some additional safety distance). It is, however, appropriate to incorporate a safety distance margin S_{margin} into the conditions of collision warning triggering. To be more precise, let $g(\mathcal{Y}^N)$ be critical function on observed data $\mathcal{Y}^N$ of size N. I define output of this function as:

$$g(\mathcal{Y}^N) = \begin{cases} 1 & \text{for collision warning ,} \\ 0 & \text{otherwise .} \end{cases} \tag{6.6}$$

For collision warning, observed data $\mathcal{Y}$ represents estimated clearance and braking distance. Thus, simple form of collision warning can be written as:

$$g(\mathcal{Y}^N) = \begin{cases} 1 & \text{if } (x_{c,k} - d_{\mathrm{br}}) \leq S_{\mathrm{margin}} \ , \\ 0 & \text{otherwise .} \end{cases} \tag{6.7}$$

There are several options how to alter this simple collision warning triggering. One is to use covariances $\sigma^2(x_k)$ from error-covariance matrix P of the estimation to incorporate uncertainty of estimation into the collision warning. For instance, if estimated position of subject tram at time k is $s_{s,k}$ and covariance is $\sigma^2(s_{s,k})$, its position is with ≈ 0.96 probability in the interval $\langle s_{s,k} - 2\sigma(s_{s,k}); s_{s,k} + 2\sigma(s_{s,k})\rangle$. The same applies to estimated speed and acceleration. This subsequently influence both uncertainty of d_{br} and $x_{c,k}$. To get collision warning in more conservative manner, 2σ is added to estimated state of the subject tram and subtracted from estimated state of target tram. Collision warning can be then written as:

$$g(\mathcal{Y}^N) = \begin{cases} 1 & \text{if } (\tilde{x}_{c,k} - \tilde{d}_{\mathrm{br}}) \leq S_{\mathrm{margin}} \ , \\ 0 & \text{otherwise} \ , \end{cases} \tag{6.8}$$

where both $\tilde{x}_{c,k}$ and $\tilde{d}_{\mathrm{br}}$ are computed using Eq. (6.1) and Eq. (6.4) with:

$$\begin{bmatrix} s_{t,k} \\ \dot{s}_{t,k} \\ \ddot{s}_{t,k} \end{bmatrix} := \begin{bmatrix} s_{t,k} \\ \dot{s}_{t,k} \\ \ddot{s}_{t,k} \end{bmatrix} - 2 \begin{bmatrix} \sigma(s_{t,k}) \\ \sigma(\dot{s}_{t,k}) \\ \sigma(\ddot{s}_{t,k}) \end{bmatrix} \ , \ \begin{bmatrix} s_{s,k} \\ \dot{s}_{s,k} \\ \ddot{s}_{s,k} \end{bmatrix} := \begin{bmatrix} s_{s,k} \\ \dot{s}_{s,k} \\ \ddot{s}_{s,k} \end{bmatrix} + 2 \begin{bmatrix} \sigma(s_{s,k}) \\ \sigma(\dot{s}_{s,k}) \\ \sigma(\ddot{s}_{s,k}) \end{bmatrix} \ . \tag{6.9}$$

6.3 Simulation scenario

I create two scenarios, first scenario as collision free and second scenario with two added collisions. These scenarios serves to evaluate performances of proposed collision warning algorithms.

6.3.1 Collision free scenario

Target tram (leading) is following the same trajectory as described in 5.2. The subject tram (proceeding) is travelling the same path but with 20 s delay, see Fig. 6.4a and Fig. 6.4b which show the generated trajectories and the clearance (relative distance between the trams) respectively. Similarly to target tram, total weight of subject tram also changes at tram stops due to varying number of passengers in the tram:

- "Václavské náměstí": 19 tonnes,
- "Jindřišská": 17 tonnes,
- "Masarykovo nádraží": 21 tonnes,
- "Náměstí Republiky": 22 tonnes.

To make the simulation more general (closer to a real situation), I assume GPS outages on both trams and also a small constant bias in acceleration measurement. In every simulation bias of acceleration measurement (again for both trams) is randomly chosen from a uniform distribution in the interval $\langle -0.2; 0.2\rangle$. GPS outages for target tram are the same as in the third dataset. GPS outages for subject tram again consist of continuous one-step outages with probability ≈ 0.25. I then add two burst outages, one overlapping with high-duration outage (around time 80 s) and one with burst outages (around time 200 s). Lastly, I add one outage at time 300 s with a duration of 9 s. See Fig. 6.5a displaying GPS outages for subject tram.

