A predictive-control framework to eliminate bus bunching

Master's Thesis in Technomathematics

Author:
Matthias ANDRES

Supervisors:
Prof. Dr. René PINNAU
Dr. Rahul NAIR

September 2016

Dedication

To my parents Helga and Johannes Andres

Declaration

I hereby declare that I composed this thesis on my own and did not use any sources or aids other than those mentioned. The contents of this thesis are original, except where specific reference is made, and have not been submitted in whole or in part for consideration in any other examination procedure.

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die übernommenen Stellen wurden als solche kenntlich gemacht. Ich erkäre weiterhin, dass die vorliegende Arbeit noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Matthias Andres
September 2016

Acknowledgements

I would like to thank Prof. Dr. René Pinnau for his great support and commitment during this master's thesis.

Furthermore I would like to thank Dr. Rahul Nair for giving me the great opportunity to work with him in IBM's Smarter Cities Research Laboratory in Dublin and introducing me to the topic of bus bunching as well for the support on the way.

Abstract

Buses not arriving on time and then arriving all at once - this phenomenon is known from busy bus routes and is called *bus bunching*.

This thesis combines the well studied but so far separate areas of bus-bunching prediction and dynamic *holding* strategies, which allow to modulate buses' dwell times at stops to eliminate bus bunching. We look at real data of the *Dublin Bus* route *46A* and present a headway-based predictive-control framework considering all components like data acquisition, prediction and control strategies. We formulate time headways as *time series* and compare several prediction methods for those. Furthermore we present an analytical model of an artificial bus route and discuss stability properties and dynamic holding strategies using both data available at the time and predicted headway data. In a numerical simulation we illustrate the advantages of the presented predictive-control framework compared to the classical approaches which only use directly available data.

Contents

1. Introduction

1.1 Overview

Nowadays many cities suffer from huge traffic density and congestion. One attempt to the solution of this problem is to motivate the city residents to use public transportation systems instead of private cars. To this many bus service providers such as *Dublin Bus* operate bus routes with a high frequency to ensure high mobility within the city area. One problem that comes along with highly frequent bus routes is called *bus bunching*, also known as *platooning*, which refers to the phenomenon of two buses of the same route arriving at the same time at a bus stop, like shown in figure 1.1. One of the main reasons for bus bunching is varying passenger-boarding times and can be illustrated by a vicious cycle like in figure 1.2. If a bus arrives delayed at a stop and the previous bus has departed from that stop on time there will be more passengers waiting. This increases the overall boarding time at this stop and further delays the arrival time of this bus at the next stop.

Bus bunching has negative effects as well for the passengers as for the service providers. Like illustrated in figure 1.3 bunching leads to larger headways between separate bunches of buses. This leads on the one hand to longer waiting times for the passengers at the stops downstream and on the other hand to longer travel times for the passengers inside the bus due to longer boarding and dwell times at the stops. The disadvantage for the service providers is that the costs are those of a highly frequent bus route, i.e., they need to provide many vehicles and drivers, but the real frequency experienced by the passengers is much smaller due to the larger headways between the separate bunches, which results in a low cost efficiency.

Figure 1.1: Bus bunching at the *General Post Office* in *Dublin*, route *46A* [1]

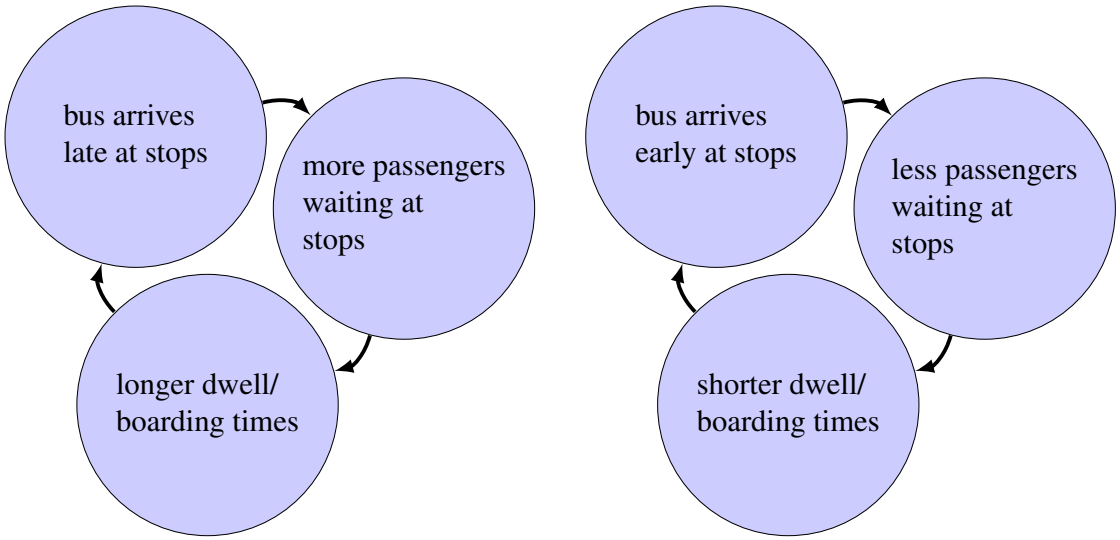


Figure 1.2: Bus bunching as vicious cycle

[1] photo taken by: Matthias Andres, 19th Apr 2016

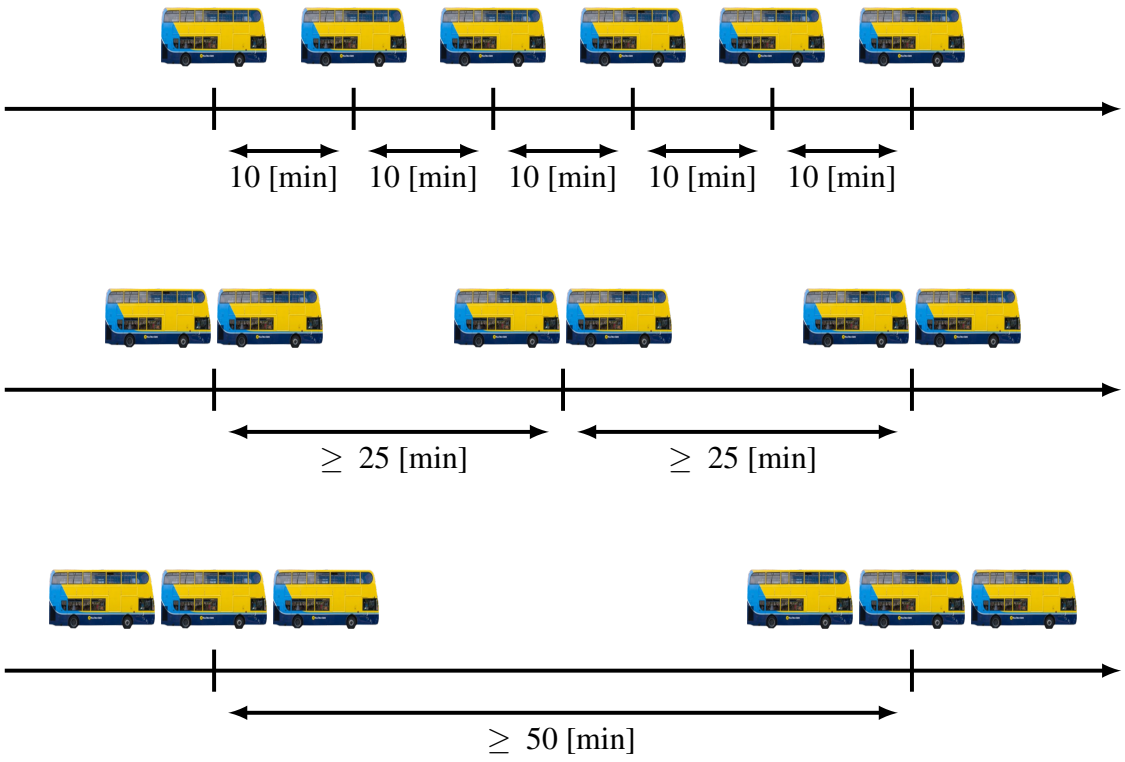


Figure 1.3: Negative effects of bus bunching [2]

The field of bus bunching has been studied for more than 50 years but got new attention recently due to modern GPS based techniques that provide online data on the bus positions. The two main areas of research based on the newly created data focus on the prediction of bunching events and on corrective control strategies.

In this thesis we present a predictive-control framework based on real data of the *Dublin Bus* route *46A, outbound,* that combines these two main areas. The prediction of future headway values based on real data is done, to our best knowledge, for the first time not by arrival time prediction but by interpreting continuous-time headways as time series and predicting these directly. Furthermore we extend the existing stability analysis of a headway-based dynamic holding strategy using data available at the time such that we are able to include predicted headway data. Where earlier work, to our best knowledge, only considered single components of a predictive-control strategy regarding bus bunching, we consider all components.

This thesis is structured like the predictive-control framework shown in figure 1.4 and is organized as follows. In the following section we give a literature review on topics related to bus bunching. In chapter 2 we discuss some preliminary topics like the concepts

[2] photo of the Dublin Bus vehicle taken from: `http://www.thejournal.ie/dubiin-bus-strike-2919690-Aug2016`

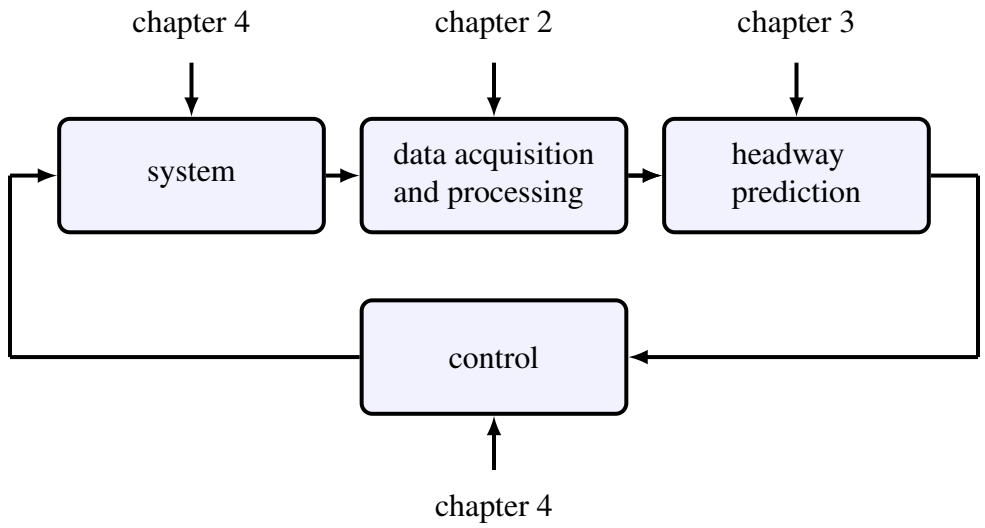


Figure 1.4: Predictive-control framework and organization of this thesis

of *space-time diagrams* and *continuous-time headways*, and the acquisition of the real data of the *Dublin Bus* route *46A, outbound*. In chapter 3 we discuss preprocessing steps of the headway trajectories and we formulate the data-driven prediction of headway values in the context of time-series forecasting. We compare the predictive performances of *linear-regression-and-extrapolation-, kernel-regression-and-extrapolation-, artificial-neural-network-* and *autoregressive-models* applied to the real data. In chapter 4 we present an analytical model of an artificial bus route. We discuss stability properties and present a classical headway-based control strategy using directly available data as well as a headway-based predictive-control strategy using predicted headway values. In a numerical simulation we show the advantage of our suggested predictive-control scheme compared to the classical one. In chapter 5 we summarize our findings and refer to starting points for future work.

1.2 Literature Review

In the following section we give an overview on some of the existing literature regarding bus bunching. We organize the literature review with respect to the two main topics of this thesis: prediction and control.

1.2.1 Prediction

As described in chapter 2 there are two main tools to investigate the bus bunching phenomenon. One is the *space-time diagram* and the other is the *time-headway* distribution, which we abbreviate simply by *headway* distribution, unless stated otherwise.

An intuitive approach to predict bus bunching events is based on the space-time diagram: if we would be able to predict future trajectories of two neighboring buses, we could use the space-time diagram to check if the trajectories intersect or become close w.r.t. some distance metric and use this information to predict bunching events. This allows us to use methods out of the widely explored field of *bus arrival time estimation*. Since this is the most common tool to predict bunching events and there is just little work explicitly focusing on bunching prediction most of the listed literature is related to bus arrival time estimation. The literature in this field differs basically in two points: the used data and the used prediction algorithms.

The headway distribution is a second tool to explore the bunching phenomenon and was used, for instance, in [Mor+14] to detect bus bunching events. Though to our best knowledge there has been no work so far where the authors explicitly predicted the headway trajectories using *Automated Vehicle Location (AVL)* data only using headway information and without predicting the bus arrival times at stops as an intermediate step.

Among the most common ways to gather data of bus arrival times are on-board or on-station surveys ([Yu+10]), evaluation of smartcard transactions ([Zho+16]), use of mobile phone data ([ZZL12]), road side terminals or toll stations ([CK03],[YLT11]), and most recently automated systems like AVL and *Automated Passenger Counting (APC)*. An overview on the AVL and APC technology is given in [Fur+03]. An intuitive way to gather *Global Positioning System (GPS)* data from buses that is easy to implement is to install a smartphone in each bus, which was explored in [Bia+11].

A first general framework for the prediction of *transit vehicle arrival and departure times* based on AVL data has been presented by Cathey in 2003 ([CD03]), using a tracker, filter and predictor element. In [Pad+10] we find a list of different types of methods for the prediction that is similar to the following: *schedule-based methods, statistical methods, machine-learning methods* and *model-based methods*.

The first type of methods are *schedule-based methods*. The information contained in the schedule is combined with historical data of the bus trajectories to estimate travel times on route segments, which can then be used in a cumulative way to predict arrival times. In [MK04] and [LZ99] the estimated travel times were combined with real-time data of the bus positions to predict the arrival times of the buses. Furthermore they introduced an *Intelligent Transportation System (ITS)*, which consists of the three modules *automatic bus location system, real time traffic monitoring system* and *passengers information system*.

The next approach uses *statistical methods* like *time-series analysis*, where nonlinear time-series models were used for the prediction of corridor travel times ([DAW99],[IA02]).

The prediction of this can then be exploited in a second step to predict the bus arrival times. Further statistical methods to predict travel times are *regression models*, like a Bayesian-regression model ([FK98]), or a (multivariate) linear-regression model which combines the current traffic situation with historical information, based on both data gathered by a single loop detector ([RV04]) and APC-data ([PCB04]). Another very common technique is the *nearest neighbor (NN)* method using AVL data, like in [Cha+10] and [CPC11]. The latter work mentioned also used data from *Dublin Bus* like in this thesis. This approach determines the historical trajectory that is 'most similar' to the current partial trajectory, and the corresponding trajectory is then used as forecast. Crucial for the performance of this kind of algorithms are the used distance metric and similarity measures ([TJ08]).

The next idea is to employ *machine-learning techniques*, like *artificial neural networks (ANNs)* in [JR04]. The great advantage of ANNs is the ability to include *exogenous inputs*, which are quantities that are used for the prediction of other quantities than themselves, like schedule adherence, dwell times and traffic congestion, additionally to the real-time information on the bus positions in their model ([Jeo05],[JR05]). Another approach of this field is the use of *support vector machines (SVM)* to predict bus arrival times based on AVL data ([BZB06]), which also has been tested on the real transit system of the city San Antonio, Texas ([VR07]). A technique that could be seen as both statistical method and machine-learning technique is the *kernel regression*, which was successfully applied to predict future trajectories and thus future arrival and departure times on a dataset from *Dublin Bus* ([Sin+12],[LB15]). The same kernel-regression approach was used in [Nai+14] to predict travel times, but this time as an intermediate step to explicitly predict bus bunching events. Other works which explicitly focused on bunching prediction by considering both historical and real-time data to predict travel times and thus headway distributions are [Mor+14] and [Mor+16], where *random forests* were applied for the offline regression.

The next approach is mentioned as *model-based approach*, which most likely makes use of a mathematical description of the transit system or a part of it and then uses the *Kalman filter* to predict the future states, which could be modeled, for instance, as bus positions or arrival times, and to fuse different information. This was done, for example, in [SF04], where a combination of two different Kalman filters was presented, one for the prediction of running times and the other for the prediction of dwell times, based on APC and AVL data. The authors in [Son+04] also used a Kalman filter and AVL data for the prediction of stop-to-stop travel times, but explicitly taking the waiting times at signalized intersections into account. Another model-based technique was presented in [Han+15], where a stochastic bus operation model was used within a *particle filter*. Further model-based approaches were based on *Markov chains* ([LB04]), statistical *survival models*

([GYW16]), or *finite state automata*. The latter concept was used in [Sun+07] to combine real-time information of the bus locations given by AVL data and historical average speeds on route segments in order to take variations of traffic conditions into account.

A comparison of a linear regression, a k-nearest-neighbor algorithm, ANN and SVM approach, all based on *Automated Vehicle Identification (AVI)* data, can be found in [YLT11]. Several authors also combined different techniques mentioned above. For example, in [Yu+10] a combination of a SVM and a Kalman filter was presented. In here the SVM was employed to predict segment travel times based on exogenous inputs like weather data and the daytime while the Kalman filter fused these information with the latest bus arrival information to predict the bus arrival times. Similar to this a combination of an ANN and a Kalman filter was exploited in [Che+04] to predict bus travel and arrival times based on APC data and latest bus-arrival information, where the use of an ANN again allows the consideration of exogenous inputs like the daytime. It is often promising to use *ensembles* of different predictors instead of restricting just to one single prediction method. Thus an ensemble of several ANNs for the prediction of bus arrival times was presented in [HVV09], where also confidence intervals for the predictions were stated.

Some works like [Pad+10] distinguished explicitly between homogeneous and heterogeneous traffic conditions in their prediction model. Most of the models presented in the literature are based on simplified models of the bus operation system. A more detailed analysis of this can be found, for example, in [MQ13], where a probabilistic model for the dwell times and the impact of merging into the bus lane were explored. Nevertheless in [Mil08] it has been stated that the most significant contributing factor to dwell times of buses is the process of boarding and alighting passengers.

1.2.2 Bunching and Corrective Strategies

The bus bunching phenomenon was first studied in year 1964 by Newell and Potts ([NP64]). Based on a simplified model the authors proved the instability of a bus route, i.e., the tendency of buses to pair together. After stating the bunching problem a lot of research focused on corrective strategies to reduce bus bunching. This could be grouped into two fields: *planning* and *operational* strategies.

Planning strategies usually rely on a *schedule-based control* and try to reduce bus bunching by adjusting the timetables of the buses. In [Mor+12] historical AVL data were exploited to identify patterns of headway events and to detect critical schedule points, called *Bunching Black Spots*. A similar idea was proposed in [FF11b], where the authors statistically analyzed causes of bunching using AVL and APC data. This acquired

information could be used in two ways: either the service providers could change the locations and amount of bus stops, or they could change the arrival and departure times at the bus stops by including *slack times* in the schedule, which is one of the main tools in the planning context. Slack times used as a schedule-based (static) holding, which rule that the bus should not leave before its scheduled departure time even if the boarding process is already finished, have two disadvantages. On the one hand it was stated in [New77] that the stability of the bus route cannot be guaranteed only including a certain amount of slack time in the schedule in combination with a static holding strategy, for example, if the occurring delays are larger than the remaining slack times. On the other hand we need to consider a tradeoff between service reliability (do the buses bunch? are the buses on time?) and service quality (providing high frequency of buses, riding with maximal commercial speed and thus decrease the scheduled overall travel time) like mentioned in [Pil09]. Including scheduled slack times means that we increase the scheduled overall travel time of a bus in an undisturbed system which could be seen as decreasing the service quality. Nevertheless a simplified analytical model of a bus route considering one loop and a single checkpoint to address optimal (w.r.t. passengers' waiting time) slack time in schedule-based control can be found in [ZDB06].

The operational case has become more and more interesting with the availability of real-time data provided, for instance, by AVL and APC systems. Before this most of the control methods used decision models which only include bus arrival times at stops. The works done in the real-time setting differ in the used techniques, the used data and also in the goal they want to achieve by the control strategy.

Delgado ([Del+09]) divided the control strategies in the operational setting in the three categories *station control*, like *static holding* or *schedule-based holding*, *dynamic holding*, *boarding /dwell time limitation* and *stop skipping* (e.g., *deadheading, expressing* or *short-turns*), *interstation control*, like *cruising speed control*, *bus overtaking* or *transit signal priority* mechanisms, and other control measures like adding vehicles to the route.

The headway distribution can be seen as an indicator for service quality and reliability. In [LR09] several headway regularity metrics to measure bus reliability were given, and in [BG10] a stochastic model to identify headway distributions and their characteristics was presented. Furthermore it is very common to regularize the headways but also to minimize a certain objective function related to passenger waiting times.

As mentioned by Eberlein in [EWB01] holding seems to be a control strategy that frustrates passengers less than other techniques like stop skipping. One of the main assumptions of the dynamic holding strategy is that passengers do not adhere any schedule as far as the

bus route is operated with a certain minimal frequency. This frequency was found to have a maximal target headway between eleven and thirteen minutes ([FM09],[MC84],[JH75]) and was determined by analyzing the arrival times of new passengers at a bus station. It has been observed that the passenger arrival times tend to be uniformly distributed as far as the bus arrival frequency is less than the suggested threshold frequencies, and it has been concluded that passengers do not adhere the schedule in this case. In [FF11a] the authors investigated the major causes for bus bunching and recommended to move from a schedule-based (static) control to a headway-based (dynamic) control. An analysis based on AVL and APC data on how bus bunching, for one bus route as well as for several routes serving common bus stops, affects bus operations like dwell and running times can be found in [VDE16].

One of the first works on holding strategies was done by Newell in [New74], where the purpose was not the application to a real system but showing qualitative effects of holding strategies. The first holding strategy considering real-time AVL data was presented by Eberlein in [EWB01], where data were collected from a rail system and the total passenger waiting times were minimized within a quadratic programming context. The idea of exploiting AVL data has been applied to urban bus transit operations in [ZAJ04], where a random local search technique was used for the optimization. The first one who has given a mathematical stability analysis of the holding problem based on real-time AVL data was Daganzo ([Dag09]). He introduced an adaptive holding strategy based on AVL data and showed that the headway deviations are bounded using the proposed control scheme and assuming bounded noise terms. A similar control approach that is *self-equalizing* considering changes of the target headway, like after a bus breakdown, was developed in [BE12].

Another advantage of dynamic holding strategies over conventional schedule-based holding strategies is that these might require up to 40% less slack time in the schedule ([XAD11]), which increases the scheduled commercial speed of the bus route.

As mentioned above it is very common to set the regularization of the headways as the goal of a control strategy. Another very common approach is to state the control problem as an optimization problem, where the commands of the control strategy are taken as an input and one tries to minimize/maximize a certain objective function. Common objective functions are, for example, overall passenger waiting times, impact on passengers by the control, commercial speed, headway regularity etc.. One of the first works in this field was [ON72], where a bus route with one control point was considered and the optimal holding time w.r.t. average passenger waiting times was found via a dynamic

programming formulation. Similar approaches can be found in [IK74] and [Bar74]. In the latter one a holding strategy was presented that is optimal w.r.t. the average passenger waiting times and delay for on-board passengers, using statistical information on future arrival times. In [XAD11] a family of holding strategies to maximize the commercial speed was evaluated, whereas AVL data and future information about certain headway data were needed. Hickman provided an analytical model for the vehicle holding problem in [Hic01] and explicitly considered stochastic service elements like passenger demand, and formulated the holding problem for a single vehicle as a one-dimensional convex quadratic program.

In [Del+09] Delgado explicitly considered capacity constraints of the buses and presented a holding strategy combined with boarding limits, which is then used in a quadratic program to minimize the total times experienced by all passengers in the system. He presented a similar approach in [DMG12], but in here the objective function is the sum of at-stop waiting times per passenger, in-vehicle waiting times per passenger, extra waiting times of passengers who are prevented from boarding and a penalty term for passengers left behind in case there would be capacity available. Another work that explicitly considered capacity constraints is [Jia+03], where the authors used a *cellular automaton model*.

Little work has been done in our field of predictive control. In [Cor+10] a combined holding and stop-skipping strategy and the computation of the optimal control input w.r.t. passenger waiting times and impact on the passengers over a certain future horizon were presented as a *multi-objective dynamic program*, which was solved via a *genetic algorithm*. Another predictive-control approach was presented in [Sáe+12], where the prediction was based on an offline statistical model including AVL and APC data. As stated in those works the dynamic formulation of the system required the passenger demand forecast, based on offline and online data. Recent works in the field of predictive control have been [SKW16], in which a mathematical model for the holding problem was formulated that takes dynamic changes in running and dwell times into account, and [Mor+16], where the prediction of the headway distribution is used to determine whether the next control action should be holding or stop-skipping.

Several works were not phrased within a predictive-control frame, but also relied on information about future bus arrival times ([XAD11],[He15]). In [BWL15] the holding problem was formulated as a *stochastic decision process*, also including information about future arrival times.

Beside holding strategies there are several other methods to regularize the headway distribution and avoid bus bunching. In [Pil09] and [DP11] a control algorithm based on a

bus-to-bus cooperation and AVL data, that adjusts the cruising speed, was given. The work mentioned first also explicitly considered the tradeoff between high scheduled commercial speeds and service reliability. A combination of holding and adjusting the cruising speed is presented in [He15], such that the only information needed are arrival times of the current bus at the current stop and of the preceding bus at the current and next stop, which might improve reliability of the control since the only value that might be unknown and needs to be estimated is the latter one.

While most of the research focuses on the control of single bus routes, the authors in [Her+15] considered multiple bus routes serving common bus stops and analyzed the difference between independent operators and one central operator. Also in [Sch+16] bus bunching was investigated looking at multiple bus routes with common stops, but in here the goal was to inspect the effect of the fact whether overtaking at stops is allowed or not.

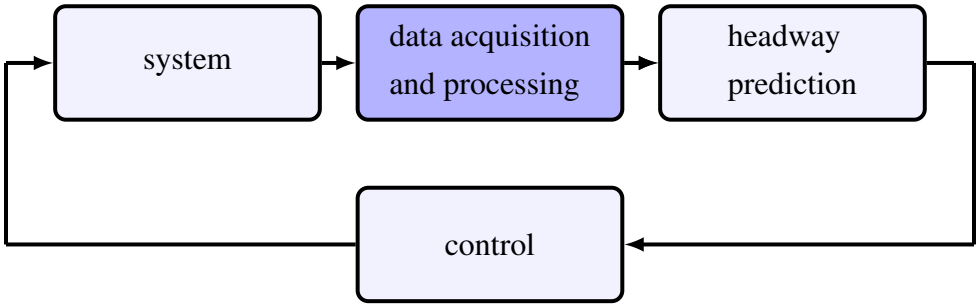
Some headway-based control strategies assume that it is both possible to decrease and increase the commercial velocity of buses. Under the assumption, that buses ride with maximal cruising speed, additional mechanisms are necessary to create the flexibility to increase the commercial velocity. Beside including slack times in the schedule in combination with dynamic holding strategies, which decreases the scheduled commercial velocity of the buses, and techniques like stop skipping ([SH05]), which irritate and frustrate passengers, one option would be *Transit Signal Priority* ([AF12]). Another mechanism to control bunching is to work with *minimum* and *maximum station waiting time*. This was considered in addition to a social aspect, namely passenger behavior, in [GP09], where also technical suggestions for engineers and social suggestions for passengers were given.

There are several works which explored the performance of control strategies within a field study. One of those is [Liz+14], where a real-time system was implemented in two bus services in Santiago de Chile and shows encouraging results.

The literature presented so far does either focus on the prediction part or on the control part related to bus bunching. To our best knowledge there does not exist any work that presents a predictive-control framework and considers all related components like data acquisition, prediction and control strategies. We found that considering the first two components are substantial for every predictive-control method, since without discussing the quality of the data and the reliability of the prediction it is hard to evaluate the feasibility and performance of a predictive-control strategy, unless this is implemented in a field study.

2. Preliminaries

In this chapter we first introduce a few basic concepts needed for the analysis of bus bunching, like the *space-time diagram* and *time headways*. Secondly we focus on the real data of the *Dublin Bus* route *46A, outbound,* and discuss the data structure and preprocessing steps.



2.1 Space-Time Diagram

In order to analyze the bus bunching problem we transform the spatial information like positions of stops or buses to a one-dimensional representation called *offset distances*, which describe the distance a bus traveled along its scheduled path, also called *shape*, from the first stop and which is explained in detail in subsection 2.1.1. If we transform a timestamped trajectory of bus positions, either given as coordinates of a two-dimensional map representation or as coordinates of a three-dimensional representation of the earth's surface, to the one-dimensional offset representation we obtain the *space-time trajectory*, and the visualization of space-time trajectories is called *space-time diagram*. Figure 2.2

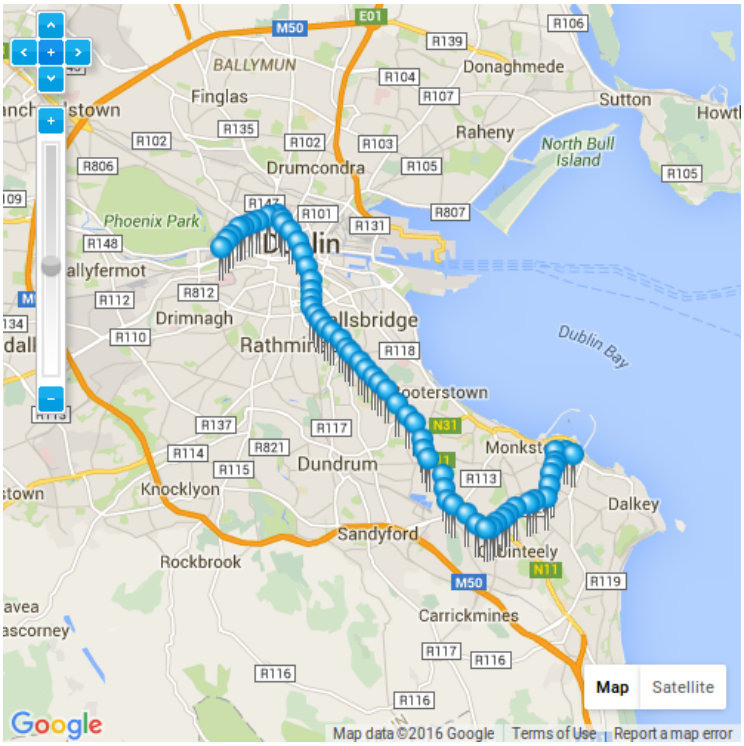


Figure 2.1: Route *46A, outbound,* on *Google Maps* [1]

shows the space-time diagram of bus route *46A, outbound,* of the 6th of November 2012. We state some general properties of the space-time diagram:

- The slope of a space-time trajectory indicates the *cruising speed* of the corresponding bus. Steeper lines indicate higher cruising speeds, flatter lines indicate lower cruising speeds.
- A bunching event corresponds to the intersection of two or more lines in the space-time diagram, or in a weakened form to a 'small horizontal distance' between two trajectories at some offset distance. Thus as mentioned in chapter 1 the space-time diagram is one option to detect, analyze and predict bunching events.
- The horizontal distance between two consecutive lines $n-1$ and n at a particular offset distance d is called *time headway* $h_{n,d}$ (see definition 2.2.1). This quantity is of major interest for this thesis and is further described in section 2.2. The vertical distance between two consecutive lines at a particular time is called *space headway*. Due to traffic conditions and varying passenger demands the time headway between two buses varies during the journey. A zero headway implies that the two related space-time trajectories intersect at the corresponding point and vice versa. Thus time headways are another tool to investigate bus bunching and in chapter 3 we address the prediction of those. In the following we refer with the term 'headway' to the

[1] source: `https://www.dublinbus.ie/RTPI/Sources-of-Real-Time-Information/`

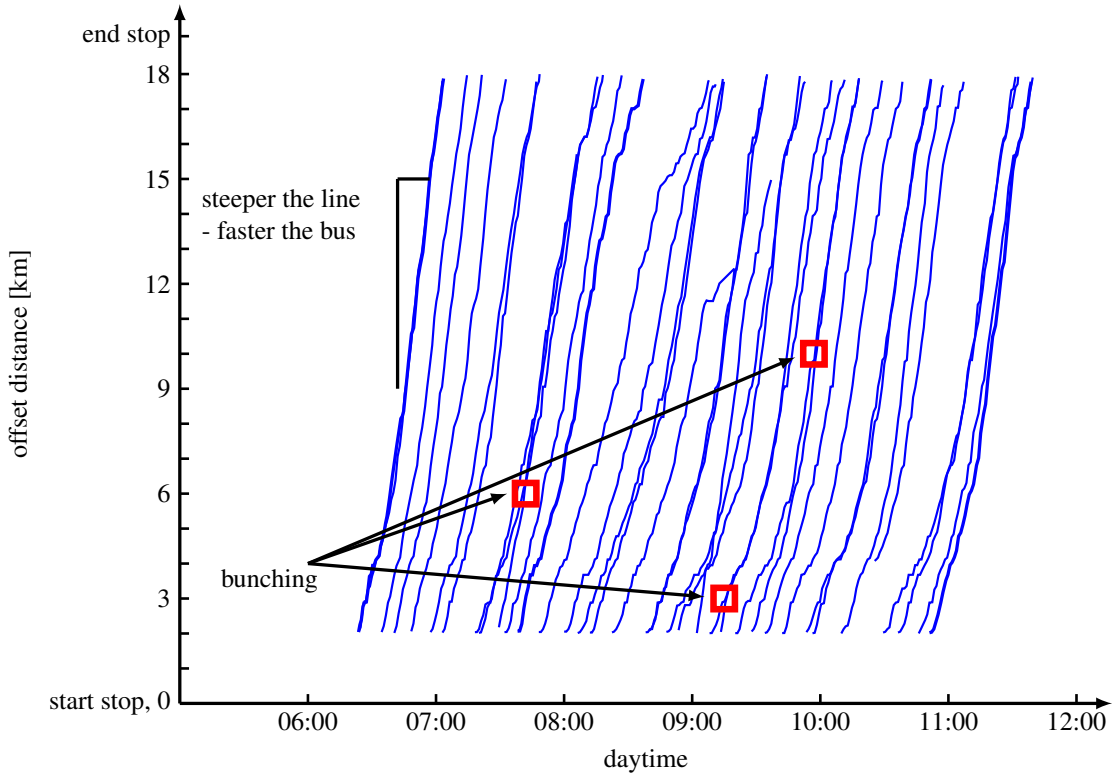


Figure 2.2: Space-time diagram: Route *46A, outbound,* 8th Nov 2012 [2]

time headway unless stated otherwise.