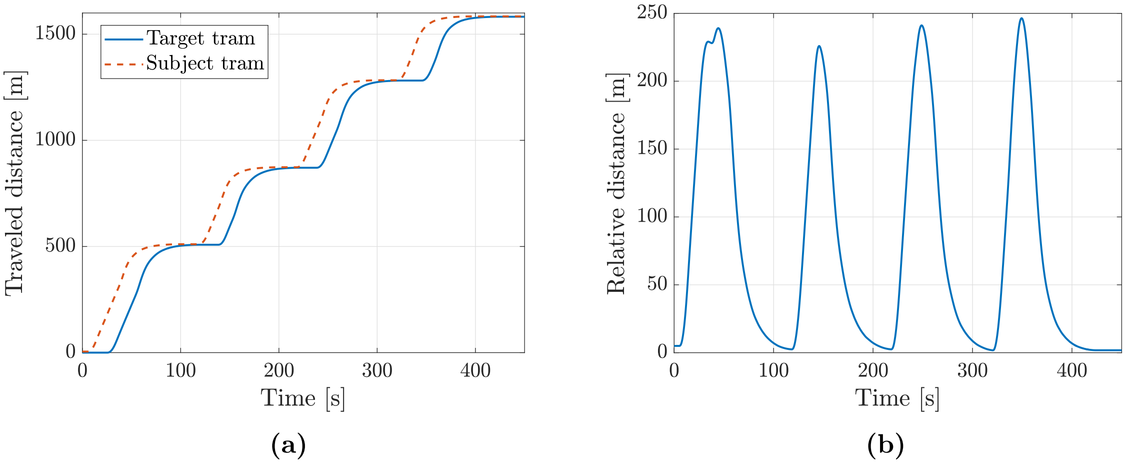


Figure 6.4: **(a)** Generated trajectories of subject and target tram. **(b)** Relative distance between the trams over time.

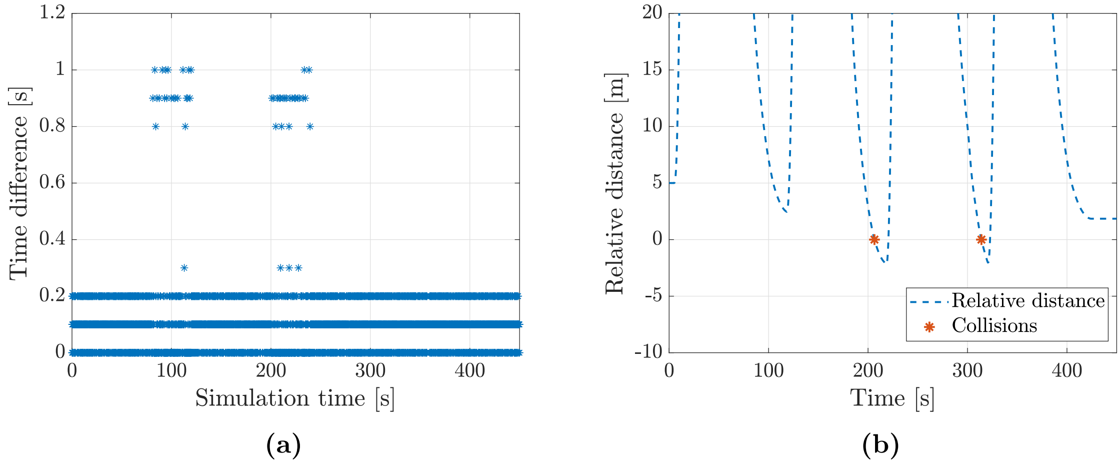


Figure 6.5: **(a)** Generated GPS outages for subject tram in simulation scenario. Red rectangle mark high-duration outage. **(b)** Generated trajectory with collisions of trams. Overshoot of relative distance in both collisions is $\approx -2\,\text{m}$.

Communication between the trams using CAM is modelled as described in Sec. 6.1. I assume constant communication delay $D = 250\,\text{ms}$. I do not include packet loss into the scenario.

6.3.2 Collision scenario

Scenario with collisions differs from the collision-free scenario only in the trajectory of subject tram. I set the trajectory of subject tram to collide with the trajectory of target tram. In particular, these collisions will be added while stopping at tram stop "Masarykovo nádraží" and "Náměstí Republiky", see Fig. 6.5b. Collision is represented by a negative overshoot in

		Average-based	Conservative
Clearance est.	Forward propagation	Eq. (6.2)	Eq. (6.3), $a_{\mathrm{targ}} = -1.74$
Braking dist. est.	t_{lag}	1.3	3.2032
	u	-0.8	-0.6517
	a_{subj}	-1.74	-0.57
Collision warn.	Critical function $g(\mathcal{Y}^N)$	Eq. (6.7)	Eq. (6.8)

Table 6.2: Summary of 'Average-based' and Conservative approaches to collision warning.

the relative distance (clearance). In both cases, the overshoot is ≈ $-2\,\mathrm{m}$.

6.4 Evaluation of designed collision avoidance method

In total, there are four experiments to be evaluated: two proposed design of collisions warning system (average-based and conservative) on two simulation scenarios (without and with collisions). Summary of the two approaches is given in Tab. 6.2. Every experiment is simulated 100 times with different seed of random numbers. I then statistically evaluate the results.

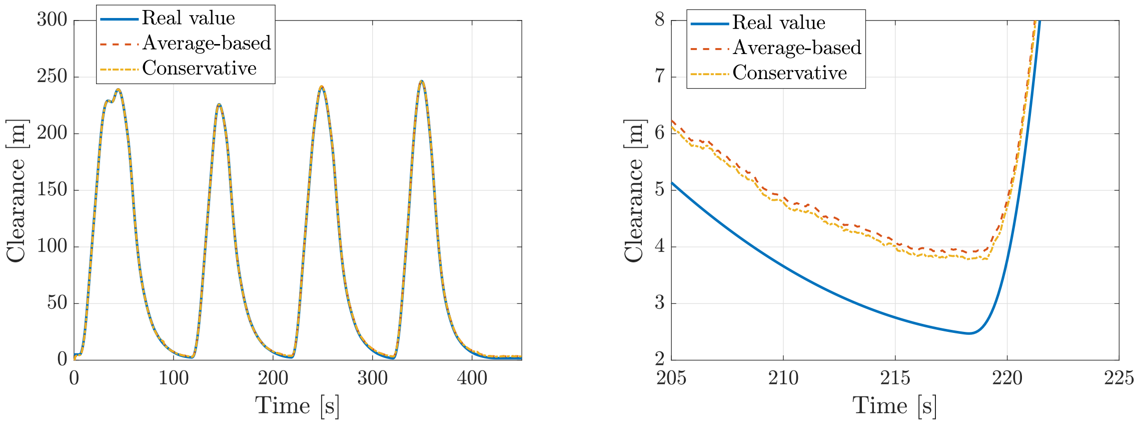
6.4.1 Clearance

I first compare the two approaches for clearance estimation. See Fig. 6.6a which displays real value of clearance $x_{c,k}$ (from simulation) and mean (from 100 runs with no collision) estimation of two approaches. Clearly, both approaches give similar results. RMSE of average-based approach is $e_{\mathrm{RMSE}}(\hat{x}_{c,\mathrm{avg}}) = 2.7608\,\mathrm{m}$ and of conservative approach is $e_{\mathrm{RMSE}}(\hat{x}_{c,\mathrm{cons}}) = 2.7233\,\mathrm{m}$. This error in the estimation of $x_{c,k}$ is given mainly by an error of absolute position estimation.

I assume that the reason why the two approaches give a similar result is that the conservative approach is an acceptable approximation of state-based forward propagation. In addition, the delay between two consecutive CAM is usually small and thus possible inaccuracy in the conservative approach is propagated only for a short time interval.

6.4.2 Braking distance

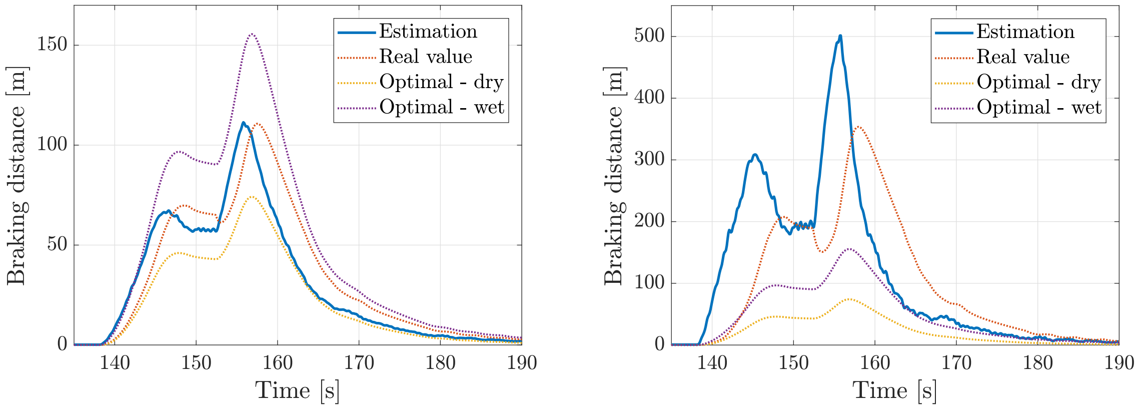
Let's now focus on the estimation of braking distance using the two proposed approaches. I compare the estimations with several other values. First is the real value obtained from the simulation using the same time lag t_{lag} and braking intensity u as in Eq. (6.4) for computation of d_{br}. Second values is optimal braking distance, that is with $t_{\mathrm{lag}} = 0$ and $u = -1$, assuming good adhesion conditions (dry railtrack). The third is also optimal braking distance but with worse adhesion conditions (wet railtrack). See the results in Fig. 6.7a for average based



(a) Whole run

(b) Closer look

Figure 6.6: Comparison of average-based and conservative approaches for clearance estimation with real value (from simulation).