2.1.1 Offset Metric and Offset Distances

In the previous part we motivated the transformation of spatial coordinates to the one-dimensional representation called *offset distances*. In this subsection we formulate this problem more precisely. We present several methods to compute the offset distances and address the problem of disturbed data.

In order to define the term of *offset distances* we need to introduce a few technical terms, which are illustrated in figure 2.3.

Definition 2.1.1 — Shape. Let $S_{\text{nodes}} = \{n_0, n_1, \ldots, n_{N_S}\} \subset S_{\text{earth}}$ be a set of points on the earth's surface S_{earth}, which we approximate by a *reference ellipsoid* w.r.t. the WGS84 in this thesis. We define the *shape S corresponding* to S_{nodes} as the *geodetic polygon* that connects the points in S_{nodes} consecutively, i.e.,

$$\begin{aligned} S &:= \bigcup_{i=1}^{N_S} S_i, \\ S_i &:= \{x \in \gamma_i\} \subset S_{\text{earth}}, \quad i = 1, \ldots, N_S, \end{aligned}$$

[2] based on [Nai+14]

where γ_i is the shortest geodetic curve on S_{earth} connecting n_{i-1} and n_i (also called *geodesic* between n_{i-1} and n_i). The *geodetic distance* $\|\cdot\|_{\text{gd}}$ between two points on S_{earth} is defined as the arc length of the geodesic between the two points. We call the sets S_i *shape segments*.

For details on *geodesy*, which describes the analysis on the surface of a sphere, we refer to the appendix A.4.

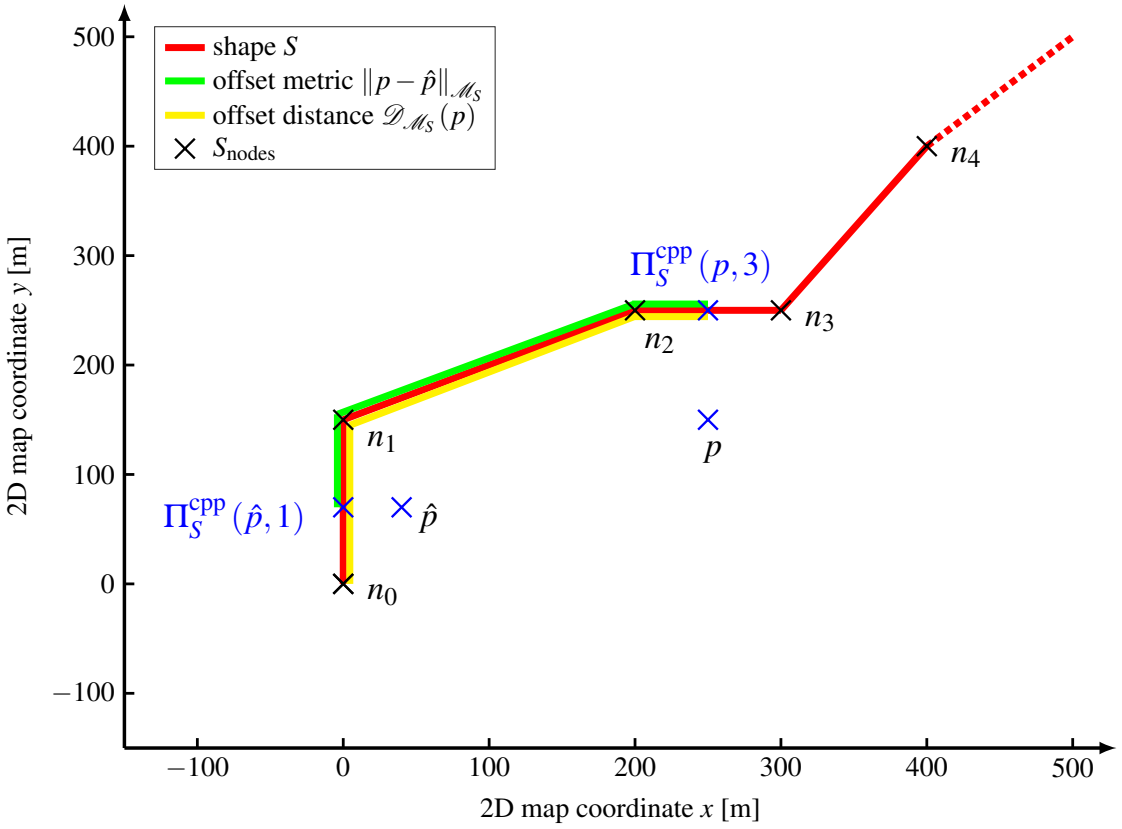


Figure 2.3: Illustration of the terms *shape, offset distance* and *offset metric*

For the following let $S_{\text{nodes}} = \{n_0, n_1, \ldots, n_{N_S}\} \subset S_{\text{earth}}$ be a set of points and S the corresponding shape. Let further $P = (p_0, p_1, \ldots, p_{N_T}) \subset S_{\text{earth}}^{N_T+1}$ be a finite sequence of points on the earth's surface and let $T = (t_0, t_1, \ldots, t_{N_T}) \subset \mathbb{R}_{\geq 0}^{N_T+1}$ be a monotonically increasing finite sequence of equidistant points in time, which we interpret as timestamps of the corresponding points in P.

For some shape S with the shape segments $\{S_i\}_{i=1}^{N_S}$ we introduce the *segment mapping* as a function

$$\mathscr{M}_S \colon S_{\text{earth}} \to \{1, \ldots, N_S\},$$

which maps each point on the earth's surface to the index of a 'suitable' shape segment S_i.

We call a segment mapping $\mathscr{M}_S$ *canonical*, if it maps each point in S onto the smallest of indices of the segments it is contained in.
The *segment projection* is defined as a *closest-point-projection* in the mathematical sense (see, e.g., [Alt12]) onto a shape segment S_i, i.e.,

$$\begin{aligned}\Pi_S^{\text{cpp}} \colon & S_{\text{earth}} \times \{1,\dots,N_S\} \to S,\\ & (p,i) \mapsto \overline{\underset{q\in S_i}{\operatorname{argmin}} \left(\|q-p\|_{\text{gd}} \right)}^{S_i},\end{aligned}$$

where $\overline{\cdot}^{S_i}$ describes the *mean value w.r.t. the segment* S_i: for a set of points $\{q_1,\dots,q_N\} \subset S_i$ we define $q^* := \overline{\{q_1,\dots,q_N\}}^{S_i}$ as the point $q^* \in S_i$ such that

$$\|q^* - n_{i-1}\|_{\text{gd}} = \frac{1}{N} \sum_{j=1}^{N} \|q_j - n_{i-1}\|_{\text{gd}}$$

holds. In here we take the mean value w.r.t. the segment S_i to encounter the case that the 'argmin' operator does not give us a unique output. We would like to point out here that this special case is rather a formal consideration since in this thesis we restrict our geodesic data on a 30[km]×30[km] region around Dublin.
For some segment mapping $\mathscr{M}_S$ we call its concatenation with the segment projection Π_S^{cpp} the *shape projection* $\Pi_{\mathscr{M}_S}$:

$$\begin{aligned}\Pi_{\mathscr{M}_S} \colon & S_{\text{earth}} \to S,\\ & p \mapsto \Pi_S^{\text{cpp}}\left(p, \mathscr{M}_S(p)\right).\end{aligned}$$

Definition 2.1.2 — Offset Distance, Offset Metric. We consider a set of points $S_{\text{nodes}} = \{n_0, n_1, \dots, n_{N_S}\} \subset S_{\text{earth}}$ and its corresponding shape S. Let further be $p, \hat{p} \in S_{\text{earth}}$ and let $\mathscr{M}_S$ be a *segment mapping*. We define the *offset distance* $\mathscr{D}_{\mathscr{M}_S}(p)$ of p as

$$\begin{aligned}\mathscr{D}_{\mathscr{M}_S}(p) &:= \left\|\Pi_{\mathscr{M}_S}(p) - n_{i_p-1}\right\|_{\text{gd}} + \sum_{i=1}^{i_p-1} \|n_i - n_{i-1}\|_{\text{gd}},\\ i_p &:= \mathscr{M}_S(p).\end{aligned}$$

Further we define the *offset metric* as

$$\begin{aligned}\|\cdot\|_{\mathscr{M}_S} \colon & S_{\text{earth}} \times S_{\text{earth}} \to [0, \mathscr{D}_{\mathscr{M}_S}(n_{N_S})],\\ & (p,\hat{p}) \mapsto \|p-\hat{p}\|_{\mathscr{M}_S} := \left|\mathscr{D}_{\mathscr{M}_S}(p) - \mathscr{D}_{\mathscr{M}_S}(\hat{p})\right|.\end{aligned}$$

If we restrict the offset metric to S and choose a canonical segment mapping $\mathscr{M}_S$, i.e., each point in S is mapped onto its true shape index, we obtain a metric in the mathematical sense on S (see, e.g., [Alt12]). We would like to point to the connection to differential geometry: we could interpret a shape as the image of a continuous and piecewise continuously differentiable *curve* on the earth's surface, and the offset distance would be the corresponding arc length ([Bro+12]).

The motivation to introduce the segment mapping $\mathscr{M}_S$ is the following: in this thesis we consider real data from the *GPS*[3] (see, e.g., [LRT15]) of *Dublin Bus*. Due to noise disturbing the GPS data (see [MHD99]) and the fact that buses happen not to follow precisely the shape of the prescribed journey pattern (e.g., due to changed traffic conditions like construction sites, new drivers, etc.), in general the spatial coordinates corresponding to the GPS data are not element of the considered shape. In this case we need to project the sample points to the shape and thus decide in which segment they were originally contained in. Furthermore formulating the shape projection using a segment mapping allows us to include contextual information later on.

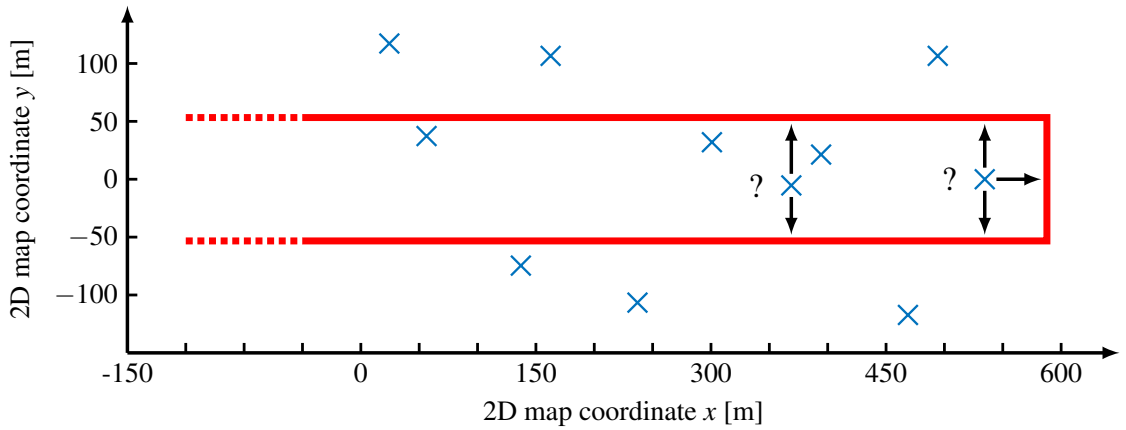


Figure 2.4: Map matching problem illustrated by a U-turn

An intuitive choice for the segment mapping is based on the closest-point projection to S w.r.t. the geodetic norm, i.e.,

$$\begin{aligned} \mathscr{M}_S^{\mathrm{cpp}} \colon S_{\mathrm{earth}} &\to \{1,\dots,N_S\}, \\ p &\mapsto \left\lfloor \overline{\operatorname*{argmin}_{i=1,\dots,N_S} \left(\left\| \Pi_S^{\mathrm{cpp}}(p,i) - p \right\|_{\mathrm{gd}} \right)} \right\rfloor , \end{aligned} \tag{2.1}$$

where $\Pi_S^{\mathrm{cpp}}(p,i)$ again is the classical closest-point projection onto the segment S_i and $\lfloor \overline{\cdot} \rfloor$ the rounded standard mean value. We need to take the mean value in equation (2.1)

[3] `http://www.gps.gov/systems/gps/`

in order to define a unique mapping, since the 'argmin' operator in general does not give us a unique result, and we need to round the output to obtain an integer number.

Beside the possibly ambiguous output of the 'argmin' operator a second problem of this approach is that due to data noise even in the case of a unique output of the 'argmin' operator equation (2.1) may not give us the original shape segment of the sample point. This could happen for example in U-turns or tight curves of the shape. This problem is also called *map matching problem* ([NK09]) and is illustrated in figure 2.4, where the red lines represent shape segments and the blue markers represent GPS samples, which have been projected to a two-dimensional map (see appendix A.5)).

2.1.2 Predecessor-Based Approach

The closest-point shape projection $\Pi_{\mathscr{M}_S^{\text{cpp}}}$ maps each point in $P \subset S_{\text{earth}}$ uniquely to a point in S without using contextual information of the sequence provided by the timestamps, i.e., information on neighboring points. Thus it is not able to solve the map matching problem accordingly due to data noise like illustrated in figure 2.4. This motivates us to introduce a new kind of shape projections which explicitly include contextual information. We would like to point out that this does not give us a projection from S_{earth} to S, since a point $p \in S_{\text{earth}}$ might be projected to two different points in S depending on the finite sequence P it is contained in. We rather get a mapping from the set of all possible finite sequences in S_{earth}, which also contains P, to the set of finite sequences in S.

A very simple approach which we call the *predecessor-based approach* modifies the classical segment mapping in equation (2.1) and is based on the assumption that the position of a point should be "close" to the position of the preceding point, which is true if the frequency of location samples is "sufficiently high". For a control parameter $\alpha \in [0,1]$ we define the predecessor-based segment mapping as

$$\begin{aligned}
\mathscr{M}_S^{\text{predec}} \colon & \left\{ (p_0, p_1, \ldots, p_{N_T}) \in S_{\text{earth}}^{N_T+1} \;\middle|\; N_T \in \mathbb{N} \right\} \to \bigcup_{N_T \in \mathbb{N}} \{1, \ldots, N_S\}^{N_T}, \\
& p_k \mapsto \bar{p}_k = \left\lfloor \overline{\underset{i=1,\ldots,N_S}{\operatorname{argmin}} \Big(\alpha \left\| \Pi_S^{\text{cpp}}(p_k, i) - p_k \right\|_{\text{gd}}} \right. \\
& \qquad \left. \overline{+ (1-\alpha) \left\| \Pi_S^{\text{cpp}}(p_k, i) - \bar{p}_{k-1} \right\|_{\mathbb{1}} \Big)} \right\rfloor, \quad k > 0, \\
& p_0 \mapsto \mathscr{M}_S^{\text{cpp}}(p_0),
\end{aligned} \tag{2.2}$$

where $\|\cdot\|_{\mathbb{1}}$ is the offset metric w.r.t. a canonical segment mapping on S and $\bar{p}_{k-1}$ denotes the priorly projected point. For $\alpha = 1$ we again get the standard closest-point segment mapping in equation (2.1), which we also apply to the first point p_0. Again we need to

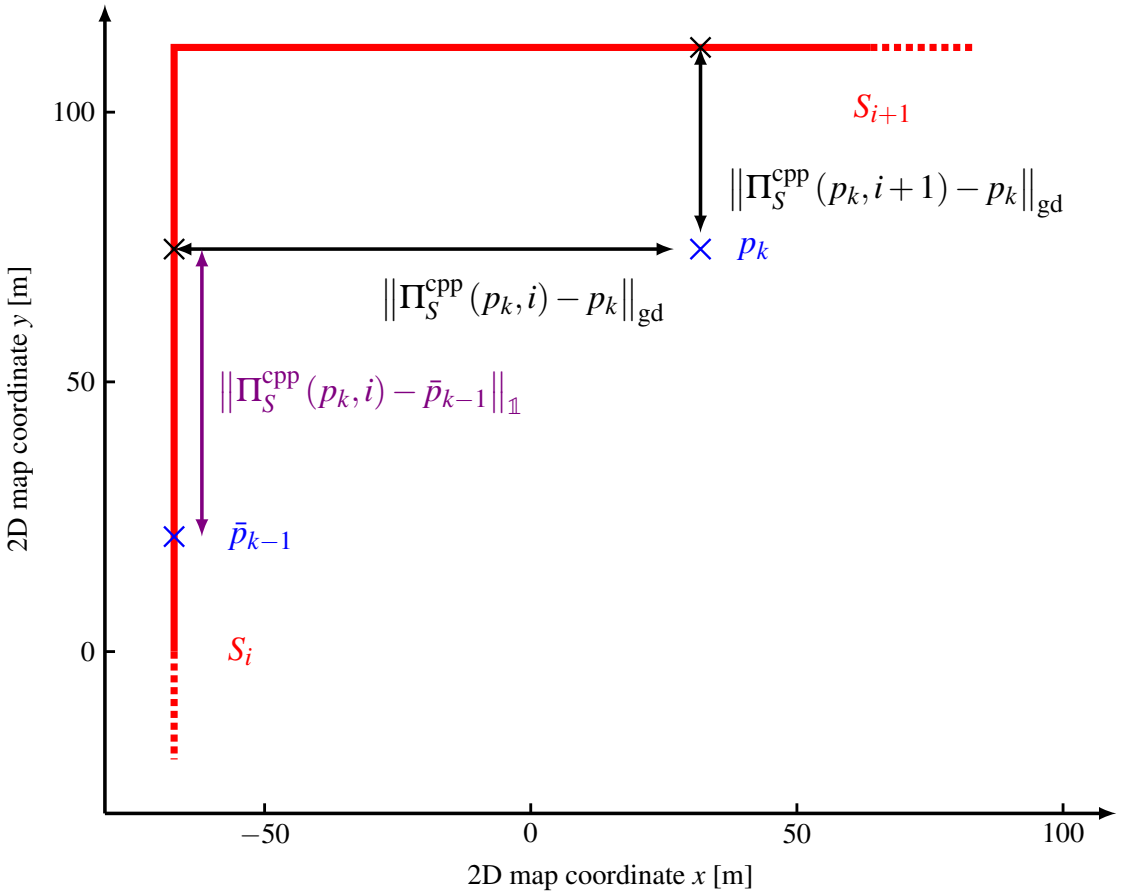


Figure 2.5: Illustration of the predecessor-based shape projection

consider the rounded mean value in equation (2.2) to define a unique mapping to integer values.

The parameter α represents the "trust" in the current measurement. The last term $(1-\alpha)\left\|\Pi_S^{\text{cpp}}(p_k, i) - \bar{p}_{k-1}\right\|_{\mathbb{1}}$ in equation (2.2) penalizes the suggested projection $\Pi_S^{\text{cpp}}(p_k, i)$ being far away from the priorly projected point w.r.t. the offset metric. This approach is illustrated in figure 2.5.

We see that this simple approach gives a qualitative improvement compared to single point projections. The method is computationally efficient and only uses one design parameter α. One problem of this approach is that there is no way to ensure or to give preference to monotonically increasing space-time trajectories. It might happen that outliers are projected to points on the shape with smaller offset distances than the projections of their predecessors, which causes negative slopes in the space-time diagram.

A more sophisticated approach that we present in the appendix A.1 uses *Hidden Markov Models* to construct a segment mapping. This approach also does not ensure monotonically increasing space-time trajectories, but at least allows to give preference to monotonically

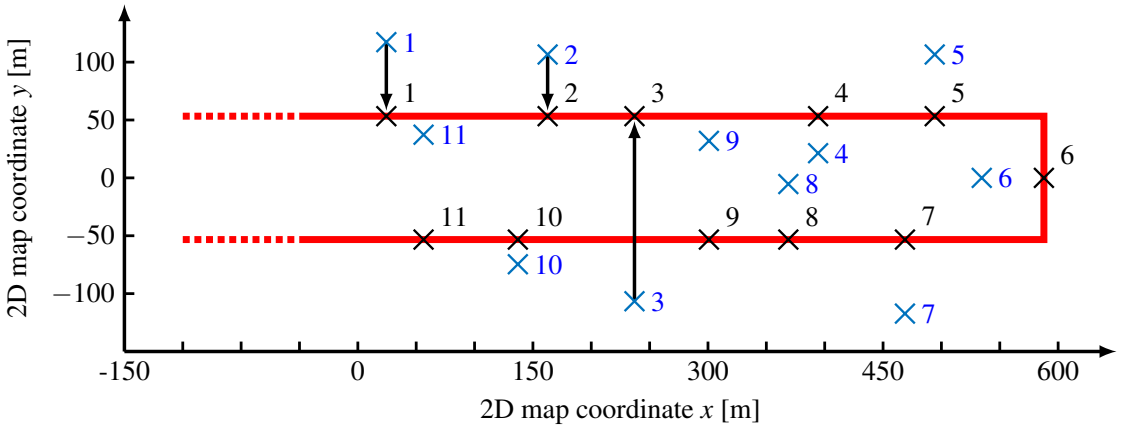


Figure 2.6: Map matching problem of the U-turn solved by the HMM shape projection

increasing sequences of segment indices returned by the segment mapping.

Figure 2.6 shows the application of our HMM approach to the U-turn problem.

2.2 Time Headways

In this section we describe the concept of *time headways*. As mentioned above this is the second concept beside the space-time diagram for investigating the bus bunching phenomenon. Before we define the time headways we point out that in the following we refer with the term 'bus' not to one specific vehicle of the vehicle fleet of the service provider but to the bus serving a particular *vehicle journey*, i.e., a certain scheduled bus run.

Definition 2.2.1 — Time Headway. Let $s \in S$ be a position along the shape S. We denote by $d_n(t)$ the offset distance of the location of bus n at time $t \in \mathbb{R}_{\geq 0}$, w.r.t. some segment mapping. Let $n-1$ denote the preceding bus of the same route and direction according to the schedule.

1. The *space-based time headway* $h_{n,s}$ between the buses $n-1$ and n at position s is defined as the difference between their arrival times $a_{n-1,s}$ and $a_{n,s}$ at s, i.e.,

$$h_{n,s} = a_{n,s} - a_{n-1,s}.$$

2. The *continuous-time headway* $h_n(t)$ between the buses $n-1$ and n at time t is defined as

$$h_n(t) := h_{n,d_n(t)}.$$

The two concepts of time headways are illustrated in the space-time diagram 2.7. In chapter 3 we focus on the continuous-time headways, which are defined similar to [Pil09],

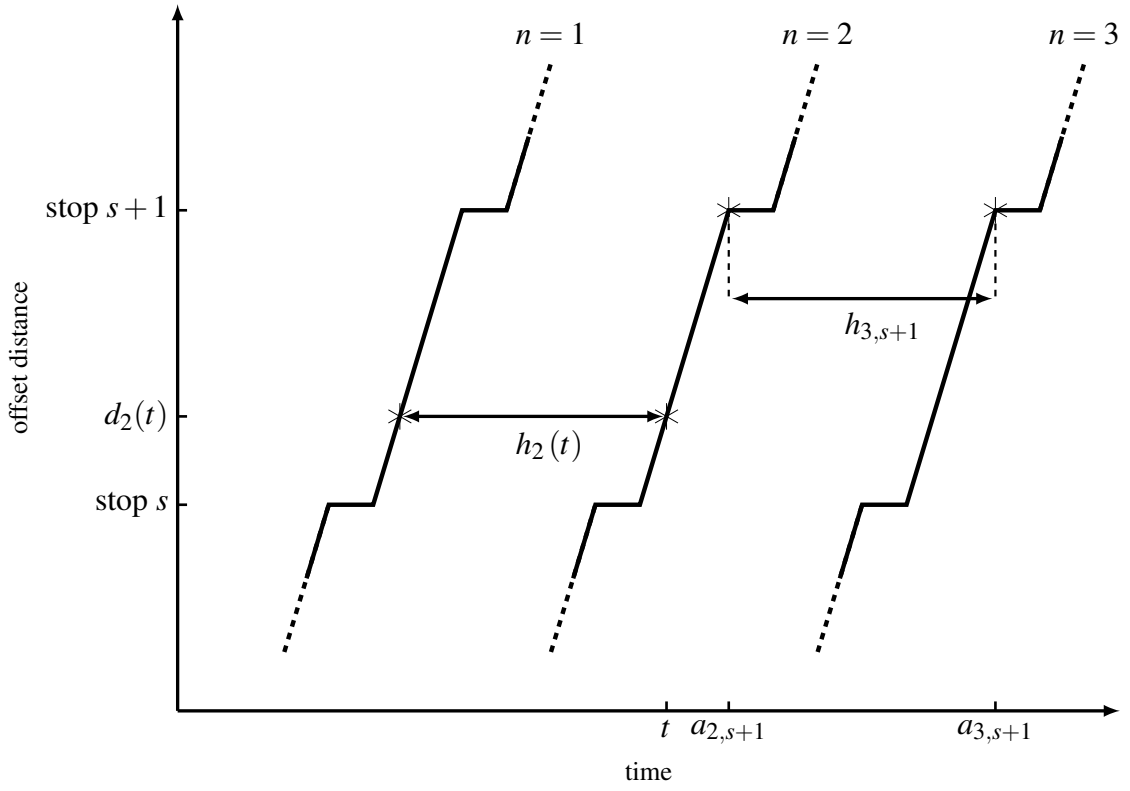


Figure 2.7: Time headway

since these provide us with headway samples at equidistant points in time. Therefore we are able to view the continuous-time headways as time series which allows us to apply techniques out of the field of *time-series analysis* in order to predict future headways. We study this approach in chapter 3.

In chapter 4 we investigate *stop-based headways* like in [Dag09], i.e., we study the headways for locations s that are chosen to be the bus stops of the route. For these we show that they tend to zero under certain assumptions in a disturbed and uncontrolled system and we present a dynamic-holding-control scheme to stabilize the headway distribution w.r.t. a scheduled *target headway*.

In figure 2.8 we visualize the headway distribution of route *46A, outbound,* of the 8th of November 2016. The X-axis shows the daytime and the Y-axis contains the vehicle journey IDs. The colors represent the time headways in minutes. We are able to identify bunching events as those points where the absolute values of the headways become close to zero. We also observe so called *second moment effects*, which describe the phenomenon that under certain assumptions small headways of a bus pair are followed by large headways of the succeeding bus pair and vice versa. This phenomenon is further studied in proposition 4.1.

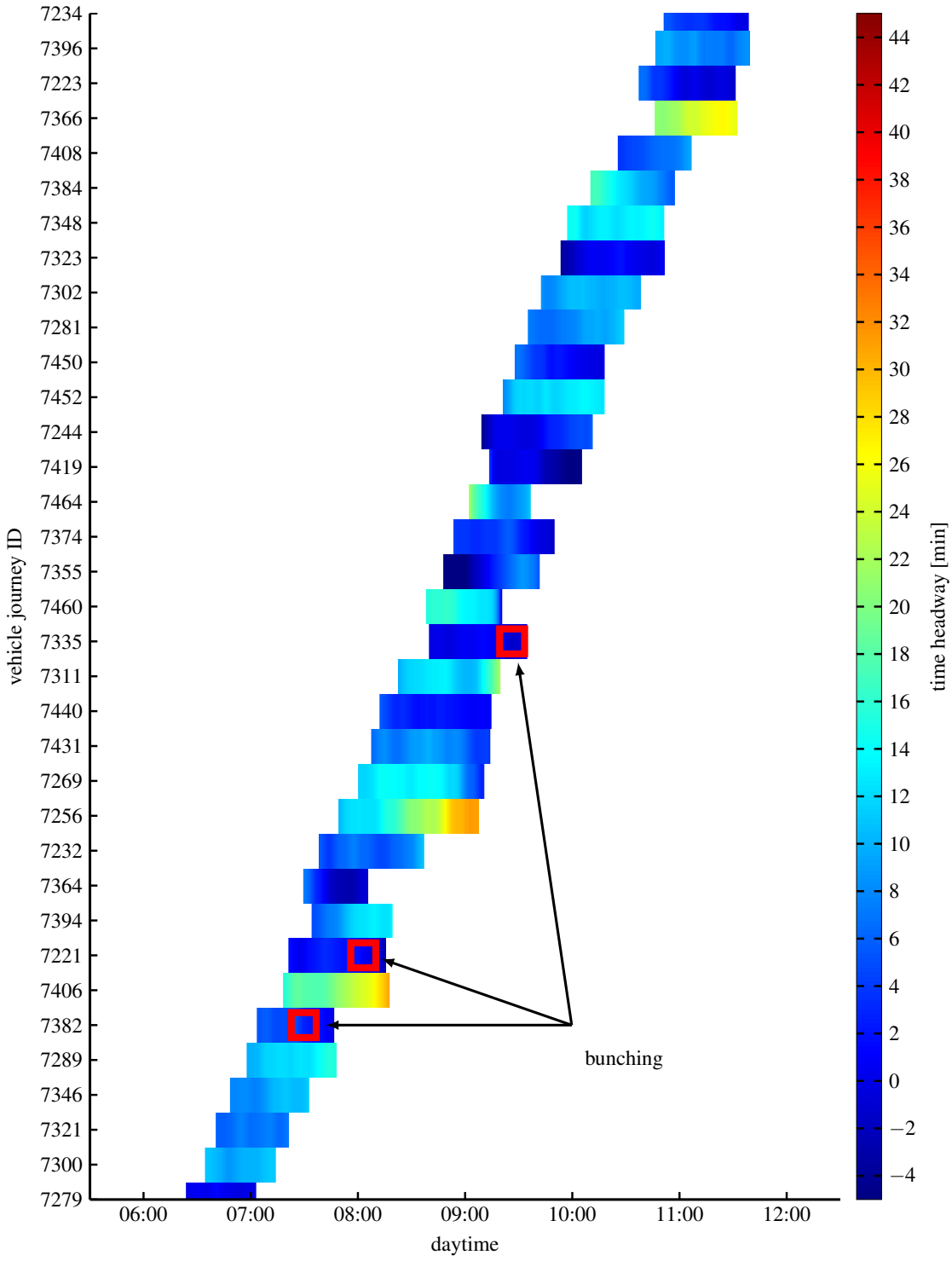


Figure 2.8: Visualization of time headways of route *46A, outbound,* 8th Nov 2016

2.3 Data Acquisition

In this section we briefly discuss the real data used in this thesis.

2.3.1 Data Structure

The data are freely available and are provided by a partnership between the four *Dublin Local Authorities* and *Maynooth University* called *Dublinked* [4]. We use two main datasets. The first set contains *Service Interface for Real-Time Information (SIRI)* data and gives information on the bus locations like timestamped GPS data. The second set contains *General Transit Feed Specification Reference (GTFS)* data and provides infrastructure data of the transit system like stop locations and schedules.

In figure 2.9 an overview on the main parts of both data structures and a link between those considering different notation is given. Especially it is shown which information is needed to extract the trajectory of a single bus run from the dataset. For details and notation of the SIRI and GTFS datasets we refer to the appendices A.2 and A.3.

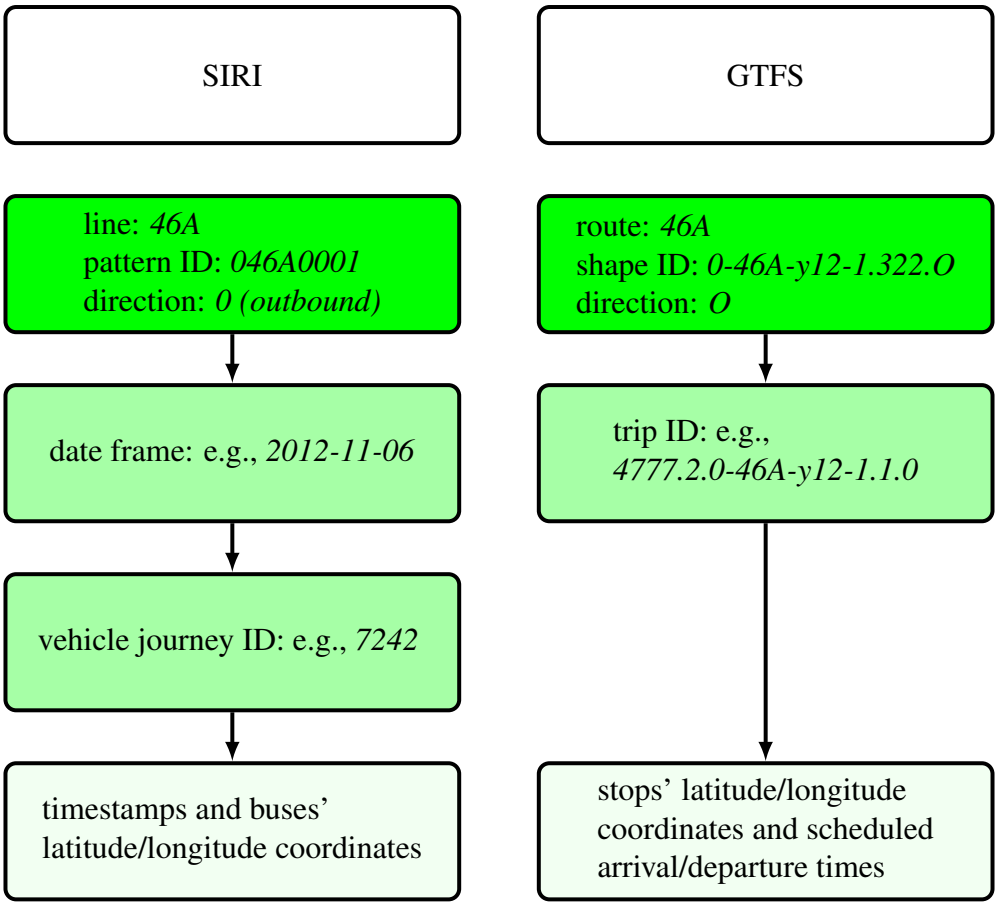


Figure 2.9: Matching SIRI and GTFS notation

[4] http://dublinked.ie/

For the following we assume that we are able to extract timestamped GPS samples for each vehicle journey from the SIRI dataset and match those with the scheduled trips given by the GTFS dataset. Furthermore we assume that for each vehicle journey we are able to extract the information on precedent and subsequent vehicle journeys according to the schedule.

It takes some effort to satisfy these assumptions when working with the real data. We omit the technical details of the implementation here. In order to give an idea about the additional effort for the implementation caused by working with real data we just would like to point out the following: even though buses belong to the same bus route and direction they might be operated on different *journey patterns* with different shapes, depending, for example, on the time of the day. This causes problems since we need to compute offset distances w.r.t. different shapes what makes it hard to analyze the space-time trajectories and the relation between those. Furthermore it might be ambiguous to determine for vehicle journey n the preceding vehicle journey $n-1$ since the bus corresponding to $n-1$ might be operated just on a part of the same shape that the bus corresponding to n is operated on.

In order to give an intuition of the different shapes of the route *46A, outbound,* we visualize those in figure 2.10 using the *Mercator projection* described in the appendix A.5.

On the webpage of Dublin Bus[5] another representation of one of those shapes can be found in a *Google Maps Representation* like shown in figure 2.1.

2.3.2 Preselecting the Offset Data

We observe that the data are disturbed by some random noise and contain strong outliers. Figure 2.11 shows the Mercator plot of the raw GPS samples of route *46A, outbound,* of the 8th of November 2016, the corresponding shapes and the corresponding space-time diagram, which we computed using the HMM segment mapping.

The error source has two main parts: the first part is the 'internal' measurement noise included in the data, for example, caused by disturbed signal transmissions or limited computational accuracy. The second reason is caused not by the measurement device, but 'externally' by buses that do not drive along the prescribed shape of their journey pattern. This might happen for example due to temporary road works, road closures (e.g., caused by special events like St. Patrick's Day), new bus drivers etc.. For this reason it is necessary to preselect the data before using them in a prediction framework. We use a very intuitive set of criteria, which is based on the criteria in [NK09], to mark each single GPS sample within a vehicle journey with corresponding shape S as outlier or not.

[5] `https://www.dublinbus.ie/RTPI/Sources-of-Real-Time-Information/`

Our algorithm works as follows: first we only consider trajectories that provide at least $\theta_{\text{data}} = \lfloor 0.6 \times (\text{number of node points of shape } S) \rfloor$ data points. For each of those trajectories we mark those sample points as outliers, which have a geodetic distance to the shape greater than the threshold $\theta_{\text{pre}} = 200$ [m] (numeric value taken from [NK09]). After that we apply a map matching algorithm like described in the previous section with the reduced set and mark those points as outliers, which have a geodetic distance to their assigned projected point greater than the threshold $\theta_{\text{post}} = 200$ [m]. Then we check for 'jumps' in the trajectory, i.e., we mark those points in the reduced set as outliers with an offset velocity greater than the threshold $\theta_{\text{speed}} = 120$ [km/h], where the offset velocity is approximated using finite differences. Furthermore we observe that a lot of irregularities occur at the beginning and the end of a vehicle journey. This could be caused, for instance, by the fact that the GPS tracking might not start or end exactly at the start- or endpoint of the vehicle journey, but also on the way from or to another vehicle journey that is served by the same vehicle. For this reason we concentrate just on a part of each journey-pattern shape and neglect offset distances smaller than $\theta_{\text{min}} = 2000$ [m] and larger than $\theta_{\text{max}} = 18000$ [m].

Applying our algorithm does not remove all outliers and does not ensure positive offset velocities, but it provides us with a robust and intuitive method to preselect the data. Figure 2.12 shows the Mercator plot and the space-time diagram of route *46A, outbound,* of the 8th of November 2016 corresponding to the GPS samples selected by our algorithm. In the next chapter, where we address the prediction of headway data, we use the space-time trajectories of our selected data to compute headway trajectories and after that we further process these headway trajectories to prepare them for the prediction.

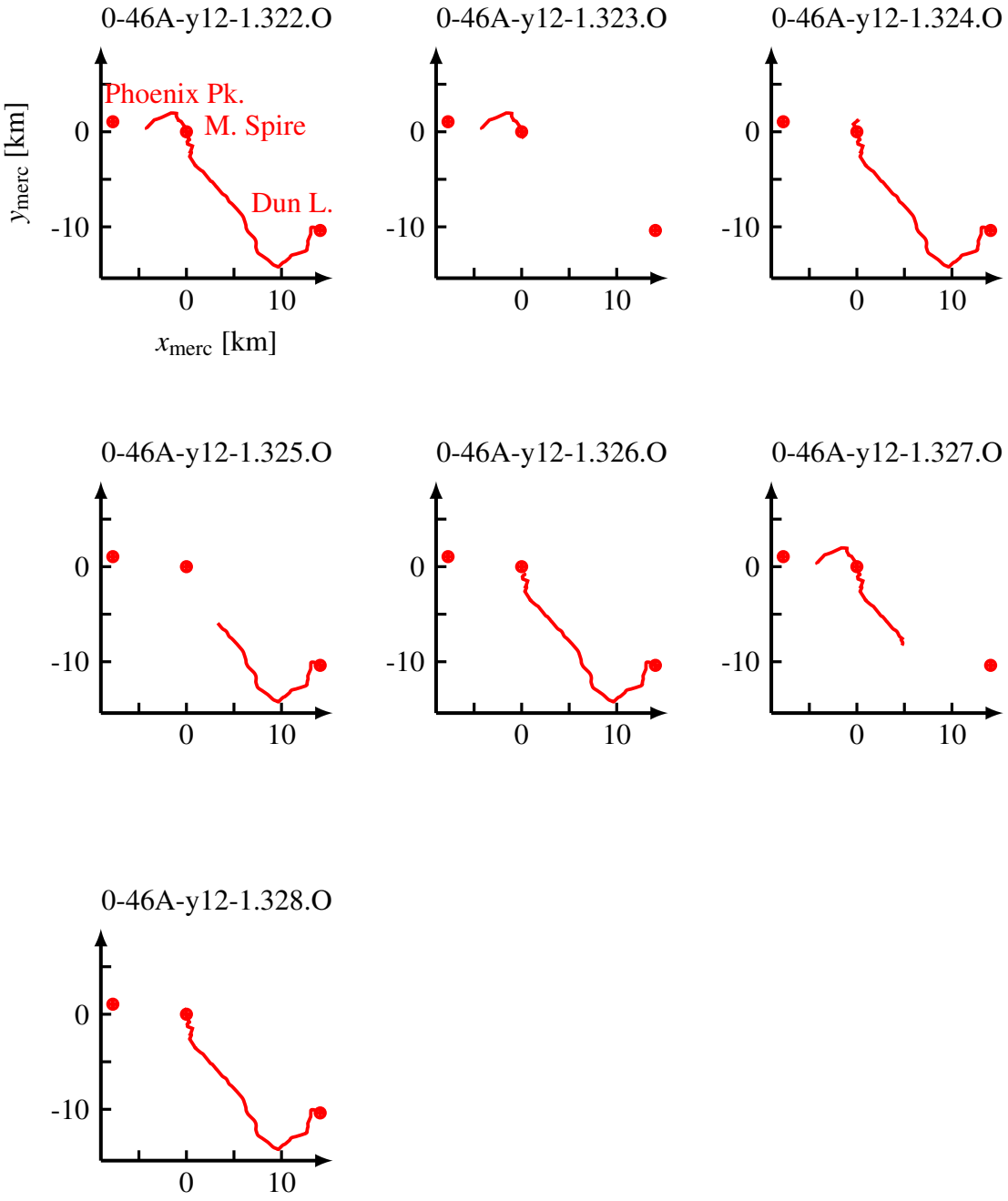


Figure 2.10: Shapes of route *46A, outbound*

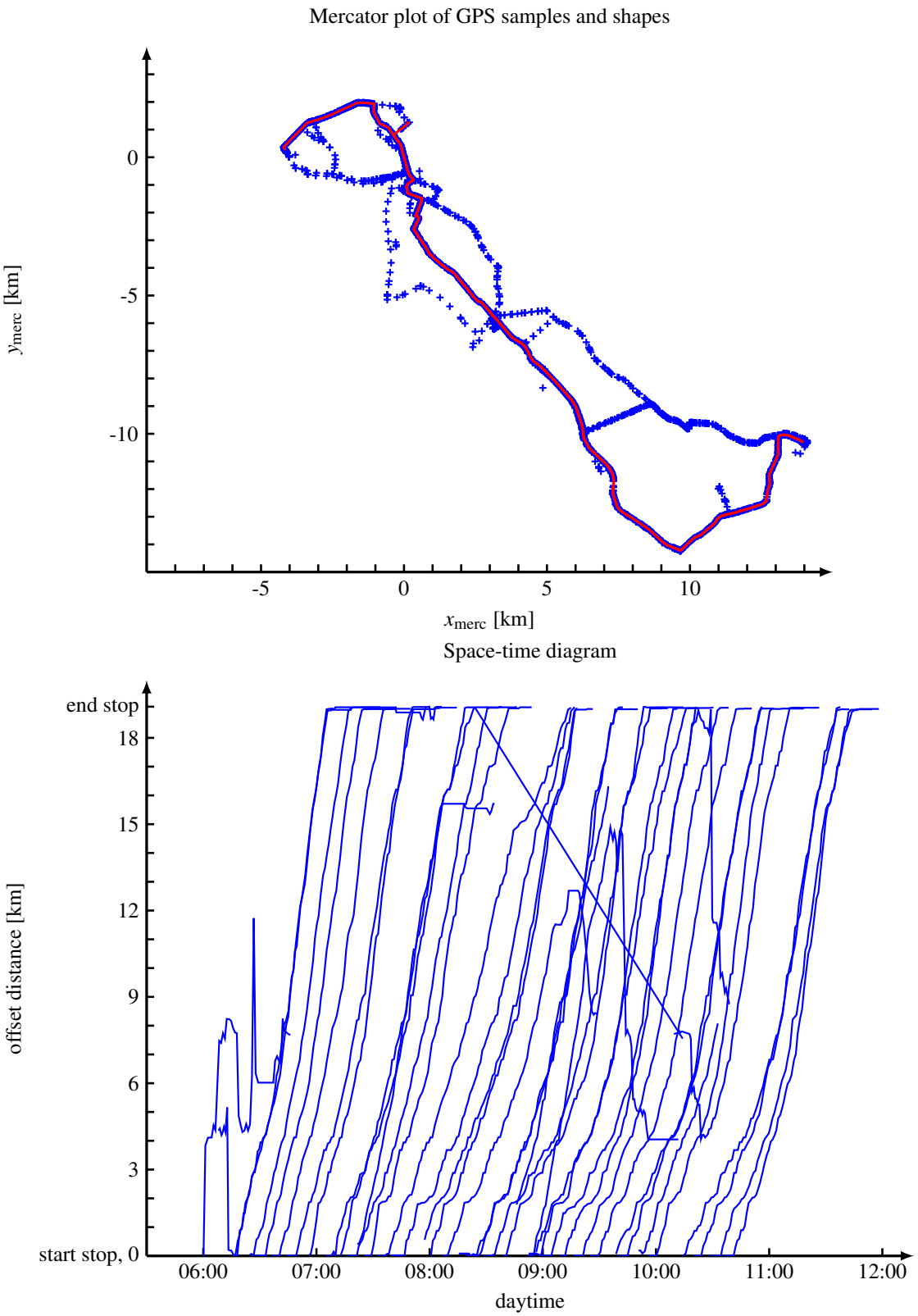


Figure 2.11: Raw GPS samples and space-time diagram of route *46A, outbound,* of 8th Nov 2016, including outliers

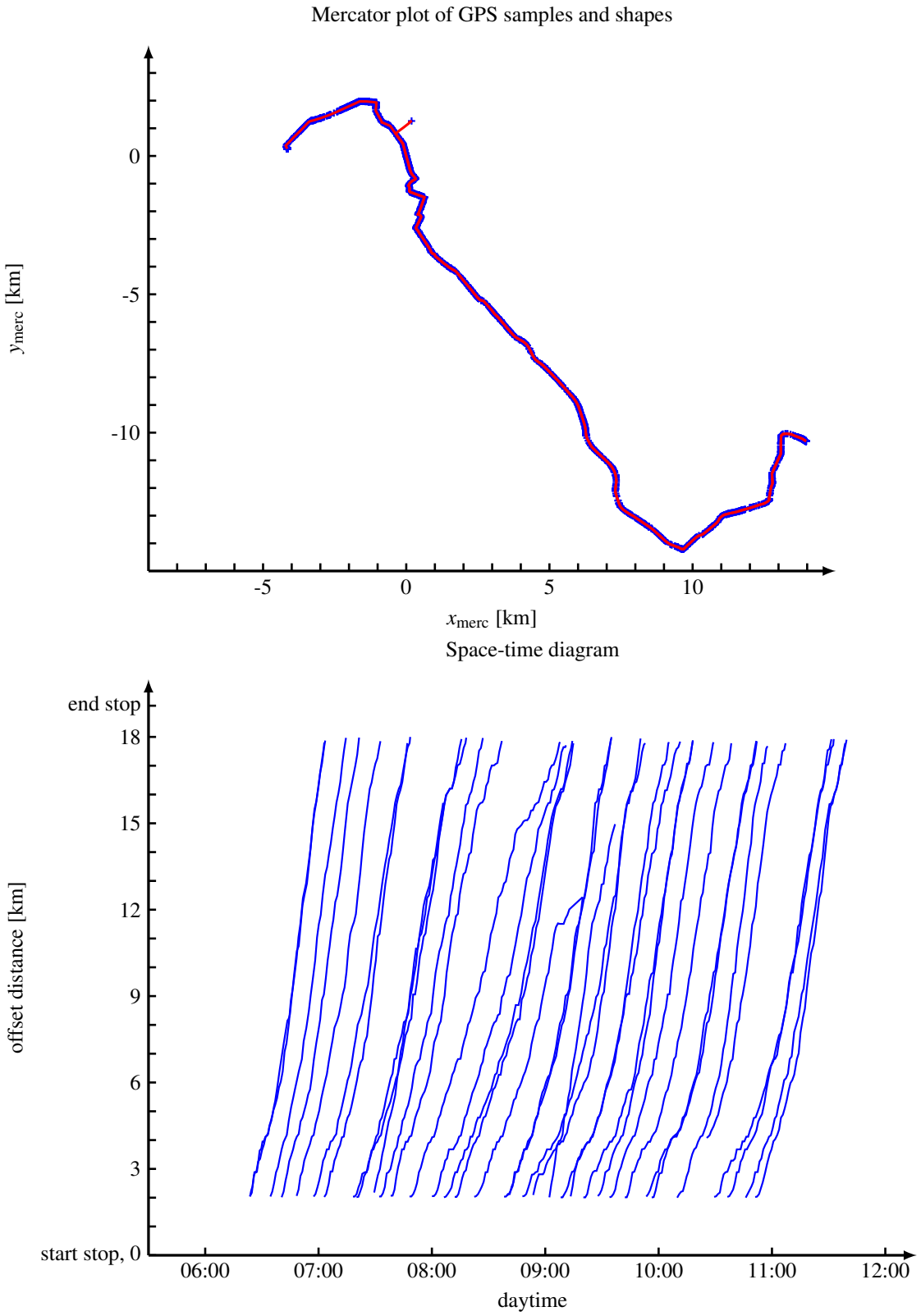
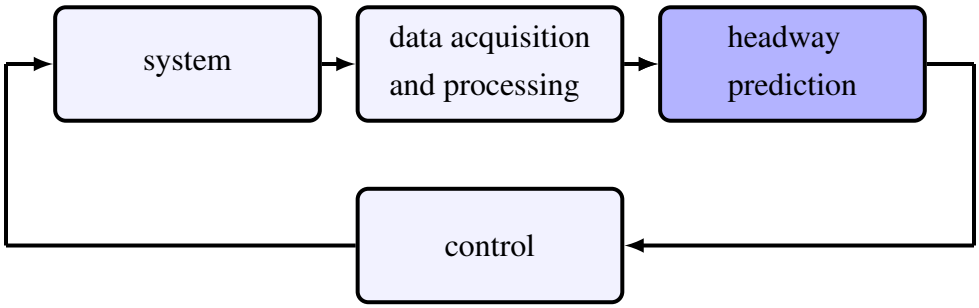


Figure 2.12: Selected GPS samples and space-time diagram of route *46A, outbound,* of 8th Nov 2016

3. Prediction of Future Headways

In this chapter we assume to have space-time diagrams available and we discuss the data-driven prediction of headway trajectories. First we present preprocessing techniques for those. Secondly we use *linear-regression-and-extrapolation-, kernel-regression-and-extrapolation-, artificial-neural-network-* and *autoregressive-models* to predict the headway trajectories and compare the performances of the different techniques.



3.1 Headway Preprocessing

As discussed in chapter 2 it is necessary to preprocess the headway data before applying an algorithm for prediction.

The first step of our algorithm is to mark those headway data points as outliers which have an absolute value greater than a certain threshold, in our case we chose $\theta_{\mathrm{h,max}} = 90$ [min]. After that we check for each headway trajectory the number of remaining data points and neglect those trajectories which have less than 100 data points. In order to obtain

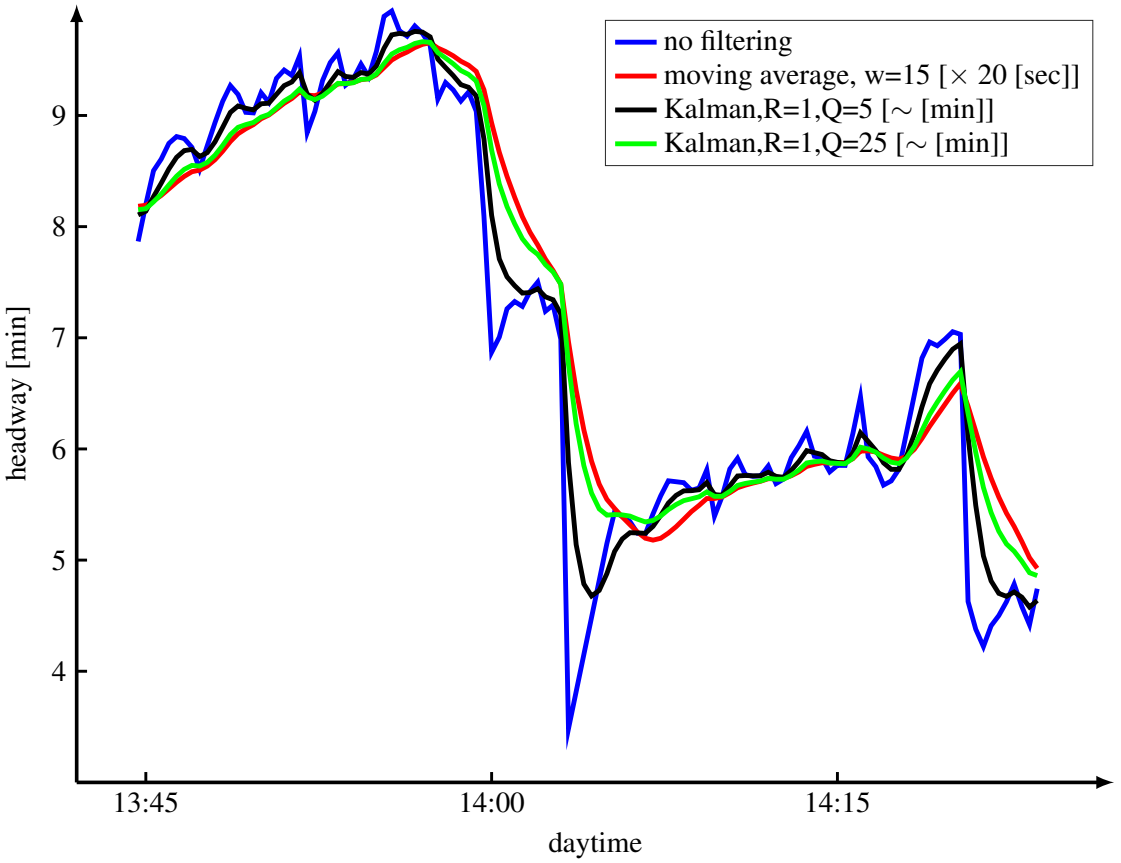


Figure 3.1: Different online filters applied to a headway trajectory

again headway trajectories with equidistant timestamps we apply a *linear interpolation* to the remaining trajectories to replace the data points rejected by the first step. As a last step we apply a *smoothing algorithm* to the trajectories. If we use the data as training data we may apply an *offline filter* like the *centered moving average filter*, or in the operational case of online data we apply some *online filter* like a modification of the moving average filter or the *Kalman filter*.

The idea of *smoothing* here is to reduce the fluctuation of the headway trajectories. Spontaneous fluctuations could be caused by unforeseen events occurring while operating the bus, like traffic lights, traffic congestions, higher or lower passenger demands. It is hard or even impossible to factor in all these components, so we just interpret those as random noise disturbing the offset trajectories and thus the headway trajectories. Since it is not our goal to detect or to predict these spontaneous events but the crucial information inherent in the data, like increase or decrease over longer time periods, we would like to remove these noise components and only work with the cleaned trajectories. To this there are plenty of algorithms available, some out of the fields of *time-series analysis* and *signal processing*. An overview can be found in [DM57].

In the appendix B.1 we describe some details of the *weighted moving average filter* and the *Kalman filter*. Here we would just like to point out that we have to deal with both *offline* and *online data*. The data we use for training or as reference data are considered to be *offline data*, i.e., the whole dataset is available at a time we apply our filter. But in the operational scenario we have to deal with *online data*, i.e., at a time we apply our filter only the data points of the current and the previous time steps are available. This means the used terms do not refer to the data structure but to the accessibility. Having this distinction in mind we are able to choose an appropriate filter.

In order to illustrate different online filters we show in figure 3.1 a headway trajectory and several filtered versions of the same, where the parameter R models the *process noise* and Q the *measurement noise* of the Kalman filter.

An overview about how to preprocess data as preparation for *data mining* and *supervised learning* can be found in [Pyl99] and [KKP06].

For the purpose of prediction we need to define the following quantities.

Definition 3.1.1 We consider a headway trajectory h with equidistant timestamps $(t_1,\dots,t_{N_h})$.

1. The *prediction horizon* $PH \in \mathbb{N}_{>0}$ determines how many time steps ahead we would like to predict the headway trajectory. For a time index $0 \leq k \leq N_h - PH$ we are interested in predicting the headway trajectory at time index $k+PH$.
2. The *historical horizon* $HH \in \mathbb{N}_{\geq 0}$ determines the range of preceding indices which we include for the prediction, i.e., for a time index $HH < k$ we make use of all data points of h corresponding to the indices $\{k-HH, k-HH+1,\dots,k\}$ for the prediction step at index k.

For a prediction horizon PH, a historical horizon HH and the headway trajectory h of length N_h we then can apply our algorithm to predict the headways corresponding to the time indices $\{HH+PH+1, HH+PH+2,\dots,N_h-PH\}$.

There are two main modes of prediction, illustrated in figure 3.2. In the first mode we apply a single-step-ahead prediction method multiple times consecutively to get the predicted value PH time steps ahead of the current time. In the second mode we apply a multi-steps-ahead prediction method which immediately computes the predicted value PH time steps ahead of the current time. (see [Cha00]).

In the following sections we present different techniques for the time-series forecasting and we discuss the performance on the given dataset. We would like to point out here that the prediction of headway data can also be done in connection with the prediction of bus arrival times, like for example in [Nai+14]. In here we focus on the prediction directly

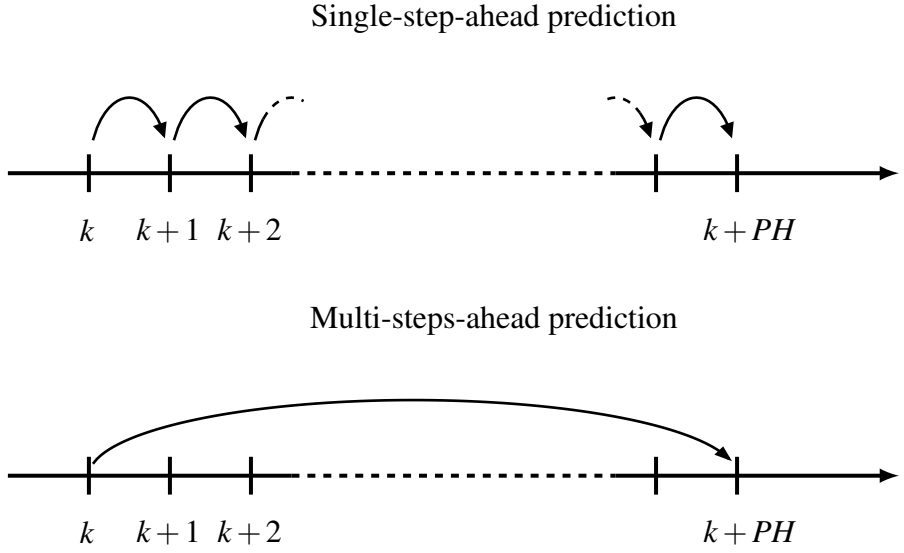


Figure 3.2: Prediction modes

based on the headway trajectories since this allows us to detect trends of the trajectories and is thus more suitable for bunching prediction.

For the following when we talk about the prediction of continuous-time headways we use the following notation for a headway trajectory h with timestamps $(t_1,\ldots,t_{N_h})$ between two buses $n-1$ and n:

$$h_k := h_n(t_k), \quad 1 \leq k \leq N_h.$$

3.2 Numerical Experiment

All tests of our prediction algorithms were performed on the data of the 28th of November 2012, which is the last Wednesday of the dataset, while the remaining data from the 6th to 30th of November 2012 might be used as reference data for the kernel regression method in section 3.4 or as training data for an artificial neural network in section 3.5. The headway trajectories are sampled with equidistant time steps of 20 seconds (frequency given by the SIRI dataset, see appendix A.2). We measured the performance of each prediction method based on four quantities. First we computed the RMSE (root-mean-square error) between the true trajectory and the predicted trajectory like in [Nai+14]:

$$\mathrm{RMSE}\left(h,h'\right) = \sqrt{\frac{1}{N_h}\sum_{k=1}^{N_h}\left(h_k - h'_k\right)^2}.$$

Secondly we checked for each trajectory the percentage of single predictions which are within certain *correctness bins*. We chose as bins one minute, two minutes and three minutes. For each method we then took the means of each of those quantities over all predicted trajectories and we compare the results for different parameter sets in the tables 3.2, 3.3, 3.4 and 3.5.