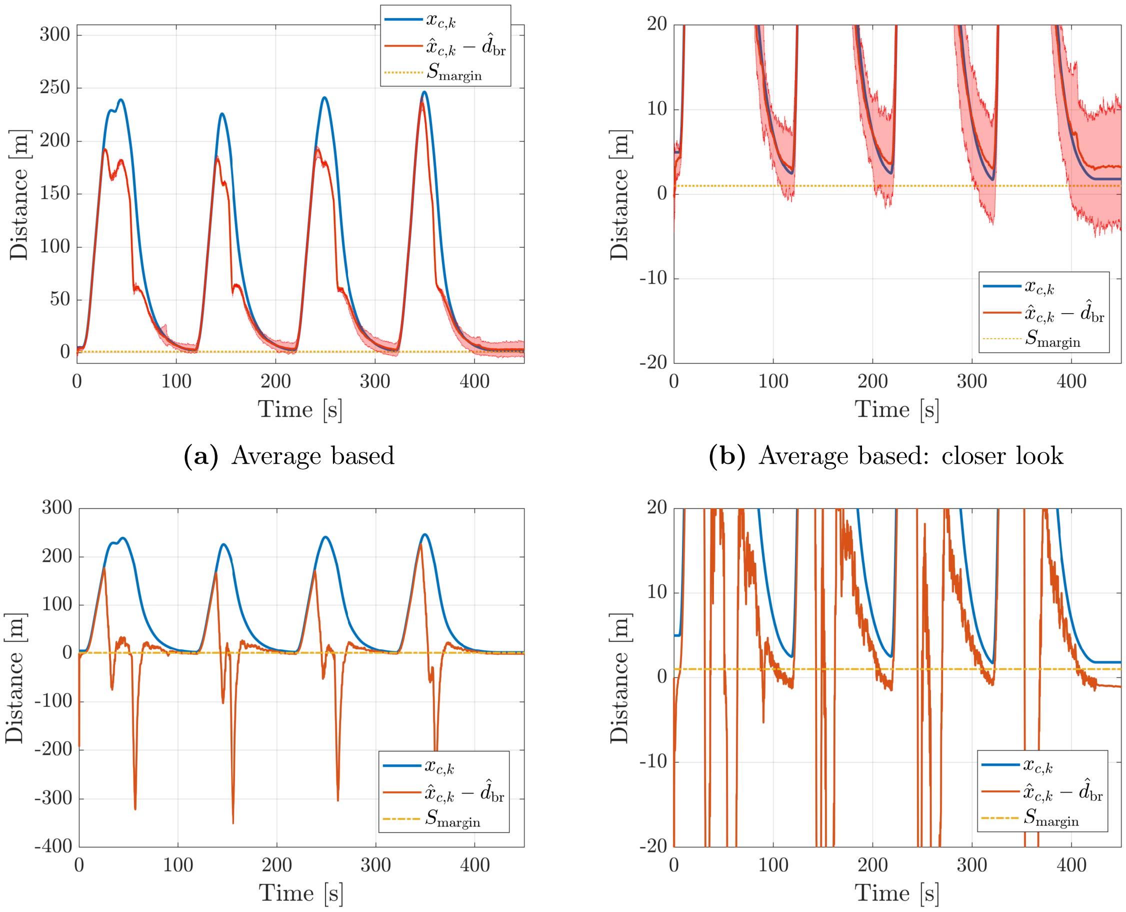
(a) Average based estimation

(b) Conservative estimation

Figure 6.7: Comparison of braking distance d_{br} calculated using Eq. (6.4) and braking distance obtained from simulations with different settings.

estimation and Fig. 6.7a for conservative estimation.

There are several things which can be deduced from the results. Clearly, the use of Eq. (6.4) for calculating braking distance lacks real braking dynamics, and therefore, the shape of an estimated d_{br} differs from the real value. One of the problems with the conservative approach is relatively high estimated value of b_{br} for high speed of a tram, which could cause false collision warnings. On the other hand, see that conservative estimation upper bounds both optimal values, whereas average-based estimation is lower than 'Optimal - wet'.



(a) Average based

(b) Average based: closer look

(c) Conservative

(d) Conservative: closer look

Figure 6.8: Collision warning for collision free scenario.