In the numerical experiments we compared for each method its performance on the test set for all combinations of parameters given in table 3.1. The meanings of the parameters are explained in the corresponding sections. In table 3.6 we compare the best performances of each method.

parameter	**value**
prediction horizon PH	5;10 [min]
historical horizon HH	3;5;10;15 [min]
model parameter	PR: deg 1 KR: σ 0.3; 1; 3 ∼[min] NN: structure $(10)\,;(20)\,;(30)\,;(10\|10)\,;$ $(20\|10)\,;(10\|10\|10)\,;(5)\,,(10\|5)$ ARMA: (p,q) $(4,0)\,;(7,0)\,;(10,0)\,;(13,0)\,;(16,0)$
window size (mov avg) w	∼ 0;3;6;9;12 [min]

Table 3.1: Parameter tuning in numerical experiment

3.3 Linear Regression and Extrapolation

The descriptions of this section and further details can be found, for example, in [Bro+12]. The idea of a *Linear-Regression-and-Extrapolation* model is illustrated in figure 3.3.

More general the idea of *polynomial regression* is to approximate a given set of data points by a real polynomial function $p(t) = \alpha_0 + \alpha_1 t + \ldots + \alpha_d t^d$ of degree d such that the error between the true data points and the approximations measured in the Euclidean norm is minimized. For the data points $(h_i)_{i=k-HH}^{k}$ for the historical horizon $HH \geq 0$ at times $(t_i)_{i=k-HH}^{k}$ we find the coefficients of this polynomial by solving the following *Least-Squares Problem*:

$$
\begin{aligned}
&(\alpha_0, \alpha_1, \ldots, \alpha_d)^T = \underset{\hat{\alpha}\in\mathbb{R}^{d+1}}{\operatorname{argmin}} \|A\cdot\hat{\alpha} - b\|_2^2, \\
&A = \begin{pmatrix} 1 & t_{k-HH} & \cdots & t_{k-HH}^d \\ \vdots & \vdots & & \vdots \\ 1 & t_{k-1} & \cdots & t_{k-1}^d \\ 1 & t_k & \cdots & t_k^d \end{pmatrix}, \quad b = \begin{pmatrix} h_{k-HH} \\ \vdots \\ h_{k-1} \\ h_k \end{pmatrix}.
\end{aligned} \tag{3.1}
$$

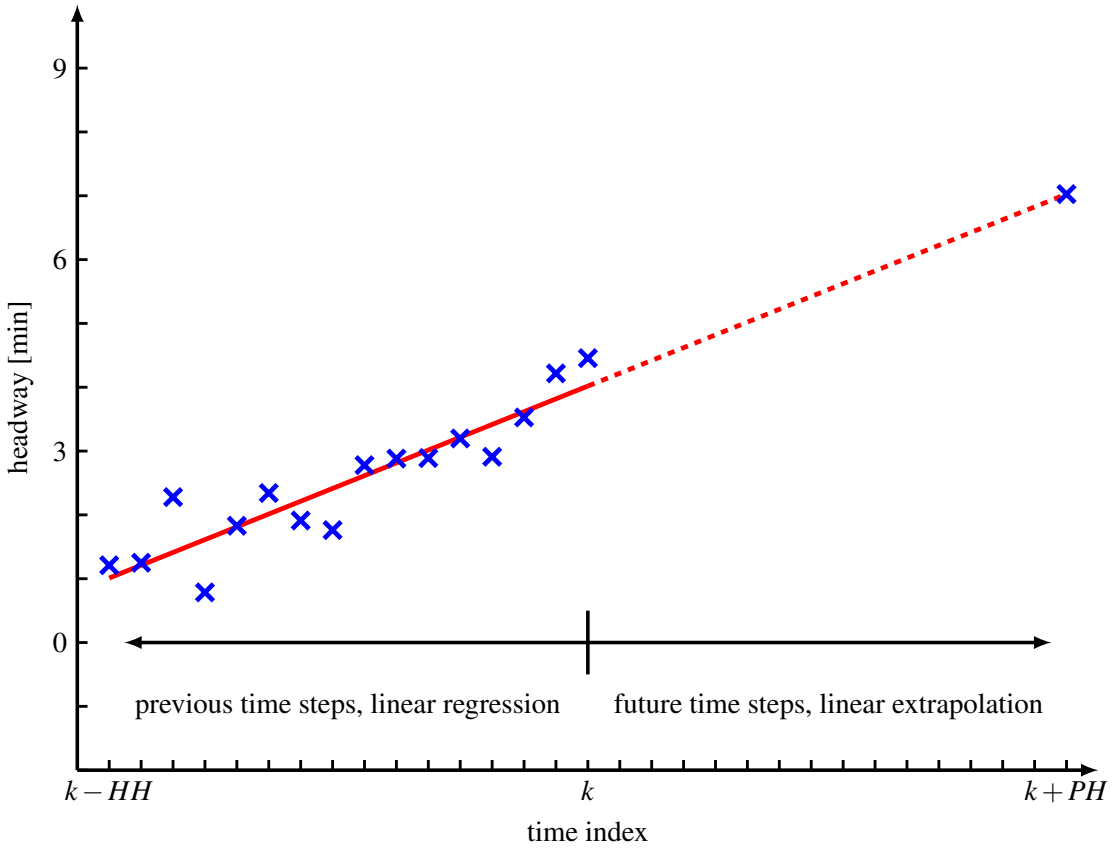


Figure 3.3: Linear regression and extrapolation

The minimization problem in equation (3.1) can be solved for example by using the *pseudo inverse* $A^{\#}$ of A (see, e.g., [And14]).
The prediction of the headway value PH time steps ahead of the current time is done by polynomial extrapolation, i.e., we evaluate the regression polynomial we found by equation (3.1) at the desired value t_{k+PH}:

$$h^{\prime\mathrm{LR}}_{k+PH} = \alpha_0 + \alpha_1 t_{k+PH} + \ldots + \alpha_1 t^d_{k+PH}.$$

In our case of *linear* regression and extrapolation we use a polynomial of degree $d = 1$ like shown in figure 3.3 and use it as a multi-steps-ahead prediction, i.e., we compute the coefficients in equation (3.1) and evaluate the linear polynomial directly at a time PH time steps ahead of the current time.

Even though this approach is very simple and the headway trajectories most likely do not show a linear behavior, we are still able to detect *trends* in the trajectory, i.e., we detect if the headway values tend to become larger or smaller as time progresses. In chapter 4 we investigate the properties of the headway trajectories based on a simplified bus-route

model. One result is that in our simplified deterministic model under certain assumptions headways tend to zero or increase until the end of the bus journey unless they are equal to the target headway. These trends can be detected by a simple linear regression what is the reason for us to consider this technique in this context.

Furthermore this technique allows to compute a *confidence value* that could be used in an operational setting to decide if we should trust the prediction or not. This could be done, for instance, by considering the error of the polynomial regression within the historical horizon measured in the Euclidean norm.

Figure 3.4 visualizes three headway trajectories beside the predicted values computed by a five-minutes-ahead prediction using linear regression and extrapolation and the corresponding RMSE of our experiment. Table 3.2 shows the best 15 parameter configurations w.r.t. the RMSE for each prediction horizon PH corresponding to 5 and 10 minutes resulting from our numerical experiment. The last three columns contain the means of the percentages of predictions within correctness bins of one, two and three minutes among all predicted headway trajectories.

PH [min]	HH [min]	w [min]	$\overline{\text{RMSE}}$ [min]	$\overline{\text{err < 1 min}}$ [%]	$\overline{\text{err < 2 min}}$ [%]	$\overline{\text{err < 3 min}}$ [%]
5	3	6	1.26	61.75	88.14	96.33
5	3	9	1.17	65.57	90.18	97.59
5	3	12	1.18	66.87	89.92	97.77
5	5	6	1.31	59.59	86.35	96.12
5	5	9	1.23	62.73	88.41	97.43
5	5	12	1.24	62.97	87.98	97.73
5	10	0	1.35	57.21	85.73	95.92
5	10	3	1.38	55.84	84.45	95.85
5	10	6	1.38	57.31	84.20	95.76
5	10	9	1.35	57.31	84.62	95.45
5	10	12	1.36	56.20	84.57	96.23
5	15	0	1.38	55.36	82.92	95.84
5	15	3	1.42	52.77	81.14	95.82
5	15	6	1.45	50.91	81.97	95.47
5	15	9	1.46	49.67	82.24	94.48
10	3	6	2.17	40.04	66.37	81.95
10	3	9	1.88	45.17	71.91	85.08
10	3	12	1.76	47.00	74.39	87.64
10	5	6	2.15	39.09	66.27	83.02
10	5	9	1.89	42.85	71.93	84.50
10	5	12	1.79	46.00	73.07	87.72
10	10	3	2.15	35.26	65.16	81.23
10	10	6	2.08	34.88	66.15	83.23
10	10	9	1.98	37.39	67.42	84.75
10	10	12	1.92	38.43	68.53	86.53
10	15	0	2.10	33.20	62.61	81.40
10	15	3	2.12	30.54	62.60	81.15
10	15	6	2.11	31.64	60.83	82.03
10	15	9	2.07	34.41	61.74	83.79
10	15	12	2.03	33.79	61.14	83.81

Table 3.2: Results of the parameter study for the linear-regression-and-extrapolation model (best 15 results w.r.t. RMSE for each prediction horizon *PH*)

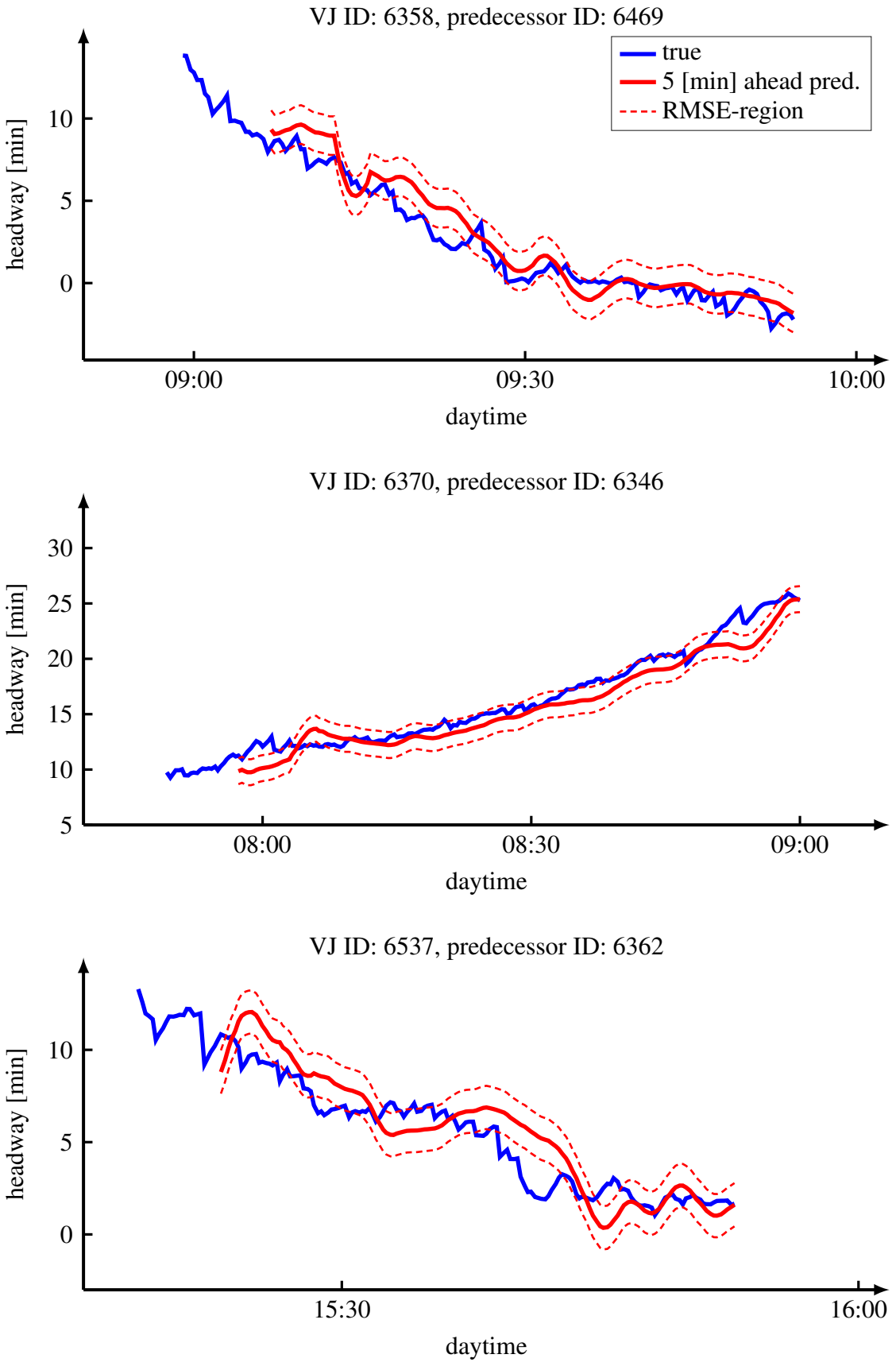


Figure 3.4: Illustration of the headway prediction using a linear-regression-and-extrapolation model, 28th Nov 2012

3.4 Kernel Regression and Extrapolation

In this section we describe the prediction technique *Kernel Regression and Extrapolation*. It is adapted from the method in [Sin+12], where the authors used it to predict space-time trajectories. For this method we need to define a subset of the headway trajectories as offline *reference data*. Instead of using this subset like in the neural network approach in section 3.5 as training data to tune parameters of the specific algorithm, we use this data directly as reference trajectories like in a data bank without applying any further training algorithm. We then compare the headway trajectory of interest h to all trajectories in the reference set within a certain range of preceding time steps of the current time step and compute a weighted sum of these trajectories, where the weights correspond to the similarity of the trajectories to h. The trajectory defined by this weighted sum is then taken as prediction of the trajectory h. This approach is illustrated in figure 3.5.

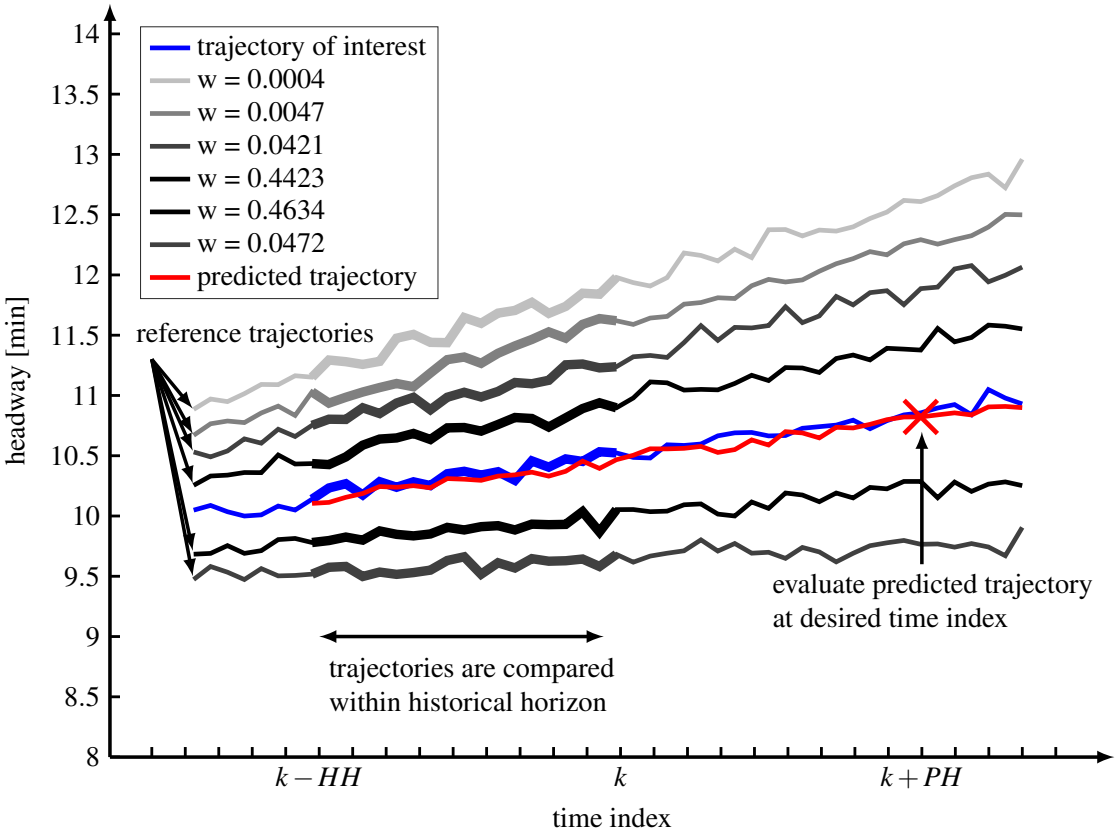


Figure 3.5: Illustration of the idea of kernel regression and extrapolation as prediction method

The motivation of this approach can be phrased like in [Nai+14]: *history repeats itself*. We formulate this idea more precisely in the following definition 3.4.1. For the sake of simplicity we assume that all headway trajectories as well those in the reference set as those we would like to predict have the same length N_h.

Definition 3.4.1 — Kernel Regression and Extrapolation. Let $\mathscr{H}_{\text{ref}} \subset \mathbb{R}^{N_h}$ be the set of reference headway trajectories, which we assume to be of the same length N_h. Let HH denote the historical horizon and PH the prediction horizon and let $K\colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ and $\sigma \in \mathbb{R}_{>0}$. We then define the predicted value h'^{KR}_{k+PH} for time t_{k+PH} at time t_k as

$$\begin{aligned} h'^{\text{KR}}_{k+PH} &= \frac{1}{Z} \sum_{\hat{h} \in \mathscr{H}_{\text{ref}}} K\left(-\frac{d_{HH}(h,\hat{h})}{\sigma}\right) \cdot \hat{h}_{k+PH}, \\ d_{HH}\left(h,\hat{h}\right) &= \frac{1}{HH+1} \cdot \left\| (h_i)_{i=k-HH}^{k} - \left(\hat{h}_i\right)_{i=k-HH}^{k} \right\|_2^2 \\ &= \frac{1}{HH+1} \sum_{i=k-HH}^{k} (h_i - \hat{h}_i)^2, \\ Z &= \sum_{\hat{h} \in \mathscr{H}_{\text{ref}}} K\left(-\frac{d_{HH}(h,\hat{h})}{\sigma}\right). \end{aligned}$$

The kernel regression and extrapolation method as defined in definition 3.4.1 is operated as a multi-steps-ahead prediction. In here the parameters that need to be tuned are again the historical horizon HH and the *bandwidth* σ. As *kernel function* we choose the exponential function $K = \exp(\cdot)$. Furthermore for our numerical experiment we only perform a prediction if there are at least 10 reference trajectories available. As reference trajectories we chose all headway trajectories of the 13th, 14th, 15th, 20th, 21st and 22nd of November 2012.

Figure 3.6 visualizes three headway trajectories beside the predicted values computed by a five-minutes-ahead prediction using kernel regression and extrapolation and the corresponding RMSE of our experiment. Table 3.3 shows the best 15 parameter configurations w.r.t. the RMSE for each prediction horizon PH corresponding to 5 and 10 minutes resulting from our numerical experiment. The last three columns contain the means of the percentages of predictions within correctness bins of one, two and three minutes among all predicted headway trajectories.

Again like in the case of linear regression and extrapolation we are able to compute the Euclidean distance between the regression trajectory, i.e., our weighted sum, and the true headway trajectory within the historical horizon $\{k-HH,\ldots,k\}$ and thus get a value that might help us in an operational setting to decide if we should trust the prediction or not.

PH [min]	**HH** [min]	σ	**w** [min]	$\overline{\textbf{RMSE}}$ [min]	$\overline{\textbf{err < 1 min}}$ [%]	$\overline{\textbf{err < 2 min}}$ [%]	$\overline{\textbf{err < 3 min}}$ [%]
5	3	0.3	0	1.15	67.48	90.55	97.78
5	3	0.3	3	1.23	63.60	88.58	97.21
5	3	0.3	6	1.31	62.57	85.23	96.29
5	3	1.0	0	1.17	65.60	90.36	97.41
5	3	1.0	3	1.27	62.87	86.65	96.65
5	3	3.0	0	1.21	64.02	89.07	97.45
5	5	0.3	0	1.20	64.63	90.47	97.16
5	5	0.3	3	1.27	62.41	87.96	96.59
5	5	1.0	0	1.21	63.50	89.45	97.09
5	5	1.0	3	1.31	62.58	85.07	96.53
5	5	3.0	0	1.26	62.37	86.18	97.07
5	10	0.3	0	1.23	63.40	88.78	97.03
5	10	0.3	3	1.30	60.95	85.61	96.39
5	10	1.0	0	1.29	61.95	85.76	96.54
5	15	0.3	0	1.29	60.47	86.01	96.57
10	3	0.3	0	1.74	50.72	76.30	88.47
10	3	0.3	3	1.80	49.11	74.38	87.18
10	3	1.0	0	1.71	52.64	74.74	87.56
10	3	1.0	3	1.80	51.36	73.46	86.30
10	3	3.0	0	1.74	51.57	74.31	87.37
10	3	3.0	3	1.83	49.24	72.76	86.08
10	5	0.3	0	1.77	48.33	74.50	87.47
10	5	0.3	3	1.83	46.88	72.87	86.94
10	5	1.0	0	1.75	51.45	74.07	87.69
10	5	1.0	3	1.83	49.80	72.63	85.98
10	5	3.0	0	1.79	50.04	72.92	86.56
10	10	0.3	0	1.77	47.85	74.08	87.65
10	10	0.3	3	1.83	44.07	71.50	86.52
10	10	1.0	0	1.80	48.35	72.58	86.68
10	15	1.0	0	1.83	47.28	70.10	85.75

Table 3.3: Results of the parameter study for the kernel-regression-and-extrapolation model (best 15 results w.r.t. RMSE for each prediction horizon *PH*)

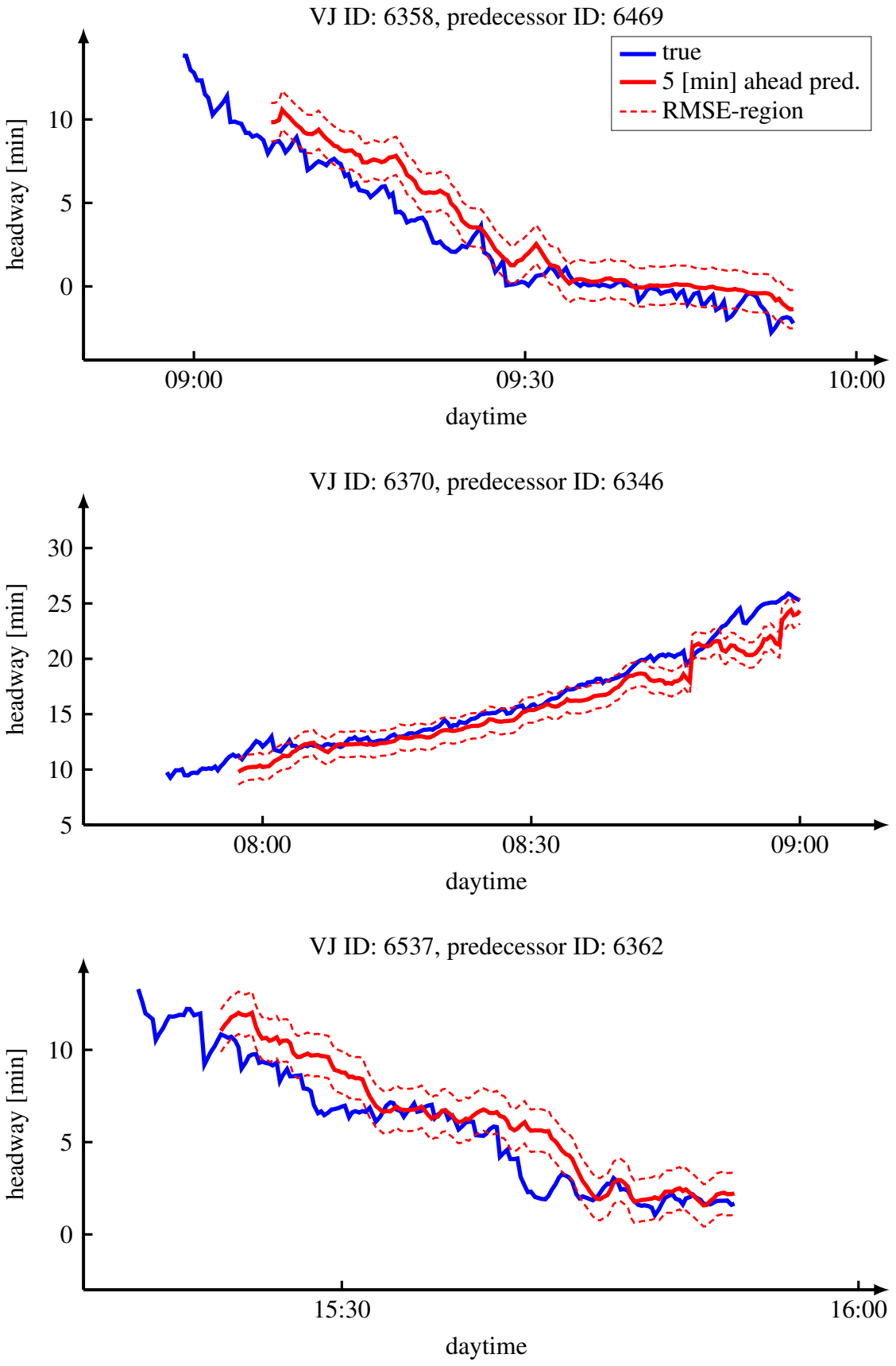


Figure 3.6: Illustration of the headway prediction using a kernel-regression-and-extrapolation model, 28th Nov 2012

3.5 Artificial Neural Network - Multilayer Perceptron

In this section we use *Multilayer Perceptrons (MLP)*, which are a special kind of *Artificial Feedforward Neural Networks*, to predict the headway trajectories. A brief description of multilayer perceptrons is given in the appendix B.1.3 and an illustration of a MLP is shown in figure 3.7. The descriptions of this section and further details on neural networks can also be found, for example, in [Kri07] or [Bis95].

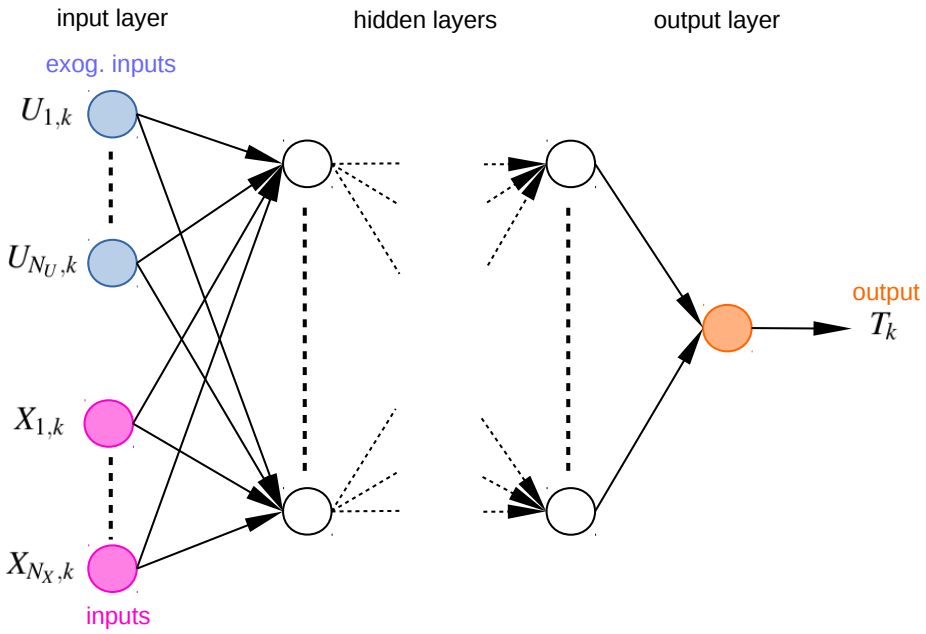


Figure 3.7: Multilayer perceptron (MLP) with exogenous inputs

MLPs are used in the field of time-series analysis (see [Cha00]) due to their ability to approximate unknown nonlinear functions. One very famous result out of the field of approximation theory is theorem 3.1, also known as *universal approximation theorem*, and can be found in [Cyb89].

Theorem 3.1 — Universal approximation theorem[a]. Let $\varphi(\cdot)$ be a non-constant, bounded and monotonically-increasing continuous function. Let I_m be a compact subset of $\mathbb{R}^m$. The space of continuous functions on I_m is denoted by $\mathscr{C}(I_m)$. Then, given any function $f \in \mathscr{C}(I_m)$ and $\varepsilon > 0$, there exists an integer N, real constants $v_i, b_i \in \mathbb{R}$ and

real vectors $w_i \in \mathbb{R}^m$, where $i = 1,\ldots,N$, such that we may define:

$$F(x) = \sum_{i=1}^{N} v_i \varphi\left(w_i^T x + b_i\right)$$

as an approximate realization of the function f where f is independent of φ; that is

$$|F(x) - f(x)| < \varepsilon$$

for all $x \in I_m$. In other words, functions of the form $F(x)$ are *dense* in $\mathscr{C}(I_m)$.

[a] formulation from: `https://en.wikipedia.org/wiki/Universal_approximation_theorem`

If we assume that the headway values can be approximated by a nonlinear function depending on previous headway data theorem 3.1 tells us that we are able to approximate this nonlinear behavior by a MLP with one hidden layer and thus are able to predict the headway trajectories in theory. The theorem does not tell us how to choose the weights and biases of the network. The process of adjusting the parameters of a network with a given structure of neurons and layers is called *training*. The training can be formulated as a *minimization problem*, where the objective function is a certain function measuring the distance between the predicted and the true values, depending on the weights and biases. One way to find a (at least local) minimum of the function would be a *steepest descent algorithm*. This idea is known as the *backpropagation algorithm* for feedforward networks (see, e.g., [Kri07]).

We use MATLAB 's *neural network toolbox* described in the appendix B.2, which uses the *Levenberg-Marquardt algorithm* (for details see, e.g., [MNT04] or [Bis95]) as a default solving algorithm. Even though this approach sounds very promising we would like to point out that it induces all problems that might occur in the classical steepest-descent-optimization framework. One very famous problem especially known from the training of *recurrent neural networks* and multilayer perceptrons with many layers is known as the *vanishing gradient* problem (see, e.g., [Hoc98]).