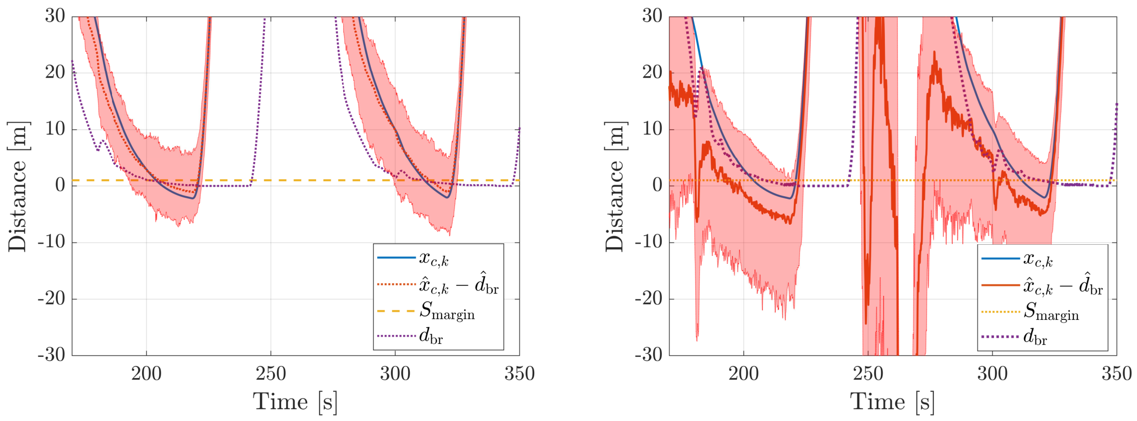
6.4.3 Collision warning

Combining the estimation of clearance $x_{c,k}$ and braking distance d_{br}, I can finally test and evaluate the collision warning algorithm. Results from all collision warning experiments are summarized in Subsec. 6.4.4.

Without collision

I first test the two approaches on collision free scenario to evaluate false warnings. See the results in Fig. 6.8 where in every figure, there is a real (simulated) value of clearance $x_{c,k}$, a mean of differences $\hat{x}_{c,k} - \hat{d}_{\text{br}}$ from all runs and margin S_{margin}. Based on the critical function $g(\mathcal{Y}^N)$, collision warning is triggered if $\hat{x}_{c,k} - \hat{d}_{\text{br}} \leq S_{\text{margin}}$.

Even though for average based approach, the mean value of $\hat{x}_{c,k} - \hat{d}_{\text{br}}$ is always above the S_{margin}, min/max bound (red area) from all runs shows that in some cases, estimated $\hat{x}_{c,k} - \hat{d}_{\text{br}}$ does cross the S_{margin}. From 100 simulation runs, collisions warning counts at



(a) Average based estimation

(b) Conservative estimation

Figure 6.9: Collision warning for scenario with collisions at second and third tram stop.

every tram stop are: $[36, 26, 33, 52]$. Highest false collision warning count is at the last tram stop. This is probably caused by GPS outage (see Fig. 5.3a) of target tram. False collision warning count at remaining tram stop is similar. Mean of false collision warnings count for all tram stops is ≈ 37.

Focusing now on the conservative approach, false collision warnings are not only right before tram stops but also when clearance is relatively high. This is caused mainly by high estimated braking distance in conservative approach as shown in Fig. 6.7b. There is also an influence of incorporating the variances according to Eq. (6.9), see Fig. A.2. These false warnings, however, could be filtered out by some heuristic, for instance using the value of clearance. The main issue here is false warnings before the tram stops. In this case, the mean of estimated $\hat{x}_{c,k} - \hat{d}_{\mathrm{br}}$ is crossing the S_{margin} at every tram stop. Among 100 simulations, there are following counts of false warnings: $[63, 56, 38, 68]$ with mean count ≈ 56. Note that lowest value of false warnings is at third tram stop which is probably thanks to the fact that neither of the trams has significant GPS outage.

With collisions

For scenario with collisions, results of the average based and conservative approaches are in Fig. 6.9a and Fig. 6.9b. Again, I display the real value of clearance $x_{c,k}$, mean estimation of $\hat{x}_{c,k} - \hat{d}_{\mathrm{br}}$ with min/max bounds and S_{margin}. In addition, I also display real braking distance d_{br}. There are two aspects which I evaluate. First is if given approach detects an imminent collision and second is if the collision warning is sufficiently in advance for a driver to prevent the collision. To test if collision warning given at time t_{warn} is sufficiently in advance, I compare the value of clearance $x_{c,k_{\mathrm{warn}}}$ with value of real braking distance d_{br}. Intuitively, if (and only if) $x_{c,k_{\mathrm{warn}}} > d_{\mathrm{br},k_{\mathrm{warn}}}$, collision warning is sufficiently in advance.

Tram stop	Without collision Average	Conservative	With collision Average	Conservative
1st	36	63	–	–
2nd	26	56	89/60	100/81
3rd	33	38	78/43	81/67
4th	52	68	–	–
Mean	36.75	56.25	51.5	74

Table 6.3: Summary of the approaches to collision warning. All numerical entries in the table represent number of collision warning at particular tram stop from 100 simulation runs. Entries in blue represent number of collision warnings provided sufficiently in advance. Some values in the table are intentionally left unfilled since such values are not relevant for evaluation.

Again from 100 simulation runs, the average based approach can detect collision at the second and third tram stop in 89 and 78 cases, respectively. Among these detections, 60 collision warnings at the second tram stop and 43 collision warnings at third tram stop are sufficiently in advance. One of the reasons why a relatively high number of warnings are given too late is probably that the estimated value of braking distance $\hat{d}_{\mathrm{br}}$ is usually lower than real value d_{br}.

Similarly, for the conservative approach, the number of collision detections at second and third tram stop are 100 and 81. Sufficiently in advance are 81 and 67 warnings. Although conservative approach gives more sufficiently in advance collision warnings, it has more false collision warnings.

6.4.4 Summary

To summarize the two approaches, I start with the clearance estimation. I propose to use conservative forward propagation (6.3) since it gives a better result (in terms of RMSE) than the state-based forward propagation. Also, this approach is more efficient since less information is sent through communication. Results from the simulations show that the approach to braking distance estimation heavily influences triggering of collision warnings. Furthermore, it is not straightforward to decide which off two approaches is better. The conservative approach gives a more false warnings, but on the other hand, it provides more correct collision warnings, see a summary of the simulations in Tab. 6.3. Nevertheless, in real application, I propose to use the average approach to both braking distance estimation and collision warning (6.7). The reason is that the collision warning system should only support a driver and it might be better to give collision warnings to at least a portion of imminent collisions than giving many false collision warning.

7 | Conclusion

In this chapter, I conclude the thesis by providing an overview of achieved results and proposing possible future improvements of the developed method for collision avoidance described in the thesis. The main goal of this thesis was to design an algorithm which would possibly run onboard a tram and give warning of imminent rear-end collision with a forward tram. I achieved this goal by developing an algorithm which utilizes V2V communication and estimation of tram's state of motion using Kalman filter. Main contributions of this thesis are:

- I analysed measurements from inertial sensors placed onboard a tram during its motion. The analysis revealed that measurements from the sensors contain relevant information of tram's motion but also that inertial measurement itself is not sufficient to estimate correctly the motion (Chapter 2).
- I examined available geographic information of tram tracks, which I referred to as the digital map. Then, I described the use of the Digital map for state estimation and model development. I explained its construction for a particular tram track, using data from Open street maps and digital elevation model (Chapter 3).
- I developed a high-fidelity model of the plant: a closed-loop system of a driver and a tram. The model allows motion simulation of (two) trams on a tram track described by the Digital map. The model generates data with acceptable accuracy close to real measurements (Chapter 4).
- I implemented a Kalman filter for tram longitudinal motion estimation exploiting sensory measurements and the Digital map (Chapter 5).
- I designed a collision warning system and extensively evaluated its performance using simulations. Besides, I proposed a configuration for a real application (Chapter 6).

7.1 Future work

Development of a collision warning system for trams continues within the collaboration with project partners. Since the proposed algorithm gave promising results on simulations, the main goal now is deployment and testing the collision warning system in practice. There are, of course, several problems imposed by a real environment which I neglected in the design and which need to be addressed.

Appendices

A | Supplementary figures

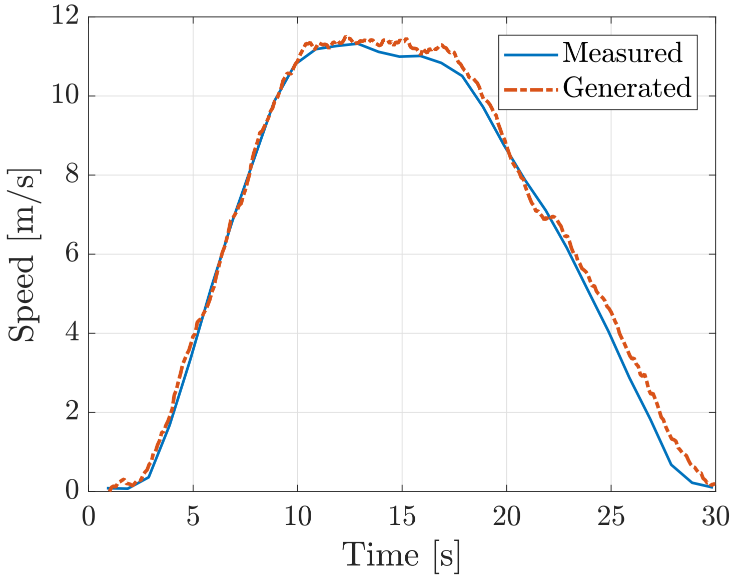


Figure A.1: Comparison of measured speed from experiment in Poruba depot and generated y_v.