In here we use the neural network in a multi-steps-ahead prediction mode and do not make use of exogenous inputs, which are quantities that are used for the prediction of other quantities than themselves. We prepare a so called *input-output dataset* and divide this in a *training set*, *validation set* and *test set*. The inputs are stored in a matrix $X \in \mathbb{R}^{(HH+1)\times N}$ and the outputs are stored in a matrix $T \in \mathbb{R}^{1\times N}$. The kth column of X for some index k contains some point of a training headway trajectory h with time index $\hat{k}$ and its HH preceding points. The kth column of T contains the point of the headway trajectory PH

time steps ahead of time index $\hat{k}$:

$$\begin{aligned} X_{:,k} &= \begin{pmatrix} h_{\hat{k}-HH} \\ \vdots \\ h_{\hat{k}} \end{pmatrix}, \\ T_k &= h_{\hat{k}+PH}. \end{aligned} \tag{3.2}$$

This is illustrated in figure 3.8. We obtain the predicted headway value by evaluating the network function (see appendix B.1.3) at the vector containing the current headway value and its HH preceding values:

$$h'^{\text{MLP}}_{k+PH} = f_{\text{net}}\left(\begin{pmatrix} h_{k-HH} \\ \vdots \\ h_k \end{pmatrix}\right).$$

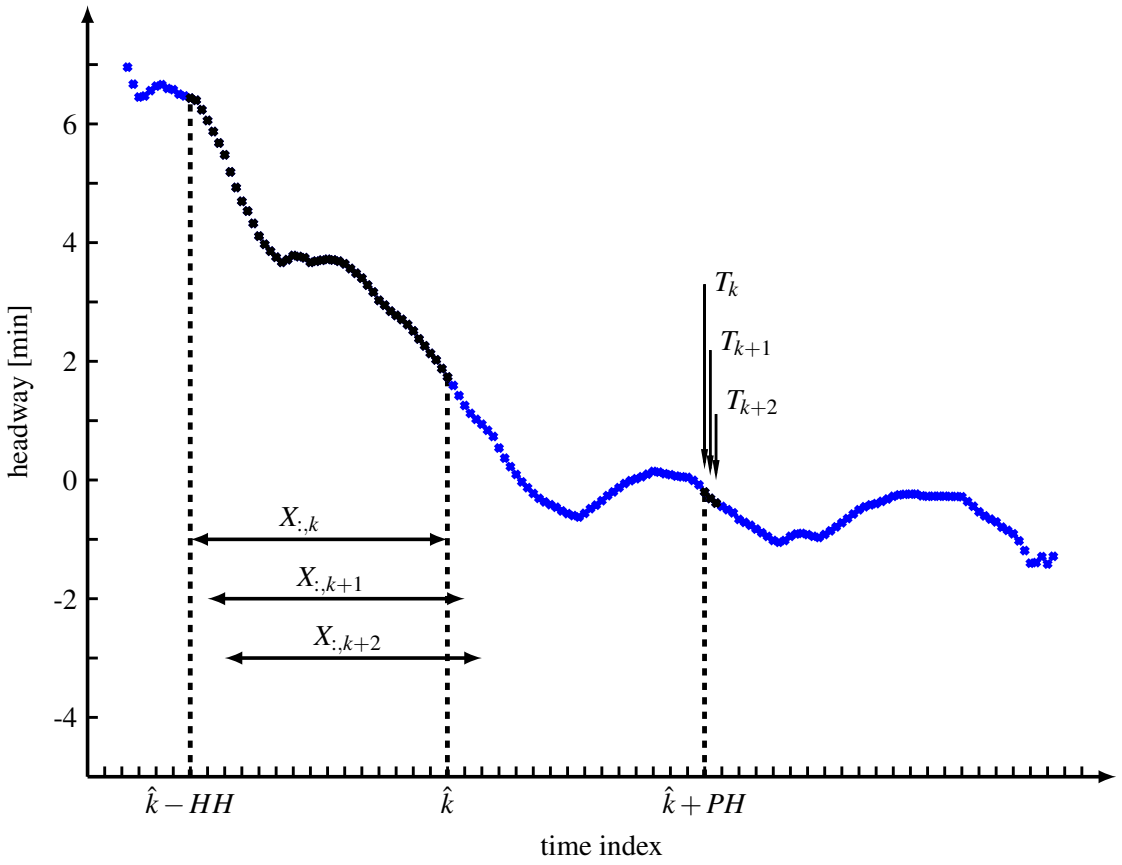


Figure 3.8: Illustration of the input-output data structure for the MLP approach

Figure 3.9 visualizes three headway trajectories beside the predicted values computed by a five-minutes-ahead prediction using a MLP and the corresponding RMSE of our

experiment. Table 3.4 shows the best 15 parameter configurations w.r.t. the RMSE for each prediction horizon *PH* corresponding to 5 and 10 minutes resulting from our numerical experiment. The last three columns contain the means of the percentages of predictions within correctness bins of one, two and three minutes among all predicted headway trajectories.

PH [min]	**HH** [min]	**net struct**	**w** [min]	$\overline{\textbf{RMSE}}$ [min]	$\overline{\textbf{err < 1 min}}$ [%]	$\overline{\textbf{err < 2 min}}$ [%]	$\overline{\textbf{err < 3 min}}$ [%]
5	3	10|10|10	0	1.12	68.30	92.02	98.13
5	3	20	0	1.11	67.70	92.50	98.21
5	5	10|10	0	1.11	67.11	91.69	98.25
5	5	10|10|10	0	1.11	66.93	91.51	98.38
5	5	10|5	0	1.11	69.35	91.37	98.18
5	5	20	0	1.11	68.63	91.90	98.25
5	5	20|10	0	1.12	67.88	91.84	98.25
5	5	30	0	1.11	68.79	91.24	98.19
5	5	5	0	1.11	68.31	91.63	98.38
5	10	10	0	1.12	68.15	92.22	97.90
5	10	10|10	0	1.10	68.38	92.11	97.98
5	10	20	0	1.12	68.25	92.18	98.05
5	10	20|10	0	1.11	68.75	92.25	98.05
5	10	30	0	1.10	68.90	92.07	97.97
5	10	5	0	1.12	67.36	91.79	97.76
10	3	10|10|10	0	1.63	52.08	76.98	89.73
10	3	10|5	3	1.68	50.94	75.23	89.52
10	5	10|10|10	0	1.67	51.46	75.88	90.47
10	5	10|5	3	1.67	51.31	75.73	89.06
10	5	5	0	1.66	50.78	74.98	90.16
10	5	5	3	1.68	49.38	75.65	89.67
10	10	10|10	0	1.64	50.77	75.91	90.84
10	10	10|10|10	0	1.66	50.97	75.67	90.60
10	10	10|5	3	1.67	50.00	76.44	89.98
10	10	30	0	1.66	51.19	75.25	89.80
10	10	5	0	1.66	48.93	76.76	90.90
10	15	10|10	0	1.68	48.39	74.44	89.57
10	15	10|5	3	1.64	49.72	75.72	90.54
10	15	20|10	0	1.67	48.59	73.46	91.25
10	15	5	0	1.63	51.44	76.78	91.02

Table 3.4: Results of the parameter study for the multilayer-perceptron model (best 15 results w.r.t. RMSE for each prediction horizon *PH*)

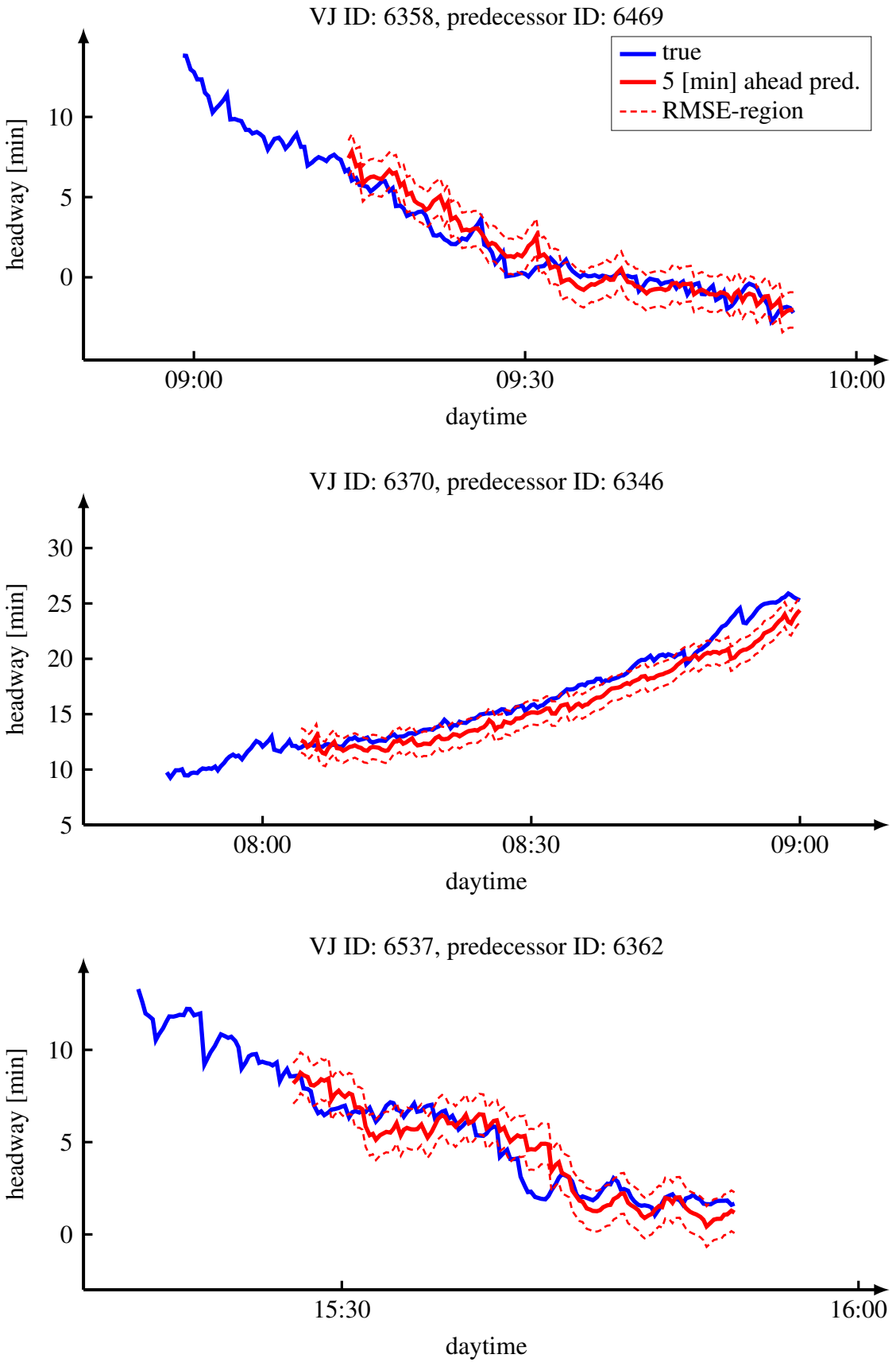


Figure 3.9: Illustration of the headway prediction using a multilayer perceptron, 28th Nov 2012

3.6 Autoregressive-Moving-Average Model (ARMA/ARMAX)

As a last model we present the *autoregressive-moving-average model*. The details for this can be found for example in [Cha00].

Definition 3.6.1 — ARMA/ARMAX model. 1. Let $h \in \mathbb{R}^{N_h}$ be a headway trajectory. An *autoregressive-moving-average (ARMA)* model with *autoregressive (AR) order* $p \in \mathbb{N}_{\geq 0}$ and *moving average (MA) order* $q \in \mathbb{N}_{\geq 0}$ is given by the real *AR coefficients* $a_1, \ldots, a_p$ and the real *MA coefficients* $c_1, \ldots, c_q$ and is for $k \in \{\max\{p,q\}, \ldots, N_h - 1\}$ based on the following equation:

$$\begin{aligned} h'^{\text{ARMA}}_{k+1} &= a_1 h_k + a_2 h_{k-1} + \ldots + a_p h_{k-p+1} \\ &\quad + c_1 \varepsilon_k + c_2 \varepsilon_{k-1} + \ldots + c_q \varepsilon_{k-q+1}, \\ \varepsilon_i &= \begin{cases} h_i - h'^{\text{ARMA}}_i & , \ i \in \{\max\{p,q\}, \ldots, N_h - 1\} \\ 0 & , \text{ else} \end{cases} . \end{aligned} \tag{3.3}$$

2. If we consider $N_U \in \mathbb{N}$ *exogenous inputs* and real coefficients $b_0, \ldots, b_{N_U - 1}$ the ARMA model can be extended to the *ARMAX* model. The basic equation (3.3) is for $k \in \{\max\{p,q,N_U\}, \ldots, N_h - 1\}$ changed to the following equation:

$$\begin{aligned} h'^{\text{ARMA}}_{k+1} &= a_1 h_k + a_2 h_{k-1} + \ldots + a_p h_{k-p+1} \\ &\quad + b_0 u_{k+1} + b_1 u_k + \ldots + b_{N_U - 1} u_{k-N_U+2} \\ &\quad + c_1 \varepsilon_k + c_2 \varepsilon_{k-1} + \ldots + c_q \varepsilon_{k-q+1}, \\ \varepsilon_i &= \begin{cases} h_i - h'^{\text{ARMA}}_i & , \ i \in \{\max\{p,q,N_U\}, \ldots, N_h - 1\} \\ 0 & , \text{ else} \end{cases} . \end{aligned}$$

3. (a) In case of $q = 0$ the ARMA/ARMAX model becomes an AR/ARX (autoregressive/autoregressive with exogenous inputs) model.
   (b) In case of $p = 0$ the ARMA model becomes a MA (moving average) model.

We obtain a prediction for a point *PH* time steps ahead of the current time by multiple single-step-ahead predictions like illustrated in figure 3.2.

For our numerical experiments we restrict ourselves to an AR model without exogenous inputs. The results are presented in table 3.5. For a sample set $(h_i)_{i=k-HH}^{k}$ the parameters

$a_1, \ldots, a_q$ are computed by solving the least-squares problem in the following equation:

$$(a_1, \ldots, a_p)^T = \underset{\hat{a} \in \mathbb{R}^p}{\operatorname{argmin}} \|A \cdot \hat{a} - b\|_2^2,$$

$$A = \begin{pmatrix} h_{k-1} & h_{k-2} & \ldots & h_{k-p} \\ h_{k-2} & h_{k-3} & \ldots & h_{k-p-1} \\ \vdots & \vdots & & \vdots \\ h_{k-HH+p-1} & h_{k-HH+p-2} & \ldots & h_{k-HH} \end{pmatrix}, \quad b = \begin{pmatrix} h_k \\ h_{k-1} \\ \vdots \\ h_{k-HH+p} \end{pmatrix}.$$

Since we fit the model parameters for each time index k to the data we would be able in an operational setting to compute online some confidence value for the prediction, at least for all time indices $k > \max\{p, q, N_U\}$. This confidence value could again be computed using the Euclidean distance between true headway values and the model outputs within the range of the historical horizon $\{k - HH + p, \ldots, k\}$.

Figure 3.10 visualizes three headway trajectories beside the predicted values computed by a five-minutes ahead prediction using an AR model and the corresponding RMSE of our experiment. Table 3.5 shows the best 15 parameter configurations w.r.t. the RMSE for each prediction horizon PH corresponding to 5 and 10 minutes resulting from our numerical experiment. The last three columns contain the means of the percentages of predictions within correctness bins of one, two and three minutes among all predicted headway trajectories.

PH [min]	HH [min]	p|q	w [min]	$\overline{\text{RMSE}}$ [min]	$\overline{\text{err < 1 min}}$ [%]	$\overline{\text{err < 2 min}}$ [%]	$\overline{\text{err < 3 min}}$ [%]
5	$\geq$ 10.0	10|0	3	1.18	65.49	89.47	97.37
5	$\geq$ 10.0	10|0	6	1.16	65.86	90.04	97.65
5	$\geq$ 10.0	4|0	0	1.18	65.24	89.60	97.16
5	$\geq$ 10.0	4|0	3	1.16	67.42	88.74	97.38
5	$\geq$ 10.0	4|0	6	1.15	66.58	90.16	97.74
5	$\geq$ 10.0	7|0	3	1.17	66.65	88.74	97.60
5	$\geq$ 10.0	7|0	6	1.15	66.37	90.56	97.97
5	$\geq$ 10.0	7|0	9	1.18	65.20	90.03	97.74
5	$\geq$ 13.0	13|0	6	1.18	64.20	89.31	97.45
5	$\geq$ 15.0	10|0	3	1.18	64.78	89.36	97.34
5	$\geq$ 15.0	10|0	6	1.16	65.15	89.75	97.63
5	$\geq$ 15.0	10|0	9	1.18	65.50	89.68	97.76
5	$\geq$ 15.0	4|0	3	1.19	66.21	87.80	97.02
5	$\geq$ 15.0	4|0	6	1.17	66.14	89.67	97.43
5	$\geq$ 15.0	7|0	6	1.16	65.75	90.10	97.76
10	$\geq$ 10.0	4|0	0	1.75	48.07	73.08	87.90
10	$\geq$ 10.0	4|0	3	1.79	50.07	72.91	85.68
10	$\geq$ 10.0	4|0	6	1.78	51.64	73.54	85.97
10	$\geq$ 10.0	4|0	9	1.82	48.37	72.79	86.73
10	$\geq$ 10.0	7|0	3	1.80	47.43	73.38	86.60
10	$\geq$ 10.0	7|0	6	1.82	48.51	72.43	86.75
10	$\geq$ 15.0	10|0	3	1.78	48.28	71.72	87.92
10	$\geq$ 15.0	10|0	6	1.80	47.55	71.32	86.53
10	$\geq$ 15.0	4|0	0	1.76	47.51	71.71	87.18
10	$\geq$ 15.0	4|0	3	1.82	48.70	70.97	84.55
10	$\geq$ 15.0	4|0	6	1.81	50.59	71.95	85.10
10	$\geq$ 15.0	4|0	9	1.83	47.93	71.02	86.13
10	$\geq$ 15.0	7|0	0	1.82	46.28	69.44	86.14
10	$\geq$ 15.0	7|0	3	1.83	47.40	71.29	85.65
10	$\geq$ 15.0	7|0	6	1.81	48.36	71.43	86.31

Table 3.5: Results of the parameter study for the AR approach (best 15 results w.r.t. RMSE for each prediction horizon *PH*)

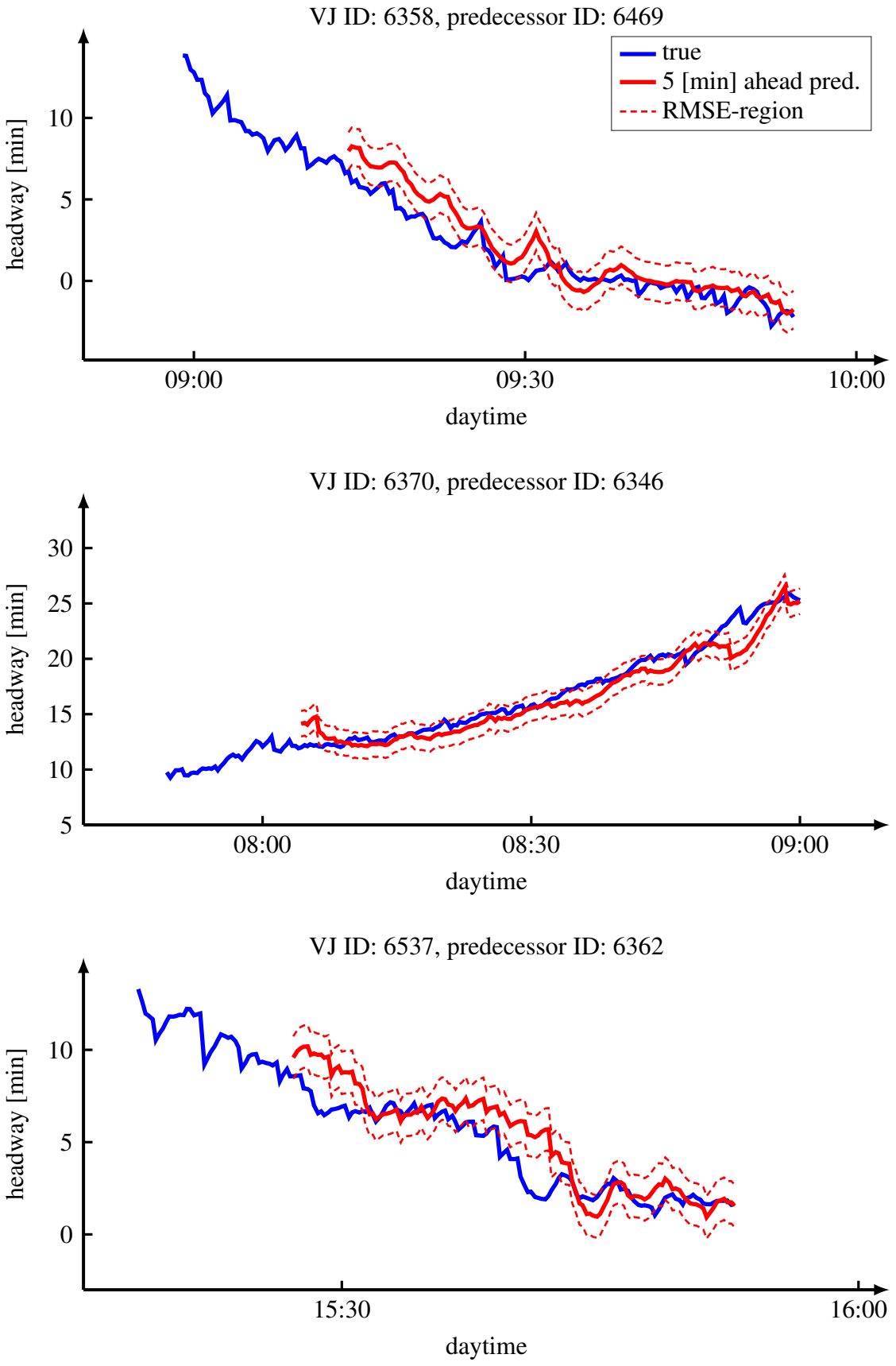


Figure 3.10: Illustration of the headway prediction using an AR model, 28th Nov 2012

3.7 Remarks on the Prediction Methods

In table 3.6 we compare the best results w.r.t. the RMSE for each method and each prediction horizon. We observe that the performances of all methods are promising and in the same range, even though the neural network gives for almost all performance measures the best results in our specific experiment. One advantage of the linear-regression-and-extrapolation, the kernel-regression-and-extrapolation and the ARMA model is that we are able to compute some confidence value of our prediction in the operational setting by computing, for example, the Euclidean distance between the model output and the true values within the historical horizon.

In chapter 4 we observe so called second moment effects, which describe the phenomenon that under certain assumptions small headways of a bus pair are followed by large headways of the succeeding bus pair and vice versa. This motivates to use the headway data of *neighboring buses* as exogenous inputs in order to improve the prediction. Other options for exogenous inputs could be information on the date like the *day of the week* or the *daytime*, *weather data*, *traffic information* like traffic light control, and later in the operational case the *control inputs* of our holding strategy which are presented in the next chapter.

Furthermore it will be necessary to investigate the *feedback effect of a data-driven predictive-control framework*: the headway prediction will influence the control actions and thus the dynamics of the bus route and its future headway trajectories, which will affect the headway prediction in the next step, etc.. Thus data-driven prediction used for control will bias the prediction.

PH [min]	**method**	**HH** [min]	**param**	**w** [min]	$\overline{\textbf{RMSE}}$ [min]	$\overline{\textbf{err}}$ **< 1 min** [%]	$\overline{\textbf{err}}$ **< 2 min** [%]	$\overline{\textbf{err}}$ **< 3 min** [%]
5	LR	3		9	1.17	65.57	90.18	97.59
5	KR	3	0.3	0	1.15	67.48	90.55	97.78
5	NN	10	10|10	0	1.10	68.38	92.11	97.98
5	ARMA	$\geq$ 10.0	7|0	6	1.15	66.37	90.56	97.97
10	LR	3		12	1.76	47.00	74.39	87.64
10	KR	3	1.0	0	1.71	52.64	74.74	87.56
10	NN	15	5	0	1.63	51.44	76.78	91.02
10	ARMA	$\geq$ 10.0	4|0	0	1.75	48.07	73.08	87.90

Table 3.6: Comparison of the best performances of the different methods

4. Bus-Route Model

In this chapter we present an analytical bus-route model. We define and discuss its stability properties and present a headway-based holding strategy to regularize the headways and to reduce bus bunching. The analytical model and the control framework using real-time data are taken and adapted from [Dag09]. Like suggested in [XAD11] we extend this model by including predicted headway data. Thereby, based on the previous chapters, we present a predictive-control framework to regularize the headways which is summarized in figure 4.1. We perform a simulation which shows the advantages of predictive control over the classical control that only uses directly available headway data.

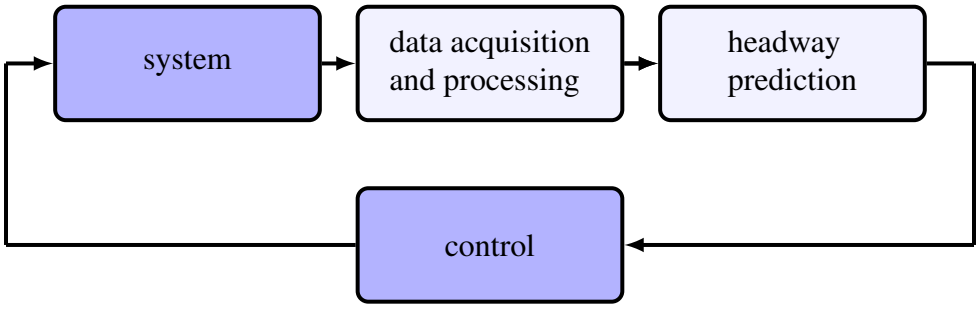


Figure 4.1: Predictive-control framework

4.1 Bus-Route Model

The *bus-route model* describes an artificial bus route and helps us to understand the headway dynamics and to design an appropriate control strategy. It is taken from [Dag09], including most of the notations and derivations.

We consider a bus route with N vehicle journeys per day, which we denote simply by 'buses'. We assume that all buses are operated on the same journey pattern with $N_S + 1$ stops (to avoid confusion: in chapter 2 we used the term N_S for the number of shape nodes, which in general does not coincide with the number of stops) and model the buses as single points on the shape without spatial extent. The first stop $s = 0$ is considered to be the bus depot where no passengers board and all buses start at scheduled times $t_{n,0} = t_0 + (n-1)h_0$. The headway h_0 is the *target headway*. We may also call it *equilibrium headway*, since in our deterministic model without external disturbances all headways are the same and it holds $h_{n,s} = h_0$ for all n and s (see proposition 4.1). In the interest of readability we index the buses by $n \in \{1,\dots,N\}$ and the stops by $s \in \{0,\dots,N_S\}$ and omit to mention the ranges every time.

The notations of the following are illustrated in figure 4.2.

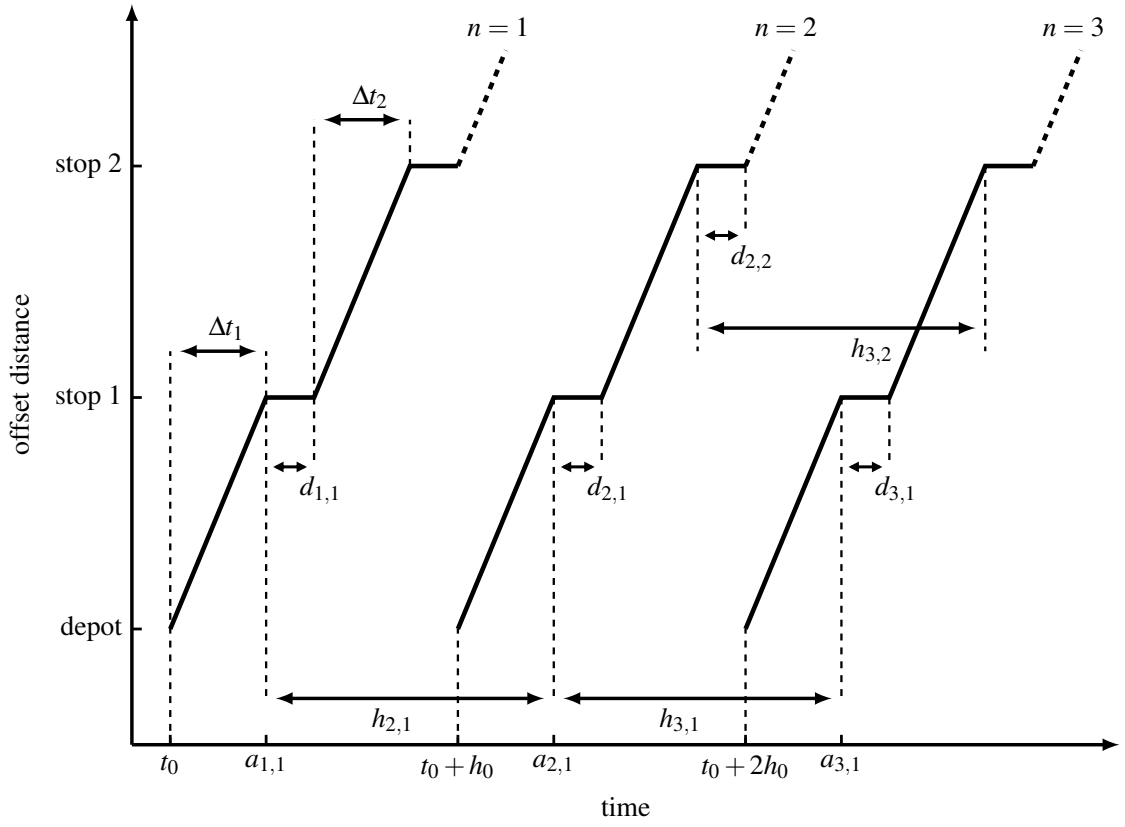


Figure 4.2: Notations of the bus-route model

A general *law of motion* of our bus-route model approximates the arrival time $a_{n,s+1}$ of bus n at stop $s+1$ by

$$\begin{aligned} a_{n,s+1} &= \max\left(\hat{a}_{n,s+1}, a_{n-1,s+1}\right), \quad \text{(no overtaking)} \\ \hat{a}_{n,s+1} &\approx \underbrace{t_0 + (n-1)h_0}_{\substack{\text{scheduled departure} \\ \text{time from stop 0} \\ \text{of bus } n}} + \underbrace{\sum_{i=1}^{s+1} p_{n,i}}_{\substack{\text{travel time from} \\ \text{stop 0 to stop } s+1}}, \\ p_{n,s} &= d_{n,s-1} + \Delta t_s + \nu_{n,s}, \\ d_{n,s} &= \hat{d}_{n,s} + u_{n,s}, \end{aligned} \tag{4.1}$$

where $p_{n,s}$ is the difference between the arrival time of bus n at stop $s-1$ and stop s, and $d_{n,s}$ is the dwell time of bus n staying at stop s, where we set $d_{n,0} = 0$ (not to be confused with the offset distance $d_n(t)$ of bus n at time t). The value $\nu_{n,s}$ is a noise term which describes the random disturbance of the difference $a_{n,s} - a_{n,s-1}$ of the expected arrival times at the stops $s-1$ and s of bus n due to varying traffic conditions and variations in passenger demand and is assumed to have mean value zero. We assume that the buses do not overtake each other, even if they bunched. The dwell time $d_{n,s} \geq 0$ consists of two components. The first component $\hat{d}_{n,s} \geq 0$ is the internal dwell time caused by passenger boardings as described below. The second component $u_{n,s} \geq 0$ describes the control input from our *dynamic holding strategy* including scheduled slack times, that we use to regularize the headways.

There are two kinds of holding strategies. A *static holding* strategy, also called *schedule-based holding* strategy, only introduces slack times into the schedule which define scheduled departure times for all buses and stops and rule that a bus should not depart from a stop earlier than the defined departure time, even if the boarding process might be finished already. A *dynamic holding* strategy also introduces additional slack times into the schedule, but allows to dispatch the bus from the stop flexibly depending on the current situation and thus enables to both decrease and increase the dwell times at stops. In here the scheduled slack times do not define earliest departure times, but fixed dwell times for the time after the boarding process is finished. This means that, once all passengers have boarded the bus, this one will stay at the stop for the length of the scheduled slack time regardless of current delays, unless the dynamic holding rule changes the corresponding dwell time. This second approach gives more flexibility for the control but also requires access to real-time data on the bus positions.
In the following we treat equation (4.1) as exact.