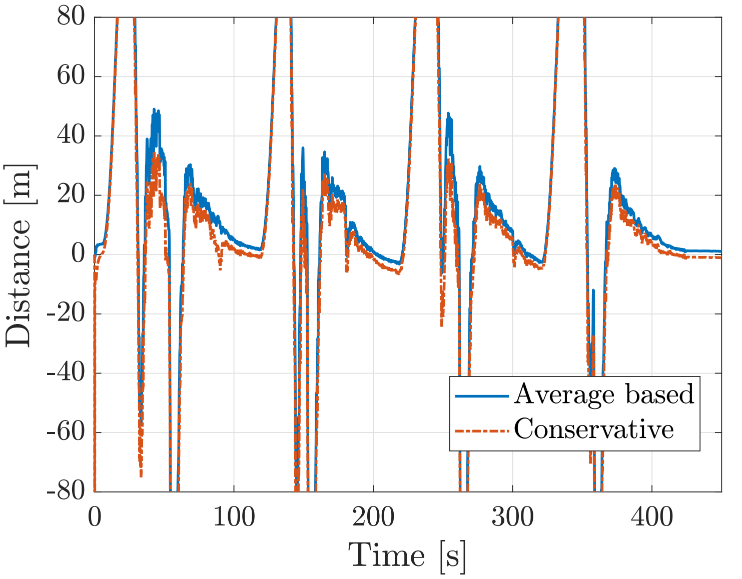


Figure A.2: Comparison of average based and conservative approach to computation of $x_{c,k} - d_{\mathrm{br}}$.

B | Poruba experiments

With our project partners, we realized set of experiments in depot Poruba in city of Ostrava, see the map of a track in Fig. B.1. Total length of the track is ≈ 1 km. Main experiments were done mainly on ≈ 300 m straight sections. All experiments were done with tram Tatra T3 with approximately eight people in the carriage. Adhesion conditions of the track were excellent according to depot employee (dry track without any significant soil). Here I list the main experiments which we did:

- Regular driving.
- Speed up with maximal acceleration and slow down (to stop) with maximal deceleration using only service brake.
- Series of experiments with speed up to given reference speed, keeping the referenced speed for few seconds and then slow down to stop.
- Emergency braking from given speed.

All experiments were acquired by two sensor platforms: low-cost and precise 2.1. Dataset from all experiments are on DVD attached to this thesis, stored in JSON format and with MATLAB® file for a visualization of the data.

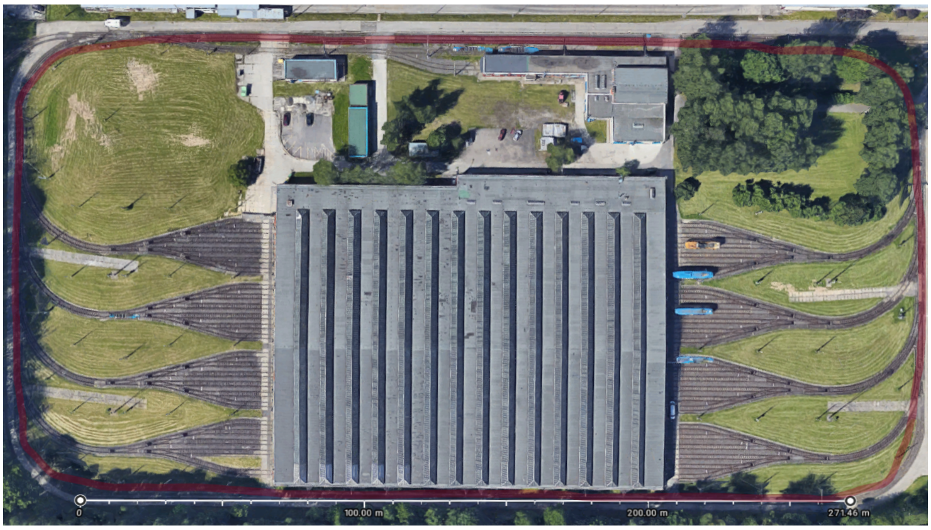


Figure B.1: Map taken from maps.google.com which shows the depot Poruba in city of Ostrava

C | LMD equations

Equations describing longitudinal motion dynamics:

$$
\begin{aligned}
\omega_{\mathrm{wh}} &= \frac{1}{J_{\mathrm{wh}}} \int \left(T_{\mathrm{mot}}(t) - rF_{\mathrm{ad}}(t)\right) dt \;, \\
\dot{s}(t) &= \int \frac{1}{M(t)} \left(F_{\mathrm{ad}}(t) - F_{\mathrm{res}}(t)\right) dt \;, \\
F_{\mathrm{ad}}(t) &= \mu(v_S) F_N(t) \;, \\
v_S(t) &= v_{\mathrm{wh}}(t) - \dot{s}(t) \;, \\
\mu(v_S) &= \frac{2K_S \mu_{\max}^2 v_S(t)}{\mu_{\max}^2 v_S^2(t) + K_S^2} \;, \\
F_{\mathrm{res}}(t) &= 1000 F_N(t) \left(0.1\frac{\theta\pi}{180} + 2.5 + \frac{(3.6\dot{s})^2(t)}{850}\right) \;, \\
F_N &= M(t) g \;.
\end{aligned}
\tag{C.1}
$$

Physical meaning of constants is described in Subsec. 4.2.3. This model is based on the model presented in

References

[1] An N., Mittag J. and H. Hartenstein, "Designing fail-safe and traffic efficient 802.11p-based rear-end collision avoidance," in 2014 IEEE Vehicular Networking Conference (VNC), 2014, pp. 9–16.

[2] Anderson B. and Moore J., "Optimal Filtering", Prentice-Hall, Englewood Cliffs, New Jersey, 1979.

[3] Antsaklis P. J. and Michel A. N., "A Linear Systems Primer", 2007 edition, Birkhauser Boston, 2007.

[4] Biswas S., Tatchikou R. and F. Dion, "Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety," IEEE Communications Magazine, vol. 44, no. 1, pp. 74–82, Jan. 2006.

[5] Brown F. T., "Engineering System Dynamics. A Unified Graph-Centered Approach", Second Edition, CRC Press, 2006.

[6] Brunson S.J. et al., "Alert Algorithm Development Program. NHTSA Rear-End Collision Alert Algorithm. Final Report", Technical Report DOT HS 809 526 U.S. Department of Transportation National Highway Traffic Safety Administration, September 2002.

[7] Candy J. V., "Bayesian Signal Processing: Classical, Modern and Particle Filtering Methods", New Jersey: John Wiley and Sons, Inc., 2009.

[8] Edstrand J., "Calculation method for powering a tramway network", Master's thesis, Chalmers university of technology, 2012.

[9] Franklin F. G., Powell J.D. and Emami-Naeini, Abbas. "Feedback Control of Dynamic Systems", ISBN 978-1-29-206890-6, NY, Pearson, Eighth edition, 2018.

[10] Heirich, O., A. Lehner, P. Robertson, and T. Strang, "Measurement and Analysis of Train Motion and Railway Track Characteristics with Inertial Sensors.", in Proc. 14th Int. IEEE Conference on Intelligent Transportation Systems, 2011, pp. 1995–2000.

[11] Intelligent Transport Systems (ITS), Vehicular communications, Basic set of applications, Part 2: Specification of cooperative awareness basic service, 2014.

[12] ISO 6709:2008. "Standard representation of geographic point location by coordinates."

[13] ISO 15623:2013. "Intelligent transport systems – Forward vehicle collision warning systems – Performance requirements and test procedures."

[14] Karnopp D. C., Margolis D. L., Rosenberg R. C., "System Dynamics: Modeling and Simulation of Mechatronic Systems.", ISBN: 978-0-470-88908-4, Wiley; Fifth edition, 2012.

[15] Mukhtar, A., L. Xia, and T. B. Tang., "Vehicle Detection Techniques for Collision Avoidance Systems: A Review.", IEEE Transactions on Intelligent Transportation Systems 16, no. 5 (October 2015): 2318–38. https://doi.org/10.1109/TITS.2015.2409109.

[16] Nilsson J., Real-Time Control Systems with Delays, Doctoral thesis, Department of Automatic Control, Lund Institute of Technology, 1998

[17] Pichlík P., "Strategy of Railway Traction Vehicles Wheel Slip Control", Doctoral Thesis, Czech technical university in Prague, 2018.

[18] Reid I., "Estimation II, Lecture notes", Available online: http://www.robots.ox.ac.uk:5000/ĩan/Teaching/Estimation/LectureNotes2.pdf, Hilary Term, 2001.

[19] Sadr S., Khaburi D. A. and Rodríguez J., "Predictive Slip Control for Electrical Trains", IEEE Transactions on Industrial Electronics, vol. 63, no. 6, pp. 3446–3457, Jun. 2016.

[20] Simon, D. "Optimal State Estimation, Kalman, H_∞ and Nonlinear Approaches." John Wiley & Sons, Inc.m Hoboken, New Jersey, 2006.

[21] Titterton D.H. and Weston, J.L. "Strapdown Inertial Navigation Technology." 2nd Edition, AIAA Education Series, Reston, 2004. https://dx.doi.org/10.1049/PBRA017E

[22] Willke T. L., Tientrakool P. and Maxemchuk N. F., "A survey of inter-vehicle communication protocols and their applications", IEEE Communications Surveys Tutorials, vol. 11, no. 2, pp. 3–20, Second 2009.

[23] Xiang X., Qin W. and Xiang B., "Research on a DSRC-Based Rear-End Collision Warning Model", "IEEE Transactions on Intelligent Transportation Systems", vol. 15, no. 3, pp. 1054–1065, Jun. 2014.