We call our model *deterministic* if we assume that there are no random disturbances in our model, i.e., it holds $\nu_{n,s} = 0$ for all n and s. We call our model *undisturbed* or *uncontrolled* if there are no interactions with the system beside the scheduled slack times

of the dynamic holding strategy, i.e., $u_{n,s} = d_s \geq 0$ for all n and s.

In the *deterministic model* we do not consider any changes in traffic conditions and assume that for all $s > 0$ the travel time between stop $s-1$ and s is always Δt_s for each bus. Thus the only component beside the input from our control strategy that slows down the buses on their journeys are passengers boarding and alighting at each stop.

We model the passenger demand in a continuous way like Daganzo in [Dag09] and [DP11]. We assume that $\hat{\beta}_s \in \mathbb{R}_{>0}$ passengers arrive at stop s per minute. We denote the boarding time per passenger by $\lambda \in \mathbb{R}_{>0}$. According to [Dag09] the time passengers need to alight from the bus is small compared to the time passengers need to enter the bus. Thus we only consider the dwell time caused by passenger boardings and model the *undisturbed internal dwell time* $\hat{d}_{n,s}$ of bus n at stop s as the product of the passenger arrival rate, the boarding time per passenger, and the corresponding headway:

$$\begin{aligned} \hat{d}_{n,s} &= \beta_s \cdot h_{n,s}, \quad s > 0, \\ \beta_s &= \hat{\beta}_s \cdot \lambda, \\ \hat{d}_{n,0} &= 0. \end{aligned} \tag{4.2}$$

Like in chapter 2 we define the stop-based headways as

$$\begin{aligned} h_{n,s} &= a_{n,s} - a_{n-1,s} \geq 0, \quad s > 0, 1 < n \leq N, \\ h_{n,0} &= h_0, \end{aligned}$$

and for the first bus we introduce the artificial headway $h_{1,s} = h_0$ for all s. This is equivalent to assuming a constant passenger demand for the first bus which causes a dwell time of $\hat{d}_{1,s} = \beta_s h_0$ for $s > 0$ and $\hat{d}_{1,0} = 0$, independent of the actual trajectory of the first bus. Since we assume that buses do not overtake each other all headways are non-negative.

We would like to point out that our formula for the passenger demand is based on the simplifying assumption that those passengers which arrive at stop s after the arrival of bus n will enter the next bus $n+1$. Since we consider the first stop $s = 0$ as the bus depot where buses depart at scheduled times we do not consider any dwell times there. A more detailed interpretation of the parameter β_s as the *expected number of passenger arrivals between stops s and* $s+1$ *during the delay induced by one boarding move* is given in [Dag09].

Formally we define bus bunching and the stability of the bus route in our model as follows.

Definition 4.1.1 — Stability. 1. We say that the buses $n-1$ and n are *bunched*, if for a certain $\theta_b \in \mathbb{R}_{\geq 0}$ and some stop s^* it holds

$$|h_{n,s^*}| \leq \theta_b.$$

2. We call a subset of buses of our bus-route model *stable*, if for a certain $\theta_{st} \in \mathbb{R}_{\geq 0}$ the deviation of the headways from the target headway h_0, which we denote by $\Delta h_{n,s} = h_{n,s} - h_0$, is bounded by θ_{st}, i.e.,

$$|\Delta h_{n,s}| \leq \theta_{st}$$

holds for all stops s and buses n of the subset. We call the stable situation also *equilibrium*.

In [Nai+14] the value θ_b is chosen to be one minute. For the following we choose $\theta_b = 0$.

The main reason why buses tend to bunch can be described as a *vicious cycle* like illustrated in figure 4.3.

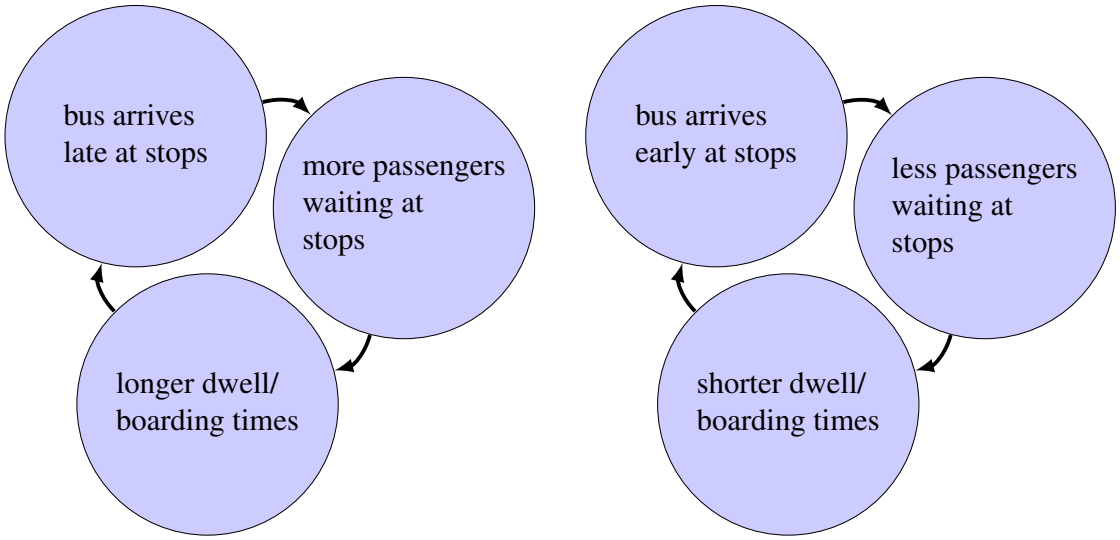


Figure 4.3: Bus bunching as vicious cycle

If we assume that, for example, bus $n-1$ is operated without any disturbances and the headway of bus n to bus $n-1$ at stop s is for some reason smaller than the equilibrium headway h_0, then less passengers are waiting for bus n at stop s. This leads to bus n staying less time at stop s and thus departing from there earlier than expected, which corresponds to the definition of the undisturbed internal dwell time in equation (4.2). As a result bus n catches up on bus $n-1$, what will further decrease the headway $h_{n,s+1}$ at the next stop.

Without any further disturbances this will repeat itself and will lead after a certain time to the bunching of the buses $n-1$ and n.

The impact of the headway deviations on the dwell times can also be seen by reformulating equation (4.2):

$$\hat{d}_{n,s} = \beta_s h_{n,s} = \underbrace{\beta_s h_0}_{\substack{\text{equilibrium dwell} \\ \text{time w.r.t.}\theta_{\text{st}}=0}} + \underbrace{\beta_s \left(h_{n,s} - h_0\right)}_{\substack{\text{dwell time depending} \\ \text{on headway deviation}}} .$$

We observe that the time to get from stop s to stop $s+1$ in the deterministic and uncontrolled model is approximately *affine-linear in* $h_{n,s}$, i.e.,

$$\underbrace{a_{n,s+1} - a_{n,s}}_{\substack{\text{travel time from} \\ \text{stop } s \text{ to stop } s+1}} = \Delta t_{s+1} + d_s + \beta_s h_{n,s} = \underbrace{\Delta t_{s+1} + \beta_s h_0 + d_s}_{\substack{\text{travel time without} \\ \text{disturbances}}} + \underbrace{\beta_s \left(h_{n,s} - h_0\right)}_{\substack{\text{dwell time depending} \\ \text{on headway deviation}}} , \tag{4.3}$$

where the parameter β_s models again the increase in bus delay arising from an unit increase in headway due to passenger boardings, like described by the vicious cycle.

We reformulate the law of motion (4.1) in terms of headways using equation (4.3) and get for $1 < n \leq N$ and all s

$$\begin{aligned} h_{n,s+1} &= \max\left(0, \hat{h}_{n,s+1}\right), \\ \hat{h}_{n,s+1} &= \hat{a}_{n,s+1} - \hat{a}_{n-1,s+1} \\ &= a_{n,s} + \beta_s h_0 + \Delta t_{s+1} + \beta_s \left(h_{n,s} - h_0\right) + u_{n,s} + \nu_{n,s+1} \\ &\quad - \left(a_{n-1,s} + \beta_s h_0 + \Delta t_{s+1} + \beta_s \left(h_{n-1,s} - h_0\right) + u_{n-1,s} + \nu_{n-1,s+1}\right) \\ &= \left(1+\beta_s\right) h_{n,s} - \beta_s h_{n-1,s} + u_{n,s} - u_{n-1,s} + \hat{\nu}_{n,s+1} \end{aligned} \tag{4.4}$$

with $\hat{\nu}_{n,s+1} = \nu_{n,s+1} - \nu_{n-1,s+1}$. For the following part we define $\hat{\nu}_{n,s} = 0$ for $n \leq 1, n > N$ and for all s.

The following proposition states the instability of the deterministic and uncontrolled bus-route model in case of a single deviation of an arbitrary headway from the target headway.

Proposition 4.1 — Instability. We consider our deterministic bus-route model, i.e., $\hat{\nu}_{n,s} = 0$ for all buses n and stops s. We choose for all stops s the control inputs $u_{n,s} = d_s \geq 0$ as the scheduled slack times for all buses n.

1. If there are no external disturbances and no further control inputs, i.e., $u_{n,s} = d_s$ for all n and s, then the bus route is in its equilibrium w.r.t. $\theta_{\text{st}} = 0$ and it holds $h_{n,s} = h_0$ for all n and s.
2. If under the same assumptions like in the first item two or more buses bunched w.r.t.

$\theta_b = 0$, they will stay bunched for the rest of the journey.

3. We consider the time that a bus $\hat{n}$ arrives at stop $\hat{s}$. We assume that until that time there have been no external disturbances of the bus route and thus no deviations of the headways $h_{n,s}$ from the target headway h_0, as far as a corresponding bus n already reached some stop s. Then, if we add [subtract] some $\varepsilon > 0$ to the dwell time $d_{\hat{n},\hat{s}}$ in equation (4.1) and do not disturb the system at any other point in time, the headways $(h_{n+1,s})_{s\geq\hat{s}}$ $\left[(h_{n,s})_{s\geq\hat{s}}\right]$ are monotonically decreasing in s and become zero at some point as far as N_S is sufficiently large, i.e., the buses $\hat{n}$ and $\hat{n}+1$ [$\hat{n}-1$ and $\hat{n}$] will bunch.
4. Under the same assumptions as for the previous item the following statement holds for two buses $n-1$ and n: if for some s_1, s_2 with $0 < s_1 < s_2$ it holds $0 < h_{n-1,s} < h_0$ for all s with $s_1 \leq s < s_2$ and $h_{n,s_1} \geq h_0$ then the finite subsequence $(h_{n,s})_{s=s_1}^{s_2}$ is strictly monotonically increasing. If $h_{n-1,s} > h_0$ holds for all $s_1 \leq s < s_2$ and $h_{n,s_1} \leq h_0$ then the finite subsequence $(h_{n,s})_{s=s_1}^{s_2}$ is strictly monotonically decreasing, unless the buses already bunched w.r.t. $\theta_b = 0$. We call this phenomenon *second moment effect*.

The third item of proposition 4.1 could be phrased in simpler terms: A single disturbance in the deterministic and otherwise undisturbed bus-route model will lead to bus bunching.
Also the fourth item has a simple interpretation: in the deterministic and from some point in time undisturbed bus-route model small headways of a bus pair are followed by large headways of the succeeding bus pair and vice versa, as far as bunching events do not change this.

Proof of proposition 4.1.

1. We assume $h_{n,s} = h_{n-1,s} = h_0$ for some s and n with $1 < n \leq N$. By equation (4.4) with $u_{n,s} = u_{n-1,s} = d_s$ and $v_{n,s+1} = 0$ it follows

$$\hat{h}_{n,s+1} = (1+\beta_s)\, h_{n,s} - \beta_s h_{n-1,s} = h_0.$$

   The statement follows by a nested induction over n and s since $h_{n,0} = h_0$ holds for all n and $h_{1,s} = h_0$ holds for all s by assumption.
2. We present the proof for a single bus pair that bunched. The statement for several bunched buses follows by the same argument. Let's assume that the buses $n-1$ and n bunched at some point and arrive at the same time at stop s, i.e., $a_{n,s} = a_{n-1,s}$. By

equation (4.1) we get for the deterministic and undisturbed case

$$\begin{aligned}\hat{a}_{n,s+1} &= a_{n,s} + \Delta t_{s+1} + \beta_s \cdot \underbrace{h_{n,s}}_{\substack{=a_{n,s}-a_{n-1,s}\\=0}} \\ &\leq \underbrace{a_{n,s}}_{=a_{n-1,s}} + \Delta t_{s+1} + \underbrace{\beta_s h_{n-1,s}}_{\geq 0} \\ &= \hat{a}_{n-1,s+1} \\ &\leq a_{n-1,s+1},\end{aligned}$$

and thus

$$a_{n,s+1} = \max\left(\hat{a}_{n,s+1}, a_{n-1,s+1}\right) = a_{n-1,s+1}.$$

This implies that the buses arrive at the next stop again at the same time. The statement follows now by induction over $\hat{s} \geq s$.

3. We prove the third item of proposition 4.1 only for the second case, i.e., we subtract some $\varepsilon > 0$ from the dwell time $d_{\hat{n},\hat{s}}$. The proof for the first case goes analogously. W.l.o.g. we assume that we have infinitely many stops, i.e., $N_S = \infty$ and for all s holds $\beta_s \geq \beta > 0$, and we consider the sequence $h = (h_{\hat{n},s})_{s \geq \hat{s}}$. As long as the buses $\hat{n}-1$ and $\hat{n}$ did not bunch it holds $h_{\hat{n},s} = \hat{h}_{\hat{n},s}$. By assumption it holds $h_{\hat{n},\hat{s}} = h_0$ and $h_{\hat{n},\hat{s}+1} = h_0 - \varepsilon$. We consider the deterministic case in equation (4.4), i.e., $\hat{v}_{n,s+1} = 0$, and get with $u_{n,s} = u_{n-1,s} = d_s$

$$\begin{aligned}\hat{h}_{\hat{n},\hat{s}+2} &\overset{(4.4)}{=} (1+\beta_{\hat{s}+1})\, h_{\hat{n},\hat{s}+1} - \beta_{\hat{s}+1} \underbrace{h_{\hat{n}-1,\hat{s}+1}}_{=h_0} \\ &= (1+\beta_{\hat{s}+1}) \underbrace{(h_{\hat{n},\hat{s}+1} - h_0)}_{=-\varepsilon} + h_0 \\ &\leq -(1+\beta)\,\varepsilon + h_0.\end{aligned}$$

With the same argument for $\hat{s}+i$ with $i > 1$

$$\hat{h}_{\hat{n},\hat{s}+i} \leq -(1+\beta)^{i-1}\,\varepsilon + h_0,$$

follows by induction. We observe that the differences between consecutive auxiliary headways

$$\hat{h}_{\hat{n},\hat{s}+i} - \hat{h}_{\hat{n},\hat{s}+i-1} = -\beta\,(1+\beta)^{i-2}\,\varepsilon < -\beta\varepsilon < 0, \quad \text{for all } i > 1,$$

are always smaller than $-\beta\varepsilon < 0$, which implies that there exists some $i \geq 0$ such that $\hat{h}_{\hat{n},\hat{s}+i} < 0$. The statement follows now with $h_{\hat{n},\hat{s}+i} = \max\left(0, \hat{h}_{\hat{n},\hat{s}+i}\right)$.

4. We show the statement only for the first case. We get from equation (4.4) with $u_{n,s} = u_{n-1,s} = d_s$ and $\hat{v}_{n,s+1} = 0$

$$\begin{aligned} h_{n,s_1+1} = \max\left(0, \hat{h}_{n,s_1+1}\right) &\geq \hat{h}_{n,s_1+1} \\ &= (1+\beta_{s_1})\, h_{n,s_1} - \beta_{s_1} h_{n-1,s_1} \\ &= h_{n,s_1} + \beta_{s_1} \left(\underbrace{h_{n,s_1} - h_0}_{\geq 0} - \underbrace{(h_{n-1,s_1} - h_0)}_{<0} \right) \\ &> h_{n,s_1} \\ &\geq h_0. \end{aligned}$$

The statement follows now with the same argument for $s_1 + i$ by induction over i, for $i \leq (s_2 - s_1)$.

■

We illustrate the instability result of proposition 4.1 in figure 4.4, which shows the space-time diagram and the continuous-time headways (see definition 2.2.1) of our simulation of the deterministic model in the corresponding colors. For the simulation we consider a bus route with $N_S + 1 = 51$ equidistant stops with constant boarding coefficient $\beta_s = 1$ and zero scheduled slack times $d_s = 0$ for all s and a target headway $h_0 = 5$ [min]. We simulate the run of $N = 8$ buses. All buses depart with equal headways like in equation (4.1), except the first bus which departs from the first stop with an additional delay of $u_{1,0} = \varepsilon = 5$ [sec]. We do not disturb the dwell times $d_{n,s}$ for any other bus n or any stop s.

We observe the last three statements of proposition 4.1: once two (or more) buses bunched, they will stay together until the end of their journeys. Furthermore we observe that all occurring bunching events are just caused by one additional delay of one bus at one stop. At last we observe the second moment effects, i.e., small headways of a bus pair are followed by large headways of the succeeding bus pair and vice versa as far as no bunching events change the dynamics of the headway trajectories.

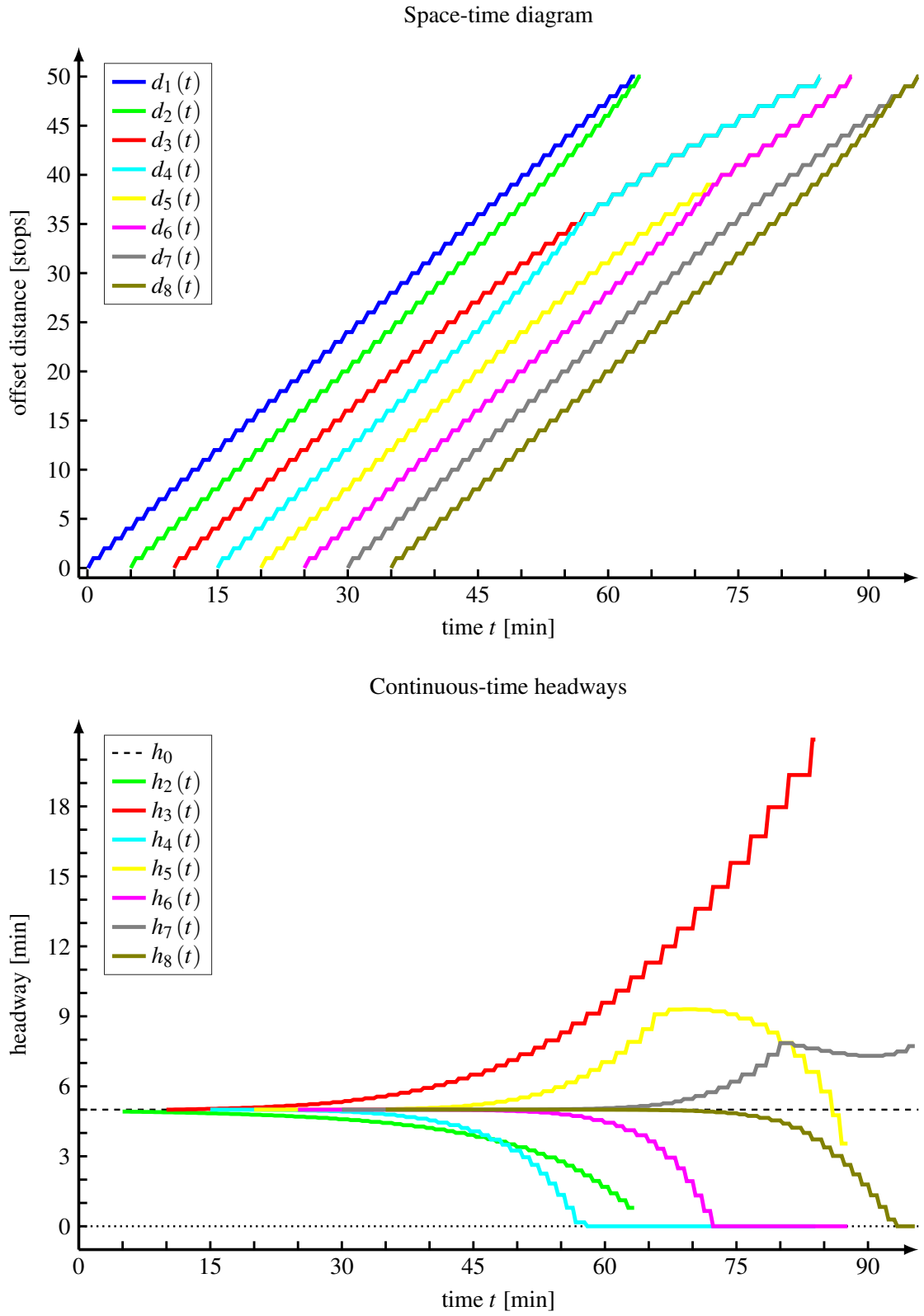


Figure 4.4: Illustration of proposition 4.1: Instability of headway trajectories

4.2 Control

In this section we present a dynamic headway-based holding framework introduced by Daganzo in [Dag09] to reduce the headway deviation and avoid bunching. This gives us one possible rule for choosing the inputs $u_{n,s}$ in equation (4.1).

The control strategy aims to reduce the headway deviation from the target headway, thus it is necessary to increase the headways if they are too small and to decrease the headways if they are too large. To this there are multiple different control strategies such as *holding, stop skipping, short turns, traffic light control*, etc. as described in section 1.2.

Decreasing the headway by speeding up the succeeding bus is only possible if we operate the buses as default mode not with their maximum cruising speed or if we introduce scheduled slack times at certain stops, i.e., additional scheduled dwell times which allow to dispatch a bus from a stop earlier if needed. In our model we assume that buses are operated with their maximum cruising speed and use a headway-based dynamic holding strategy as control.

We would like to point out that using a dynamic holding strategy and thus introducing a scheduled slack time at each stop slows down the scheduled commercial speed, which is the average speed of the bus over several stops, and increases the scheduled travel time of each vehicle journey which might come along with a lower service quality and higher costs for the service provider.

According to [FM09] and [MC84] passengers arrive randomly distributed at bus stops in case that the scheduled headways are smaller than 11-13 minutes. This means that the passengers do not rely on any scheduled arrival times but on the fact that buses arrive in the ideal case every 11-13 minutes. For this reason we do not try to achieve schedule adherence by our control method but evenly distributed headways.

In [Dag09] Daganzo assumes that $a_{n,s} = \hat{a}_{n,s}$ and thus $h_{n,s} = \hat{h}_{n,s}$. This simplified model coincides with our original model in equation (4.1) as long as there are no bunching events. So in case we could show that our control avoids buses from bunching in the simplified model, this would also hold for our original model. So to simplify the analysis and the notation and to avoid several case distinctions we also assume $a_{n,s} = \hat{a}_{n,s}$ and $h_{n,s} = \hat{h}_{n,s}$ in the following.

4.2.1 Forward-Headway Control

A very simple realization of the holding strategy is called *forward-headway control* and is expressed in equation (4.5) in terms of the dwell time of bus n at stop s, depending on the

forward headway $h_{n,s}$

$$\begin{aligned} u_{n,s} &= \max(0, \hat{u}_{n,s}), \\ \hat{u}_{n,s} &= \underbrace{d_s}_{\substack{\text{scheduled} \\ \text{slack time}}} + \gamma_s \cdot \underbrace{(h_0 - h_{n,s})}_{\substack{\text{negative} \\ \text{headway deviation}}}, \end{aligned} \tag{4.5}$$

where d_s denotes the scheduled slack time for each bus at stop s, i.e., once all passengers have boarded the bus will wait for an additional time d_s. In order to simplify the following analysis and to avoid several case distinctions we assume that the headways $h_{n,s}$ are bounded by some constant $M_h \in \mathbb{R}_{>0}$ and that the scheduled slack times are chosen as $d_s \geq \gamma_s (M_h - h_0)$ for all s, such that we get $u_{n,s} = \hat{u}_{n,s}$ (like in [Dag09]). With this assumption we get the controlled dwell time as

$$\begin{aligned} d_{n,s} &= \hat{d}_{n,s} + u_{n,s} \\ &= \underbrace{\beta_s h_0 + \beta_s (h_{n,s} - h_0)}_{\substack{\text{dwell time due to} \\ \text{passenger boardings}}} + \underbrace{d_s + \gamma_s (h_0 - h_{n,s})}_{\text{holding strategy}}. \end{aligned} \tag{4.6}$$

The term $\beta_s h_0 + d_s$ in equation (4.6) can be seen as the *target dwell time*. In [Dag09] it is shown that for stability reasons we should choose $\gamma_s \in (\beta_s, 1 + \beta_s)$, i.e., we rewrite $\gamma_s = \alpha + \beta_s$ for some $\alpha \in (0,1)$. The motivation for this is the following: if the dwell time at stop s differs from the target dwell time this might increase the headway deviation at the following stops as shown in proposition 4.1. Thus our control term should at least eliminate the current deviation from the target dwell time by adding $\beta_s (h_0 - h_{n,s})$, where the remaining part $\alpha (h_0 - h_{n,s})$ can be seen as the term of a classical *P*-controller (see [Son13] for details on control theory), which is expected to reduce the deviations of the headways from the target headway at the following stops. Since we interpret α later on as an entry of a discrete probability distribution function, which we use to prove the stability of this approach, we also require $\alpha \in (0,1)$.
We observe that all variables and constants on the right-hand side of equation (4.6) are known at the time we need to evaluate the equation, namely when bus n arrived at stop s.

We get the law of motion for the forward-headway-controlled model as follows:

$$\begin{aligned} a_{n,s+1} &= a_{n,s} + \Delta t_{s+1} + d_{n,s} + v_{n,s+1} \\ &= a_{n,s} + \Delta t_{s+1} + \beta_s h_0 + \beta_s (h_{n,s} - h_0) + d_s + (\alpha + \beta_s)(h_0 - h_{n,s}) + v_{n,s+1} \\ &= a_{n,s} + \Delta t_{s+1} + \beta_s h_0 + d_s + \alpha (h_0 - h_{n,s}) + v_{n,s+1}. \end{aligned}$$

The following equation for the headways holds similar to equation (4.4) for $1 < n \leq N$:

$$\begin{aligned} h_{n,s+1} &= a_{n,s+1} - a_{n-1,s+1} \\ &\overset{(4.4)}{=} (1+\beta_s)\, h_{n,s} - \beta_s h_{n-1,s} + u_{n,s} - u_{n-1,s} + \hat{v}_{n,s+1} \\ &= (1+\beta_s)\, h_{n,s} - \beta_s h_{n-1,s} + (\alpha+\beta_s)\,(h_{n-1,s} - h_{n,s}) + \hat{v}_{n,s+1} \\ &= (1-\alpha)\, h_{n,s} + \alpha h_{n-1,s} + \hat{v}_{n,s+1}, \end{aligned} \tag{4.7}$$

with $\hat{v}_{n,s+1} = v_{n,s+1} - v_{n-1,s+1}$ as before.

We would like to prove that our method leads under certain assumptions to a bounded headway deviation, where the bound depends on the corresponding stop.

First we reformulate equation (4.7) in terms of headway deviations $\Delta h_{n,s} = h_{n,s} - h_0$. By subtracting h_0 on both sides of equation (4.7) we get for $1 < n \leq N$

$$\Delta h_{n,s+1} = (1-\alpha)\,\Delta h_{n,s} + \alpha \Delta h_{n-1,s} + \hat{v}_{n,s+1}. \tag{4.8}$$

We define for $n < 1$ and $n > N$ artificial headways $h_{n,s} = h_0$ and artificial noise terms $v_{n,s} = 0$ and we obtain including the assumptions on the first bus the *boundary conditions* of equation (4.8)

$$\Delta h_{n,s} = 0, \quad \text{for all } s, n \leq 1, n > N. \tag{4.9}$$

The idea of the following is to formulate equation (4.8) as a convolution of the headway deviations with a vector $\vec{f}$ which contains the information on the control parameters. In order to include the boundary conditions we need to introduce a few technical terms. We express the convolution by the operator F and the consideration of the boundary conditions by the operator B. We denote the components of any vector $\vec{x}$ by $(\vec{x})_n = x_n$, unless stated otherwise, and define the operator B as follows:

$$\begin{aligned} &B\colon l^\infty(\mathbb{R}) \to l^\infty(\mathbb{R}), \\ &\quad \vec{x} \mapsto B(\vec{x}), \\ &\text{with } (B(\vec{x}))_n = \begin{cases} x_n & , 1 < n \leq N \\ 0 & , \text{else} \end{cases}, \end{aligned} \tag{4.10}$$

where $l^\infty(\mathbb{R})$ is the set of all real valued bounded sequences, where the indices of sequence entries range over all integer values.

For some $\vec{f} \in l^\infty(\mathbb{R}_{\geq 0})$ with $\sum_{i=-\infty}^{\infty} f_i = 1$ the convolution operator F is defined as

follows (see, e.g., [Bro+12]):

$$\begin{aligned} &F\colon l^\infty(\mathbb{R}) \to l^\infty(\mathbb{R}), \\ &\quad \vec{x} \mapsto F(\vec{x}) = \vec{f} * \vec{x}, \\ &\text{with } (F(\vec{x}))_n = \sum_{i=-\infty}^{\infty} f_{n-i} x_i. \end{aligned} \tag{4.11}$$

We can interpret the vector $\vec{f}$ as a probability mass function (p.m.f.) of a discrete random variable.

We use these operators to reformulate equation (4.8). For this we need the following lemma 4.2.

Lemma 4.2 1. The operators B from equation (4.10) and F from equation (4.11) are *linear*.

2. We define the operator

$$\begin{aligned} &BF\colon l^\infty(\mathbb{R}) \to l^\infty(\mathbb{R}), \\ &\quad \vec{x} \mapsto B \circ F(\vec{x}) = B(F(\vec{x})). \end{aligned}$$

We denote the jth application of the operator BF to itself by

$$\begin{aligned} &BF^j\colon l^\infty(\mathbb{R}) \to l^\infty(\mathbb{R}), \\ &\quad \vec{x} \mapsto BF^j(\vec{x}) = \underbrace{BF \circ \cdots \circ BF}_{j-\text{times}}(\vec{x}). \end{aligned}$$

   (a) The operator BF is linear.

   (b) If for some $M \in \mathbb{R}_{>0}$ it holds $|x_n| \leq M$ for all n then we get for all $j \geq 0$

$$\left|\left(BF^j(\vec{x})\right)_n\right| \leq M. \tag{4.12}$$

   (c) For the zero vector $\vec{0} \in l^\infty(\mathbb{R})$ it holds for all $j \geq 0$

$$BF^j\left(\vec{0}\right) = 0. \tag{4.13}$$

Proof. 1. Let $\vec{x}, \vec{y} \in l^\infty(\mathbb{R})$ and $\alpha_x, \alpha_y \in \mathbb{R}$.

- The operator B is well defined. For some n with $1 < n \leq N$ we get

$$\begin{aligned} \left(B(\alpha_x \vec{x} + \alpha_y \vec{y})\right)_n &= \alpha_x x_n + \alpha_y y_n \\ &= \alpha_x (B(\vec{x}))_n + \alpha_y (B(\vec{y}))_n \end{aligned}$$

and for any other n with $n \leq 1$ or $n > N$ we get

$$\begin{aligned}(B(\alpha_x\vec{x}+\alpha_y\vec{y}))_n &= 0\\ &= \alpha_x 0 + \alpha_y 0\\ &= \alpha_x (B(\vec{x}))_n + \alpha_y (B(\vec{y}))_n,\end{aligned}$$

what proves the linearity of B.

- By definition of $\vec{x}$ there exists some number $M \in \mathbb{R}_{\geq 0}$ such that $|x_n| \leq M$ holds for all integer n. The operator F is well defined since for any $L \in \mathbb{N}$ and any integer n it holds

$$\left|\sum_{i=-L}^{L} f_{n-i}x_i\right| \leq \sum_{i=-L}^{L} f_{n-i}\underbrace{|x_i|}_{\leq M} \leq M\underbrace{\sum_{i=-L}^{L} f_{n-i}}_{\leq 1} \leq M.$$

Thus the term $(f*g)_n$ in formula (4.11) is bounded by M for all n and thus defines an element in $l^\infty(\mathbb{R})$. The linearity of the convolution operator F is proved by the following equation:

$$\begin{aligned}(F(\alpha_x\vec{x}+\alpha_y\vec{y}))_n &= \sum_{i=-\infty}^{\infty} f_{n-i}(\alpha_x\vec{x}+\alpha_y\vec{y})_i\\ &= \alpha_x\sum_{i=-\infty}^{\infty} f_{n-i}x_i + \alpha_y\sum_{i=-\infty}^{\infty} f_{n-i}y_i\\ &= \alpha_x (F(\vec{x}))_n + \alpha_y (F(\vec{y}))_n.\end{aligned}$$

2. The linearity of BF follows by the linearity of the operators B and F.
   If we assume for some $j \geq 0$ that $\left|\left(BF^j(\vec{x})\right)_n\right| \leq M$ holds for all n we get by use of

the triangle inequality

$$\begin{aligned}
\left|\left(BF^{j+1}\left(\vec{x}\right)\right)_n\right| &= \left\{ \begin{array}{ll} \left|\sum_{i=-\infty}^{\infty} f_{n-i}\left(BF^{j}\left(\vec{x}\right)\right)_i\right| & , \ 1 < n \leq N \\ 0 & , \text{ else} \end{array} \right\} \\
&\leq \left\{ \begin{array}{ll} \sum_{i=-\infty}^{\infty} |f_{n-i}| \cdot \underbrace{\left|\left(BF^{j}\left(\vec{x}\right)\right)_i\right|}_{\leq M} & , \ 1 < n \leq N \\ 0 & , \text{ else} \end{array} \right\} \\
&\leq \left\{ \begin{array}{ll} M \underbrace{\sum_{i=-\infty}^{\infty} |f_{n-i}|}_{=1} & , \ 1 < n \leq N \\ 0 & , \text{ else} \end{array} \right\} \\
&\leq M.
\end{aligned} \tag{4.14}$$

The statement follows by inequality (4.14) and induction over j since $\left|\left(BF^{0}\left(\vec{x}\right)\right)_n\right| = |x_n| \leq M$ holds by assumption.

3. The statement follows for all $j \in \mathbb{N}_0$ by the linearity of the operator BF^j, which is linear as the jth composition of the linear operator BF.

■

Using the boundary conditions in (4.9) we are able to rewrite equation (4.8) for $1 < n \leq N$ as a convolution

$$\Delta h_{n,s+1} = \sum_{i=-\infty}^{\infty} f_{n-i}\Delta h_{i,s} + \hat{v}_{n,s+1} = \left(F\left(\vec{\Delta h}_s\right)\right)_n + \hat{v}_{n,s+1},$$

with $\vec{\Delta h}_s = \begin{pmatrix} \cdots & 0 & \Delta h_{2,s} & \cdots & \Delta h_{N,s} & 0 & \cdots \end{pmatrix}^T$. The operator BF allows us to write this equation for all n in a vectorized and more compact form as

$$\begin{aligned}
\vec{\Delta h}_{s+1} &= \begin{pmatrix} \cdots & 0 & \Delta h_{2,s+1} & \cdots & \Delta h_{N,s+1} & 0 & \cdots \end{pmatrix}^T \\
&= B\left(F\left(\vec{\Delta h}_s\right) + \vec{\hat{v}}_{s+1}\right) \\
&= BF\left(\vec{\Delta h}_s\right) + \vec{\hat{v}}_{s+1},
\end{aligned} \tag{4.15}$$

with $\vec{\hat{v}}_{s+1} = \begin{pmatrix} \cdots & 0 & \hat{v}_{2,s+1} & \cdots & \hat{v}_{N,s+1} & 0 & \cdots \end{pmatrix}^T = B\left(\vec{\hat{v}}_{s+1}\right)$. In the case of our forward-headway control we choose $f_0 = (1-\alpha), f_1 = \alpha$ and $f_i = 0$ for all other integers i. The vector $\vec{f}$ is allowed to have the form of a probability mass function of a non-negative random variable with positive mean value. In [Dag09] it is stated that this case arises

if, instead of only using the headway $h_{n,s}$ to compute a control output, we also include the information of other headways of preceding bus pairs at stop s such that we extend equation (4.6) for all n to

$$d_{n,s} = \hat{d}_{n,s} + d_s + (F_0 + \beta_s)(h_0 - h_{n,s}) + \sum_{i=1}^{\infty} F_i (h_0 - h_{n-i,s}),$$

where $\vec{F} \in l^\infty$ with $F_i = \sum_{m=i+1}^{\infty} f_m$ is the *complementary cumulative distribution function* of $\vec{f}$.

Next we would like to present the stability result from [Dag09]. For this we plug in the right-hand side of equation (4.15) in itself and get by the linearity of the operator BF (see lemma 4.2)

$$\begin{aligned}
\vec{\Delta h}_{s+1} &= BF\left(\vec{\Delta h}_s\right) + \vec{\hat{v}}_{s+1} \\
&= BF\left(BF\left(\vec{\Delta h}_{s-1}\right) + \vec{\hat{v}}_s\right) + \vec{\hat{v}}_{s+1} \\
&= BF\left(BF\left(\vec{\Delta h}_{s-1}\right)\right) + BF\left(\vec{\hat{v}}_s\right) + \vec{\hat{v}}_{s+1}.
\end{aligned}$$

Induction leads to

$$\vec{\Delta h}_{s+1} = BF^{s+1}\left(\vec{\Delta h}_0\right) + BF^s\left(\vec{\hat{v}}_1\right) + BF^{s-1}\left(\vec{\hat{v}}_2\right) + \ldots + BF^1\left(\vec{\hat{v}}_s\right) + BF^0\left(\vec{\hat{v}}_{s+1}\right).$$

By assumption we have $h_{n,0} = h_0$ for both real and artificial headways and thus $\vec{\Delta h}_0 = \vec{0}$ and get due to equation (4.13)

$$\vec{\Delta h}_{s+1} = \sum_{j=0}^{s} BF^j\left(\vec{\hat{v}}_{s+1-j}\right), \tag{4.16}$$

which gives us for each component

$$\Delta h_{n,s+1} = \sum_{j=0}^{s} \left(BF^j\left(\vec{\hat{v}}_{s+1-j}\right)\right)_n. \tag{4.17}$$

We would like to repeat the proof of Daganzo in [Dag09] for arrival times for our headway formulation.

Proposition 4.3 — Stability. If for some bound $M \in \mathbb{R}_{>0}$ it holds $|v_{n,s}| \leq M$ for all n and s, then for all n and s the headway deviations $\Delta h_{n,s}$ are bounded by $2Ms$ under the assumptions of this section.

Proof. The proof is adapted from [Dag09]. By assumption it holds for all n and s

$$|\hat{v}_{n,s}| = \left\{ \begin{array}{cc} \left|v_{n,s} - v_{n-1,s}\right| & , \, 1 < n \leq N \\ 0 & , \text{ else} \end{array} \right\} \leq |v_{n,s}| + \left|v_{n-1,s}\right| \leq 2M. \tag{4.18}$$

The statement follows by taking absolute values in equation (4.17), equations (4.12) and (4.18) and the triangle inequality:

$$\begin{aligned} \left|\Delta h_{n,s+1}\right| &\overset{(4.17)}{=} \left| \sum_{j=0}^{s} \left(BF^j \left(\vec{\hat{v}}_{s+1-j} \right) \right)_n \right| \\ &\leq \sum_{j=0}^{s} \left| \left(BF^j \left(\vec{\hat{v}}_{s+1-j} \right) \right)_n \right| \\ &\underset{(4.18)}{\overset{(4.12)}{\leq}} \sum_{j=0}^{s} 2M \\ &\leq 2M\,(s+1)\,. \end{aligned}$$

■

4.2.2 Forward-Backward-Headway Control

Even though the forward-headway control is very promising, we would like to point to the following problem: we consider three succeeding buses $n-1$, n and $n+1$. We assume that for some reason bus $n-1$ arrived early at stop s but bus n arrived on time such that $h_{n,s} > h_0$ holds. According to the forward-headway-control rule in equation (4.6) we would dispatch bus n at stop s earlier to decrease the headway $h_{n,s+1}$ at the next stop. But with this approach we do not consider the impact of this control action on the succeeding bus $n+1$. If we assume for example that bus $n+1$ left the previous stops later than scheduled, for instance, caused by an unexpected higher passenger demand, such that $h_{n+1,s} > h_{n,s} > h_0$ holds, the early dispatch of bus n would further increase the headway $h_{n+1,s}$ additional to the 'natural' increase due to the headway dynamics given by equation (4.4). This motivates us not only to include the *forward headway* $h_{n,s}$ but also the *backward headway* $h_{n+1,s}$. This approach has been introduced by Xuan and Daganzo in [XAD11]. The idea can be illustrated like in figure 4.5. We imagine that consecutive buses are connected by 'springs', which are relaxed when the corresponding headways are equal to the target headway. If we assume that $h_{n,s} < h_0$, which is equivalent to a shorter time distance between the buses $n-1$ and n, we could imagine the spring between the two buses to be compressed and creating a repelling force on the buses, which means that bus $n-1$ gets accelerated and bus n gets delayed. Vice versa if we assume $h_{n,s} > h_0$ this would correspond to the spring being stretched which will delay bus $n-1$ and accelerate bus n. Compared to the

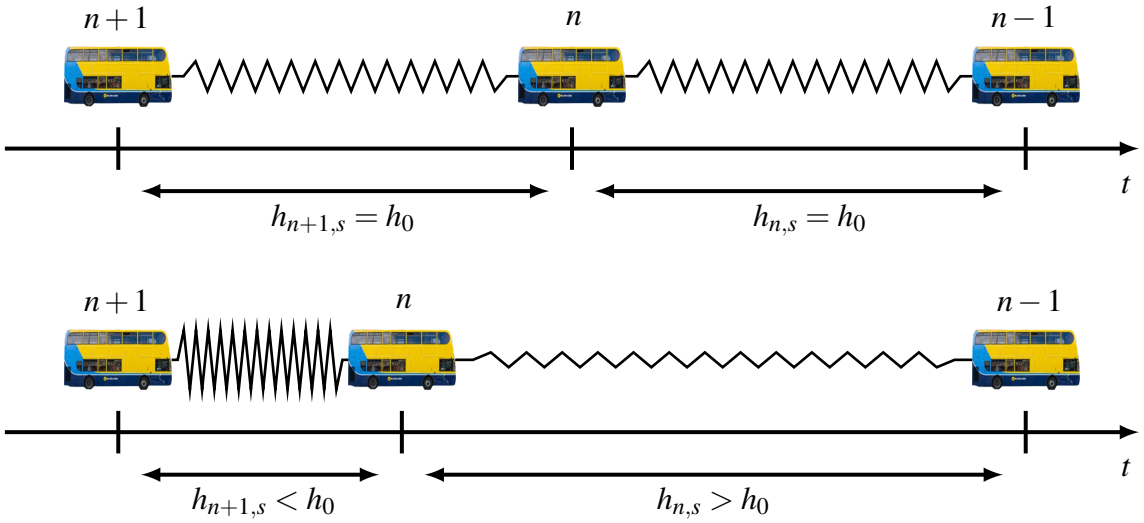


Figure 4.5: Idea of the forward-backward-headway control as connecting springs [1]

forward-headway control we control both buses that are related by their headway instead of just controlling the succeeding bus.

This approach has another major difference compared to the forward-headway approach: if bus n arrives at stop s, so at a time we need to compute the control input $u_{n,s}$, the forward headway $h_{n,s}$ is available but the backward headway $h_{n+1,s} = a_{n+1,s} - a_{n,s}$ is not available, since, as far as the buses n and $n+1$ did not bunch, bus $n+1$ did not arrive yet at stop s. This means we do not only need directly available data but also predicted headway data. In [XAD11] the authors assumed that also the backward-headway data would be available and presented a stability analysis for the resulting control strategy. This is one possible situation where we can apply the concept of the prediction of headway trajectories like described in the previous chapter.

Since we did not consider the prediction of the stop-based headways $h_{n,s}$ but of the continuous-time headways $h_n(t)$ previously, we need to use an estimation for the arrival time of bus $n+1$ at stop s to determine the temporal prediction horizon. In the literature review in chapter 1 we listed several works which address the problem of bus arrival time prediction and could be used for this task.

The forward-backward-headway-control approach is summarized in the following

[1] photo of the Dublin Bus vehicle taken from: `http://www.thejournal.ie/dubiin-bus-strike-2919690-Aug2016`

equation:

$$
\begin{aligned}
d_{n,s}^{\mathrm{fb}} &= \hat{d}_{n,s} + \underbrace{d_s}_{\substack{\text{scheduled slack}\\ \text{based dwell time}}} + \underbrace{\gamma_s^{\mathrm{f}}\left(h_0 - h_{n,s}\right)}_{\substack{\text{forward-headway}\\ \text{based dwell time}}} + \underbrace{\gamma_s^{\mathrm{b}}\left(h_{n+1,s} - h_0\right)}_{\substack{\text{backward-headway}\\ \text{based dwell time}}} \\
&= \beta_s h_0 + \beta_s\left(h_{n,s} - h_0\right) + d_s + \gamma_s^{\mathrm{f}}\left(h_0 - h_{n,s}\right) + \gamma_s^{\mathrm{b}}\left(h_{n+1,s} - h_0\right).
\end{aligned}
$$

For the same reasons like in the previous subsection 4.2.1 we choose $\gamma_s^{\mathrm{f}} = \alpha + \beta_s$ and $\gamma_s^{\mathrm{b}} = \alpha$ with $\alpha \in \left(0, \frac{1}{2}\right)$ like the authors in [XAD11]. With this choice we obtain the law of motion of the forward-backward-headway-controlled model as

$$
\begin{aligned}
a_{n,s+1} &= a_{n,s} + \Delta t_{s+1} + d_{n,s}^{\mathrm{fb}} + \nu_{n,s+1} \\
&= a_{n,s} + \Delta t_{s+1} + \beta_s h_0 + d_s + \alpha\left(h_0 - h_{n,s}\right) + \alpha\left(h_{n+1,s} - h_0\right) + \nu_{n,s+1}
\end{aligned}
$$

and the following equation for the headways for $1 < n \leq N$

$$
\begin{aligned}
h_{n,s+1} &= a_{n,s+1} - a_{n-1,s+1} \\
&= \alpha h_{n+1,s} + (1-2\alpha)\, h_{n,s} + \alpha h_{n-1,s} + \hat{\nu}_{n,s+1} \\
&= \alpha\Big(\underbrace{h'^{n}_{n+1,s}}_{\substack{\text{predicted}\\ \text{headway}}} - \underbrace{\varepsilon^{n}_{n+1,s}}_{\substack{\text{prediction}\\ \text{error}}}\Big) + (1-2\alpha)\, h_{n,s} + \alpha h_{n-1,s} + \hat{\nu}_{n,s+1},
\end{aligned}
$$

with $\hat{\nu}_{n,s+1} = \nu_{n,s+1} - \nu_{n-1,s+1}$ and $\varepsilon^n_{i,s} = h'^n_{i,s} - h_{i,s}$ as the prediction error for the headway $h_{i,s}$ if it is predicted at a time bus n arrived at stop s. We denote the approximation of the headway $h_{n,s+1}$ using predicted values like $h'^n_{n+1,s}$ by $h''_{n,s+1}$, i.e., in this case we get

$$
\begin{aligned}
h''_{n,s+1} &= \alpha(h'^n_{n+1,s}) + (1-2\alpha)\, h_{n,s} + \alpha h_{n-1,s} + \hat{\nu}_{n,s+1} \\
&= h_{n,s+1} + \alpha\varepsilon^n_{n+1,s}.
\end{aligned}
$$

Again we can reformulate this using the operators B and F from the previous subsection as convolution with a vector $\vec{f}$ as in equation (4.15) with $f_0 = 1-2\alpha, f_1 = \alpha, f_{-1} = \alpha$. This time we may allow the more general case of $\vec{f}$ being some probability mass function of a random variable, that is not necessarily non-negative. We set $h'^m_{i,s} = h_0$ for all m, s and i with $i \leq 1, i > N$. This implies $\varepsilon^m_{i,s} = 0$ for all m, s and i with $i \leq 1, i > N$ and we get for

all n with $1 < n \leq N$

$$\begin{aligned} h_{n,s+1} &= \sum_{i=-\infty}^{\infty} f_{n-i} h_{i,s} + \hat{v}_{n,s+1} \\ &= \sum_{i=-\infty}^{n} f_{n-i} h_{i,s} + \sum_{i=n+1}^{\infty} f_{n-i} \left(h'^{n}_{i,s} - \varepsilon^{n}_{i,s} \right) + \hat{v}_{n,s+1} \\ &= \underbrace{\sum_{i=-\infty}^{n} f_{n-i} h_{i,s} + \sum_{i=n+1}^{\infty} f_{n-i} \left(h'^{n}_{i,s} \right) + \hat{v}_{n,s+1}}_{=h''_{n,s+1}} - \sum_{i=n+1}^{\infty} f_{n-i} \left(\varepsilon^{n}_{i,s} \right). \end{aligned} \tag{4.19}$$

We define

$${}^{f}\varepsilon^{n}_{s} = \begin{cases} \sum_{i=n+1}^{\infty} f_{n-i} \varepsilon^{n}_{i,s} & , \ 1 < n \leq N \\ 0 & , \text{ else} \end{cases}$$

and the corresponding vector ${}^{f}\vec{\varepsilon}_{s} = \begin{pmatrix} \cdots & 0 & {}^{f}\varepsilon^{2}_{s} & \cdots & {}^{f}\varepsilon^{N}_{s} & 0 & \cdots \end{pmatrix}^{T}$ and get for n with $1 < n \leq N$ by subtracting h_0 from both sides of equation (4.19) the general relation between the headway deviations using true values and predicted values as

$$\Delta h_{n,s+1} = \Delta h''_{n,s+1} - {}^{f}\varepsilon^{n}_{s}. \tag{4.20}$$

We get the following formula for the headways of our controlled model using predicted headway values $h'^{n}_{i,s}$ for $1 < n \leq N$:

$$\begin{aligned} \Delta h''_{n,s+1} &= \Delta h_{n,s+1} + {}^{f}\varepsilon^{n}_{s} \\ &= \sum_{i=-\infty}^{\infty} f_{n-i} \Delta h_{i,s} + \hat{v}_{n,s+1} + {}^{f}\varepsilon^{n}_{s}. \end{aligned} \tag{4.21}$$

We reformulate equations (4.20) and (4.21) for all n with the vectorized forms

$$\begin{aligned} \vec{\Delta h}_{s+1} &= \begin{pmatrix} \cdots & 0 & \Delta h_{2,s+1} & \cdots & \Delta h_{N,s+1} & 0 & \cdots \end{pmatrix}^{T}, \\ \vec{\Delta h''}_{s+1} &= \begin{pmatrix} \cdots & 0 & \Delta h''_{2,s+1} & \cdots & \Delta h''_{N,s+1} & 0 & \cdots \end{pmatrix}^{T} \end{aligned}$$

and by use of the operators B and F from the previous subsection as

$$\vec{\Delta h}_{s+1} = \vec{\Delta h''}_{s+1} - {}^{f}\vec{\varepsilon}_{s} \tag{4.22}$$

and

$$\begin{aligned}
\Delta\vec{h}''_{s+1} &= B\left(F\left(\Delta\vec{h}_s\right)+\vec{\hat{v}}_{s+1}+{}^f\vec{\varepsilon}_s\right)\\
&= BF\left(\Delta\vec{h}_s\right)+\vec{\hat{v}}_{s+1}+{}^f\vec{\varepsilon}_s\\
&\overset{(4.22)}{=} BF\left(\Delta\vec{h}''_s-{}^f\vec{\varepsilon}_{s-1}\right)+\vec{\hat{v}}_{s+1}+{}^f\vec{\varepsilon}_s\\
&= BF\left(\Delta\vec{h}''_s\right)-BF\left({}^f\vec{\varepsilon}_{s-1}\right)+\vec{\hat{v}}_{s+1}+{}^f\vec{\varepsilon}_s.
\end{aligned}$$

We define

$${}^f\vec{v}_{s+1}=-BF\left({}^f\vec{\varepsilon}_{s-1}\right)+\vec{\hat{v}}_{s+1}+{}^f\vec{\varepsilon}_s$$

and get inductively the similar equation to (4.16)

$$\Delta\vec{h}''_{s+1}=\sum_{j=0}^{s}BF^j\left({}^f\vec{v}_{s+1-j}\right). \tag{4.23}$$

We find the following stability result for the general *headway-based* control in proposition 4.4.

Proposition 4.4 If for some bounds $M, M_p \in \mathbb{R}_{>0}$ it holds $|v_{n,s}| \leq M$ for all n and s, and if for the prediction error $\left|\varepsilon_{i,s}^n\right| = \left|h_{i,s}'^n - h_{i,s}\right| \leq M_p$ holds for all n, s and i with $i > n$ which are needed for the control, then for all n and s the headway deviations $\Delta h''_{n,s}$ are bounded by $2(M+M_p)s$ under the assumptions of this section.

Proof. The proof is analog to the proof of proposition 4.3.

By the assumption of bounded prediction errors $\varepsilon_{i,s}^n$ we get for all n and s

$$\left|{}^f\varepsilon_s^n\right| = \left\{\begin{array}{ll}\left|\sum_{i=n+1}^{\infty} f_{n-i}\varepsilon_{i,s}^n\right| & ,\ 1<n\leq N\\ 0 & ,\ \text{else}\end{array}\right\} \leq \underbrace{\sum_{i=-\infty}^{\infty} f_{n-i}}_{=1}\left|\varepsilon_{i,s}^n\right| \leq M_p. \tag{4.24}$$

By use of the triangle inequality we get for all n and s

$$\begin{aligned}
\left|\left({}^{f}\vec{v}_{s+1}\right)_n\right| &= \left\{ \begin{array}{cc} \left|\hat{v}_{n,s+1} - \sum_{i=-\infty}^{\infty} f_{n-i}\cdot\left({}^{f}\varepsilon_{s-1}^{i}\right) + {}^{f}\varepsilon_{s}^{n}\right| & ,\ 1<n\leq N \\ 0 & ,\ \text{else} \end{array} \right\} \\
&\leq \left|\hat{v}_{n,s+1}\right| + \left|\sum_{i=-\infty}^{\infty} f_{n-i}\cdot\left({}^{f}\varepsilon_{s-1}^{i}\right)\right| + \underbrace{\left|{}^{f}\varepsilon_{s}^{n}\right|}_{\leq M_p} \\
&\overset{(4.24)}{\leq} \underbrace{\left|v_{n,s+1}\right|}_{\leq M} + \underbrace{\left|v_{n-1,s+1}\right|}_{\leq M} + \underbrace{\sum_{i=-\infty}^{\infty} f_{n-i}}_{=1}\cdot\underbrace{\left|{}^{f}\varepsilon_{s-1}^{i}\right|}_{\leq M_p} + M_p \\
&\overset{(4.24)}{=} 2(M+M_p).
\end{aligned} \tag{4.25}$$

The statement follows by viewing equation (4.23) componentwise and taking absolute values, equations (4.12) and (4.25) and the triangle inequality:

$$\begin{aligned}
\left|\Delta h''_{n,s+1}\right| &\overset{(4.23)}{=} \left|\sum_{j=0}^{s}\left(BF^{j}\left({}^{f}\vec{v}_{s+1-j}\right)\right)_n\right| \\
&\leq \sum_{j=0}^{s}\left|\left(BF^{j}\left({}^{f}\vec{v}_{s+1-j}\right)\right)_n\right| \\
&\underset{(4.25)}{\overset{(4.12)}{\leq}} \sum_{j=0}^{s} 2\left(M+M_p\right) \\
&\leq 2\left(M+M_p\right)(s+1).
\end{aligned}$$

■

4.2.3 Numerical Experiment

In this subsection we present the results of our numerical simulation. We implemented the deterministic bus-route model described in the previous part using MATLAB, and compare the trajectories for the uncontrolled case and the controlled cases using the forward-headway-based and forward-backward-headway-based approach in the figures 4.6,4.7 and 4.8. For this we choose the parameters given in table 4.1. Like stated in [Dag09] the value β_s typically ranges between 10^{-2} and 10^{0}.

In the following we use the term *schedule*. The scheduled arrival times for $s>0$ are defined as

$$t_{n,s} = t_0 + (n-1)\,h_0 + s\Delta t + (s-1)\cdot(\beta h_0 + d_0).$$

symbol	description	value
N	number of buses	3
N_S	number of stops -1	50
Δs	distance between stops	500 [m]
$\hat{\beta}$	passenger arrival rate at stop, equal for all s	1 [passenger/minute]
λ	boarding time per passenger	5 [sec]
β	$\hat{\beta} \cdot \lambda$	1/12
d_0	scheduled slack time	15 [sec]
h_0	target headway	5 [min]
v_{op}	cruising speed	50 [km/h]
Δt	$\Delta s / v_{\text{op}}$, travel time between stops, equal for all s	36 [sec]

Table 4.1: Simulation parameters

In order to implement the forward-backward-headway-based approach we also need to decide for a prediction technique. At a time $a_{n,s}$ when bus n arrives at stop s we need to predict the value of the headway $h_{n+1,s}$ to the succeeding bus at stop s. Since the arrival time of bus $n+1$ at stop s is the only unknown value we need for that and since we are not interested in a trend detection of the headway trajectory in this case we decide to employ a simple arrival time prediction. To this we measure the offset distance $d^n_{n+1,s}$ between the position of bus $n+1$ and stop s at time $a_{n,s}$. In case bus $n+1$ is at a stop $\hat{s}$ at time $a_{n,s}$ we denote its remaining dwell time at this stop by $\hat{d}_{n+1}$, otherwise we set $\hat{d}_{n+1} = 0$. Furthermore we count the number m of stops between bus $n+1$ and stop s (excluding stops s and $\hat{s}$) and get the expected arrival time as

$$a'^n_{n+1,s} = a_{n,s} + \underbrace{d^n_{n+1,s}/v_{\text{op}}}_{\substack{\text{pure travel time} \\ \text{to stop } s}} + m \cdot \underbrace{(\beta h_0 + d_0)}_{\substack{\text{scheduled dwell} \\ \text{time per stop}}} + \underbrace{\max\{0, t_0 + n \cdot h_0 - a_{n,s}\}}_{\substack{\text{remaining dwell time at the depot} \\ \text{in case bus } n+1 \text{ is still there}}} + \underbrace{\hat{d}_{n+1}}_{\substack{\text{possibly} \\ \text{remaining} \\ \text{dwell time} \\ \text{of bus } n+1 \\ \text{at current stop}}}$$

and thus the expected headway as

$$\begin{aligned} h'^n_{n+1,s} &= a'_{n+1,s} - a_{n,s} \\ &= d^n_{n+1,s}/v_{\text{op}} + m \cdot (\beta h_0 + d_0) + \max\{0, t_0 + n \cdot h_0 - a_{n,s}\} + \hat{d}_{n+1}. \end{aligned}$$

Due to the properties of our control approach, which can be seen as a P-controller, the headways are likely to oscillate around the target headway within a certain range. In this situation neither the arrival time prediction nor our regression methods would lead to reliable headway predictions. For this reason we use the forward-backward-headway-based control inputs only if for the predicted headway value $\left| h'^n_{n+1,s} - h_0 \right| > 30$ [sec] holds, otherwise we use the forward-headway-based control inputs.

The simulation is set up as follows: we set $v_{n,s} = 0$ for all n and s, i.e., we consider the deterministic bus-route model. The scheduled departure times at stop $s = 0$ would be

$$\begin{aligned} t_{1,0} &= 0, \\ t_{2,0} &= 300 \text{ [sec]}, \\ t_{3,0} &= 600 \text{ [sec]}. \end{aligned}$$

We only disturb the departure times from the depot

$$\begin{aligned} \hat{t}_{1,0} &= 0, \\ \hat{t}_{2,0} &= 300 + 30 \text{ [sec]}, \\ \hat{t}_{3,0} &= \hat{t}_{2,0} + 300 + 30 \text{ [sec]}, \end{aligned}$$

i.e., both buses $n = 2$ and $n = 3$ leave the depot 30 seconds delayed, but we do not disturb the dwell times beside by using our control inputs.

Figure 4.6 shows the space-time diagram of our bus route for the uncontrolled case. Furthermore we visualized the headway trajectories and the deviations from the scheduled arrival times at the stops $a_{n,s} - t_{n,s}$ in corresponding colors. We see that the buses $n = 2$ and $n = 3$ bunch after around 45 minutes and that the headway $h_{2,s}\,(\sim h_2\,(t))$ and thus the deviation from the scheduled arrival times of the second bus are strictly monotonically increasing.

Figure 4.7 shows the results if we apply the forward-headway-based control to the same initial situation. We performed a parameter study by trying several values for the control parameter α^{forw} and found that for $\alpha^{\text{forw}} \to 1$ we obtain the best results in our specific case, so the results presented here are created with $\alpha^{\text{forw}} = 1$. Even though we can avoid the buses from bunching we see that the headways $h_{2,s}$ are increasing and $h_{3,s}$ are decreasing at the end of the simulation. This is due to the fact that the chosen scheduled slack time d_0 is not sufficient to compensate the initial headway deviation if we only accelerate bus $n = 2$, such that we can not reach the target headway h_0. This also results in an increasing deviation of the scheduled arrival times of the buses $n = 2$ and $n = 3$. We could create more flexibility by increasing the scheduled slack times, but this leads to longer scheduled travel times for the customers and thus to a lower default service quality and possibly higher costs for the service provider. Nevertheless, we reached our goal to prevent the buses from bunching.

Figure 4.8 visualizes our bus-route model when applying the forward-backward-headway-based approach. Again our numerical test showed that we obtain the best performance for $\alpha^{\text{forw-back}} \to \frac{1}{2}$, so we chose $\alpha^{\text{forw-back}} = \frac{1}{2}$ for our simulation. The dashed lines in the second plot indicate the predicted headway trajectories, where we set those

predicted values $h'^{n}_{n+1,s}$ to h_0 which fulfill $\left|h'^{n}_{n+1,s} - h_0\right| < 30$ [sec]. This means that for these cases our method gives the same control inputs like the forward-headway-based method. We observe that in this setting we do not only eliminate bus bunching, but also get much smaller headway deviations and deviations from the scheduled arrival times. This is due to the fact that we compensate the initial headway deviation not only by accelerating bus $n = 2$, but also by slowing down bus $n = 1$. We thus obtain in our specific case a much better performance than using the forward-headway-based approach with the same scheduled slack time.

minimum headway	$\min\limits_{n>1;s} \; h_{n,s}$
mean absolute headway deviation	$\frac{1}{(N-1)\cdot(N_S+1)} \sum\limits_{n>1;s} \lvert h_{n,s} - h_0 \rvert$
maximum positive deviation from schedule	$\max\limits_{n;s>0} \; \max\left(a_{n,s} - t_{n,s}, 0\right)$
mean positive deviation from schedule	$\frac{1}{N \cdot N_S} \sum\limits_{n;s>0} \max\left(a_{n,s} - t_{n,s}, 0\right)$

Table 4.2: Performance criteria

We compare the three cases of the uncontrolled, the forward-headway-based-controlled and the forward-backward-headway-based-controlled model based on the criteria listed in table 4.2. Even though our goal is to prevent buses from bunching and thus we mainly focus on the headways becoming close to zero or not, we also include the positive deviations from scheduled arrival times. Although we do not operate the buses using a fixed schedule since passengers do not rely on scheduled arrival times as mentioned before, we observe from the figures 4.6,4.7 and 4.8 that later arrival times at stops might go along with longer travel times like in our case which directly affect the passengers inside the bus. In table 4.3 we present the results of our simulation.

	minimum headway [min]	**mean headway deviation** [min]	**max pos. schedule deviation** [min]	**mean pos. schedule deviation** [min]
uncontrolled	0.00	4.54	24.75	3.48
forward	4.03	1.88	12.38	1.93
forward-backward	4.97	0.09	1.01	0.47

Table 4.3: Comparison: uncontrolled; forward-headway-based control; forward-backward-headway-based control

We observe that even though we used a very simple method for the headway prediction the results of the forward-backward-headway-based control are much better than the simple forward-headway-based control which indicates the potential of our predictive-control approach.

4.2.4 Predictive Control as Anticipative Correction

As mentioned before the real stop arrival times of a bus may differ from the scheduled arrival times due to varying traffic conditions and unexpected passenger demands. But these unforeseen events might also have the opposite effect, namely that in some cases they might lead to a reduction of the headway deviations. This is the reason why in a practical application we should modify for example the forward-headway-based control in equation (4.6) in such a way, that we just apply our holding strategy if the headway deviation passed a certain threshold or if it is going to pass this one in future. A control strategy based on forward headways could look like

$$\begin{aligned} d_{n,s}^{\text{th}} = \hat{d}_{n,s} + d_s - \left(\alpha^0 + \beta_s\right) \operatorname{sign} \underbrace{\left(h_{n,s} - h_0\right)}_{\substack{\text{current headway} \\ \text{deviation}}} \cdot \max\left(\left(\left|h_{n,s} - h_0\right| - \theta\right), 0\right) \\ - \alpha^p \operatorname{sign} \underbrace{\left(h'_{n,s+p} - h_0\right)}_{\substack{\text{future headway} \\ \text{deviation}}} \cdot \max\left(\left(\left|h'_{n,s+p} - h_0\right| - \theta\right), 0\right), \end{aligned}$$

for some stop prediction horizon $p \in \mathbb{N}_{>0}$ and some constants $\alpha^0, \alpha^p \in \mathbb{R}_{\geq 0}$. The idea of including future forward headways here is an *anticipatively-corrective control* strategy, i.e., the goal is "to control the system before it becomes too disrupted to be restored to a stable condition" ([Han+15]).

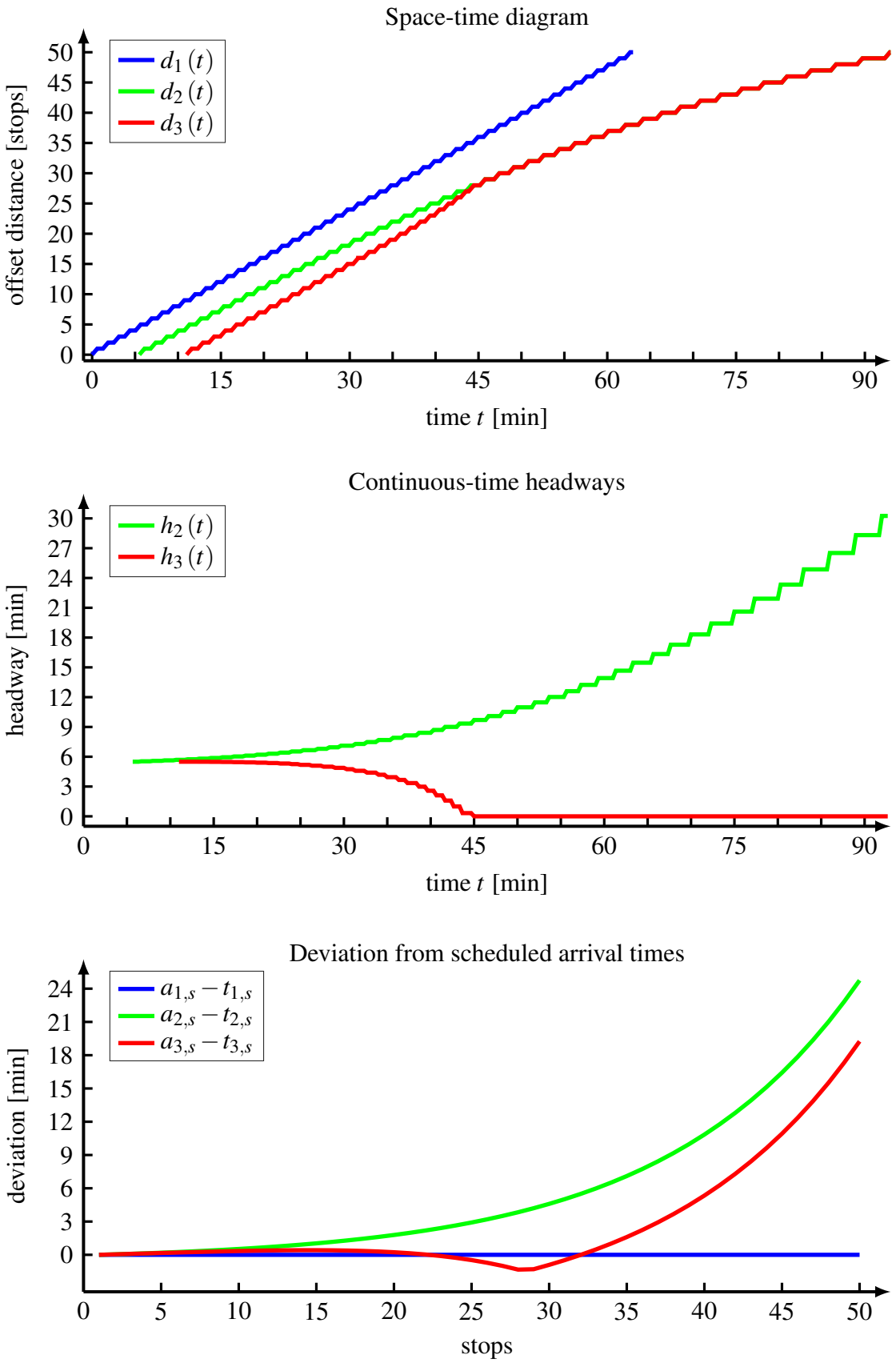


Figure 4.6: Trajectories of the uncontrolled bus-route model

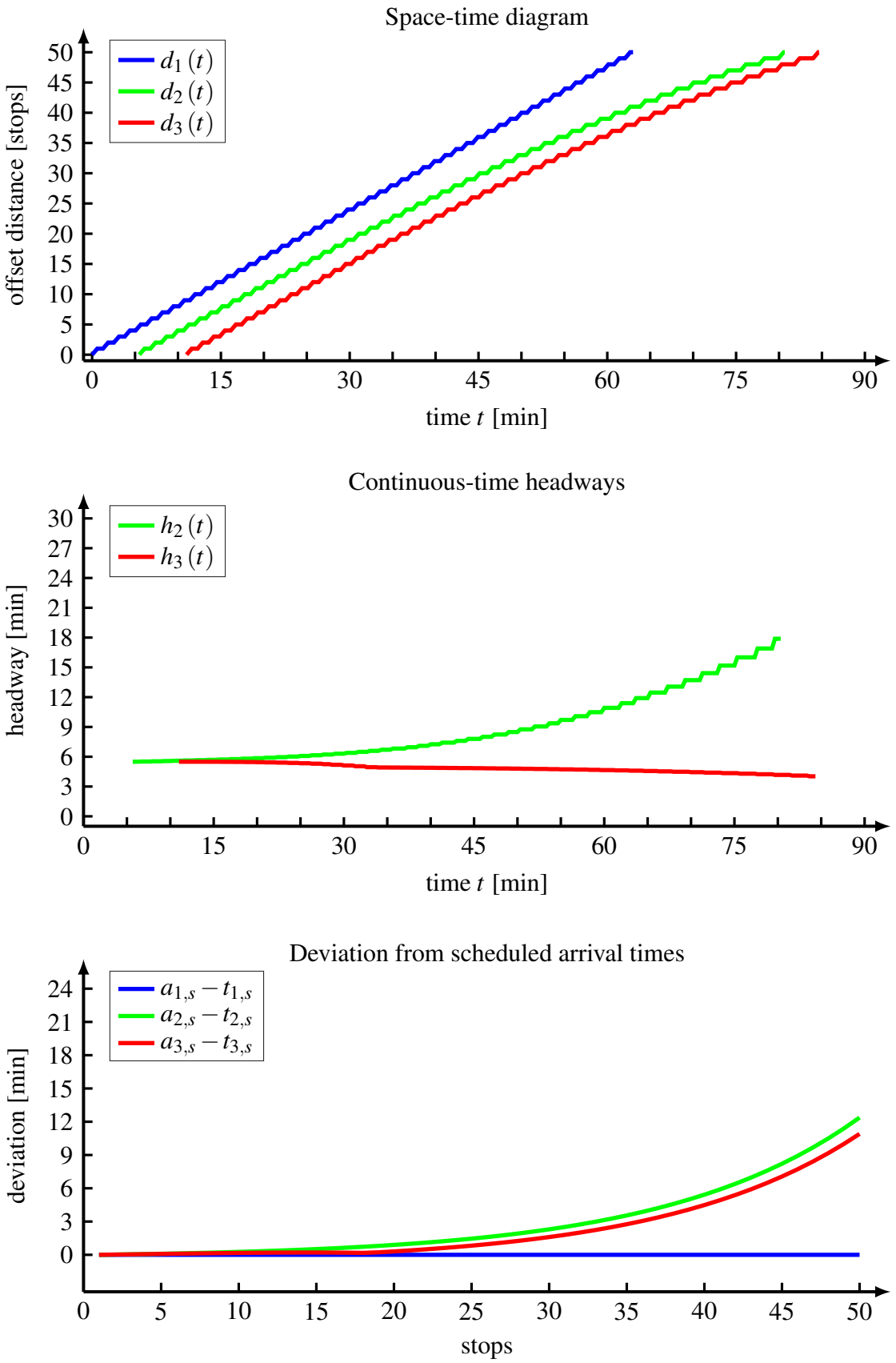


Figure 4.7: Trajectories of the forward-headway-based-controlled bus-route model

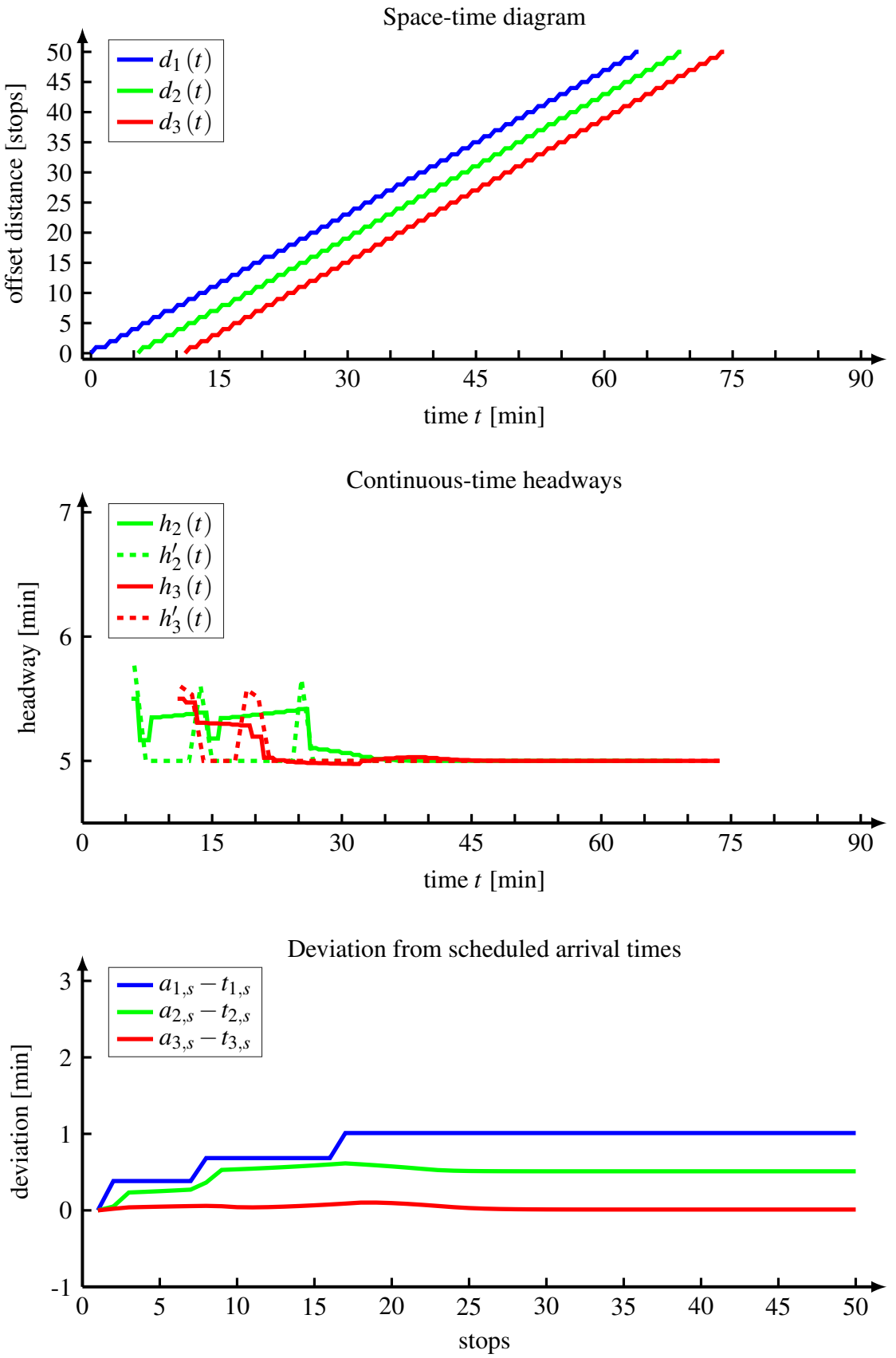


Figure 4.8: Trajectories of the forward-backward-headway-based-controlled bus-route model

5. Future Work and Conclusion

In this final chapter we review the findings of this thesis and propose potential starting points for future work based on the presented predictive-control framework.

5.1 Conclusion

Bus bunching has negative effects as well for the passengers, who have to accept longer travel and waiting times, as for the service providers, who have to face a lower cost efficiency.

In this thesis we presented a predictive-control framework based on real data to reduce bus bunching. By this we combined the two well known fields of *bunching prediction* and *corrective control to reduce bunching*. To our best knowledge none of the existing work in this area addressed all parts of the predictive-control chain like done in this thesis, including acquisition of real data, prediction of headway trajectories and corrective control strategies.

We discussed the acquisition of the real data provided by *Dublin Bus* and the transformation to space-time trajectories, including necessary preprocessing steps like the segment mapping.

Following this we presented the continuous-time headway and discussed the preprocessing of the headway trajectories. We formulated the prediction task in the context of time-series forecasting and compared the performances of *linear-regression-and-extrapolation-, kernel-regression-and-extrapolation-, artificial-neural-network-* and

autoregressive-models. In our numerical experiment we found that the methods show similar performances, even though the artificial-neural-network approach appeared to give the best results for most of the performance measures. To our best knowledge most of the existing works in the field of bunching prediction are based on bus arrival time prediction and do not explicitly consider the headway trajectories as basis for the prediction.

In the last chapter we presented an analytical model of an artificial bus route. We presented and discussed stability properties. Secondly we presented a classical control approach using directly available data as well as our predictive-control approach including predicted headway data and showed the stability of the controlled bus-route model under certain assumptions. We created a MATLAB simulation and observed based on an illustrative example the advantage of the predictive-control approach as well in terms of bunching as in terms of schedule adherence and thus impact on the passengers.

5.2 Future Work

- The modularized structure of our predictive-control framework illustrated in figure 4.1 enables us to improve the entire approach by improving its single parts.
  - The *data acquisition* part could be studied further to improve the quality of the space-time trajectories. For example, we could further tune and elaborate the HMM approach for the segment mapping to ensure linear increasing space-time trajectories or, in case this should not be possible, to return a suitable labeling for outliers.
  - As already mentioned in chapter 3 we might be able to improve the *prediction* by including exogenous inputs like the information on *neighboring buses*, *day of the week*, *time of the day*, *weather data*, *traffic information* like traffic light control, and later in the operational case the *control inputs* of our holding strategy. Furthermore it will be necessary to investigate the *feedback effect of a data-driven predictive-control framework*: the headway prediction will influence the control actions and thus the dynamics of the bus route and its future headway trajectories, which will affect the headway prediction in the next step, etc.. Thus data-driven prediction used for control will bias the prediction.
  - For the *control* it would be worth investigating different holding policies, for instance, those which include exogenous inputs. Furthermore we mentioned the alternative application of predictive-control in the context of anticipatively-corrective control strategies.
- In this thesis we concentrated on headways to analyze and reduce bus bunching. As we have seen while discussing the real data it might be difficult to assign to a bus its

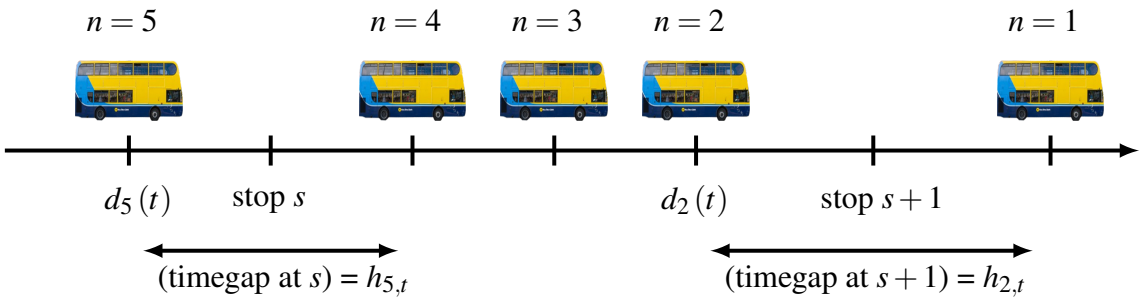


Figure 5.1: Illustration of stop-based timegaps at time t [1]

preceding and succeeding buses only using the GPS data due to data noise and the fact that buses of the same route might be operated on different shapes. Furthermore it might be interesting to investigate bunching not only within one route but among different routes serving common stops.

To this we would like suggest another way to formulate the bunching problem, which we call *stop-based timegap approach*. An illustration of this is given in figure 5.1. This approach does not focus on the headway between two specific vehicles but assigns the continuous-time headway between the two neighboring buses of a stop to this stop at a specific time. Thus independent on the number of vehicles or routes considered serving the stops we can always compute for each stop and each time step the corresponding timegap by determining the two adjacent buses and computing the corresponding headway. Figure 5.2 shows the timegap values and the corresponding space-time diagram of route *46A*, *inbound*, of the 28th of November 2012, where the X-axis indicates the time, the Y-axis indicates the stop, and the color indicates the value of the corresponding timegap in minutes. A bunching event at a stop corresponds to a small (zero) timegap.

- The next step would be the implementation of the presented predictive-control framework in a real system and to explore its performance in the field. For this we would need to consider the following issues:
  - We need to develop heuristics like mentioned in chapter 4 to incorporate predicted headway values including unknown prediction errors in a robust way in a control policy.
  - In a real system we need to find a suitable way to present the online information on headway distributions and control inputs to the drivers. This includes the implementation of suitable devices in the buses and the training and motivation of the drivers.

[1] photo of the Dublin Bus vehicle taken from: `http://www.thejournal.ie/dubiin-bus-strike-2919690-Aug2016`

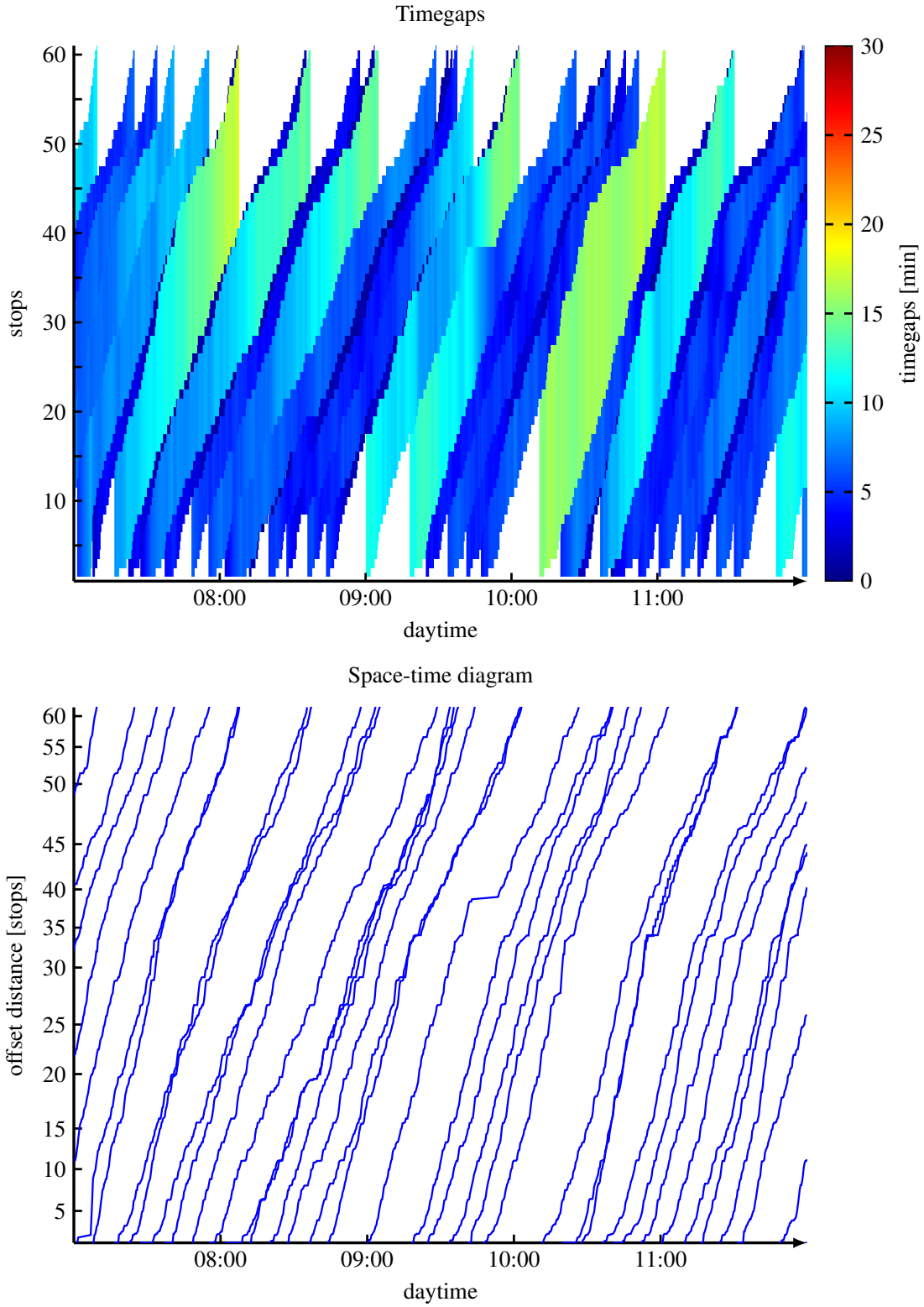


Figure 5.2: Stop-based timegaps of route *46A*, *inbound*, of 28th Nov 2012

– Instead of providing the drivers with strict control inputs of a stop-based holding rule one could think about including the driver's information on the current traffic situation into the control strategy. This could be done as follows: instead of regulating the headways by *controlled dwell times at stops* one could consider *controlled segment travel times between stops*. In this case the driver could decide based on the current traffic situation if he can reach the suggested segment travel time by waiting at a stop or if he should flow with the traffic, what might also increase the segment travel time in case of congestion.

A. Appendix to chapter 2

A.1 Hidden Markov Model Approach

In [NK09] the authors presented a map matching algorithm that includes contextual information by the use of *Hidden Markov Models (HMM)* (see, e.g., [Rab89]).

Definition A.1.1 — Hidden Markov Model. A Hidden Markov Model is a five tuple $\left(X, Z, \left\{A_k \;\middle|\; k = 2,\ldots,N_Z - 1\right\}, \left\{B_k \;\middle|\; k = 1,\ldots,N_Z\right\}, \mathbb{P}_I\right)$ corresponding to the sequence $(t_k)_{k=1}^{N_Z} \subset \mathbb{R}_{\geq 0}$ of equidistant points in time and consists of:

- $X = \{x_1,\ldots,x_{N_X}\}$ is the set of *(hidden) states*, which are the possible values for a random variable X_t.
- $Z = \{z_1,\ldots,z_{N_Z}\}$ is a tuple of observations, which are also called *emissions*.
- $A_k = A(z_{k-1}, z_k) \in \mathbb{R}^{N_X \times N_X}$ is the *transition matrix* of transitions from time t_{k-1} to t_k . The entry $(A_k)_{ij}$ gives the probability to transition from state x_i to state x_j at the transition from time t_{k-1} to t_k, i.e.,

$$(A_k)_{ij} = \mathbb{P}\left(X_k = x_j \;\middle|\; X_{k-1} = x_i\right), \quad i, j = 1,\ldots,N_X.$$

  The rows of the transition matrix sum up to one.
- $B_k = B(z_k) \in \mathbb{R}^{N_X}$ is the *emission matrix* of time t_k and its ith entry describes the probability to observe the particular measurement $z_k \in Z$ under the assumption

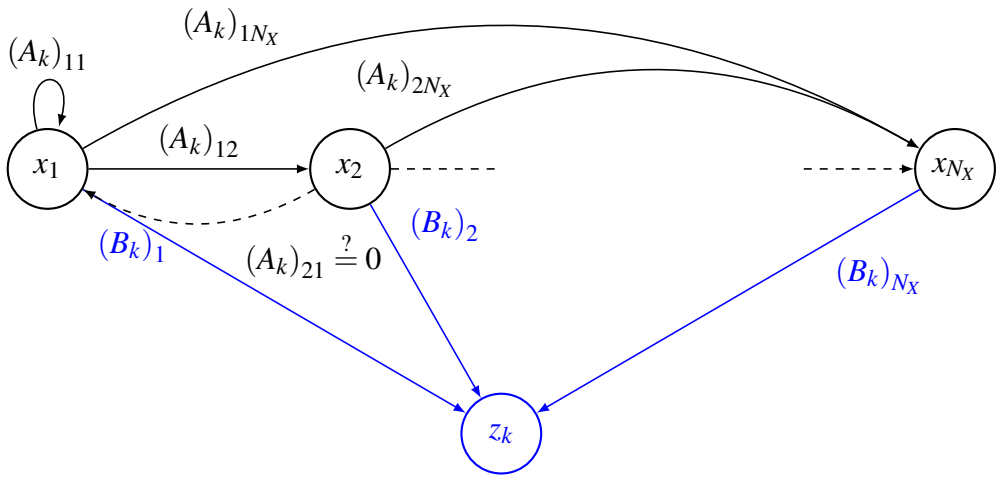


Figure A.1: Hidden Markov Model at time t_k

$X_k = x_i$, i.e.,

$$(B_k)_i = \mathbb{P}\left(z_k \mid X_k = x_i\right), \quad i = 1, \ldots, N_X.$$

- $\mathbb{P}_I \in \mathbb{R}^{N_X}$ gives the initial probabilities for the system being in one of the states in X at the beginning $k = 1$, i.e.,

$$(\mathbb{P}_I)_i = \mathbb{P}\left(X_1 = x_i\right), \quad i = 1, \ldots, N_X.$$

In our application we define the states as the shape segments S_i, i.e., $X = \{S_i \mid i = 1, \ldots, N_S\}$, and the observations are the GPS samples indicating the bus positions on S_{earth}, i.e., $Z = (z_1, \ldots, z_{N_Z}) \subset (S_{\text{earth}})^{N_Z}$. Thus the random variable X_k describes the true segment of an observation z_k at time t_k. Our goal is to find the most likely sequence of hidden states $q^* = (q_1^*, \ldots, q_{N_Z}^*) \subset X^{N_Z}$ for a given sequence Z of observations and given model parameters $\lambda_Z = (\{A_k \mid k = 2, \ldots, N_Z\}, \{B_k \mid k = 1, \ldots, N_Z\})$, i.e., using *Bayes theorem* (see, e.g., [Jam+13]) we need to consider[1]

$$\begin{aligned} q^* &= \underset{q \in X^{N_Z}}{\operatorname{argmax}} \left(\mathbb{P}\left(q \mid Z; \lambda_Z\right)\right), \\ \mathbb{P}\left(q \mid Z; \lambda_Z\right) &\propto \mathbb{P}\left(q; Z \mid \lambda_Z\right) = \left((\mathbb{P}_I)_{q_1} \cdot (B_1)_{q_1}\right) \cdot \prod_{k=2}^{N_Z} \left((A_k)_{q_{k-1} q_k} \cdot (B_k)_{q_k}\right), \end{aligned} \tag{A.1}$$

where $(B_k)_{q_k}$ denotes the entry of B_k at the index corresponding to state q_k.

This method is successfully used for example in the field of *speech recognition* (see [Rab89] for further details). Here the phonemes, which are elementary speech units, are modeled as states, which suits very well to our map matching problem since on the one

[1] based on `https://de.wikipedia.org/wiki/Viterbi-Algorithmus`

hand some phoneme-to-phoneme transitions are more likely than others, and on the other hand the speaking rate depends on the speaker and changes the physical input signal but not the content. The HMM compensates these individual variations in speaking rate, since there is the possibility to stay within a state or even to skip some states by choosing an according sequence of states. This problem is similar to the map matching problem, since certain segment-to-segment transitions are more likely than others w.r.t. geographical relations between the shape segments and even though buses might drive with individual commercial speeds the fact remains unchanged that ideally the buses should drive in one direction along the shape and thus the sequence of segment indices should be monotonically increasing.

There are two ways to determine the transition and emission probabilities for our problem.

- The first one is using a technique out of the field of *machine learning*. Based on historical data the *Baum-Welch-Algorithm* (see [Bil+98] for further details) computes the model parameters λ^* with a constant transition matrix A and a constant emission matrix B by using the maximum likelihood approach

  $$\lambda^* = \underset{\lambda}{\operatorname{argmax}} \left(\mathbb{P}\left(Z \mid \lambda \right) \right),$$

  for some offline available observations Z, which we denote as *training data*. This approach has two problems. The first problem is that in our case we have an infinite set of possible observations and thus non-constant emission matrices, which would be a prerequisite for applying the standard Baum-Welch-Algorithm. Another problem is that we need historical data for the training of our model. But this means that we need to compute the space-time trajectories, i.e., we need to compute offset distances. This is exactly the problem we try to solve with our HMM approach, what means we need another method like the presented predecessor-based approach as an initial step to provide us with training data.
- In order to encounter the fact that the set of possible observations is infinitely large we use time dependent emission matrices that are constructed using the current observation at the end of a transition. Furthermore instead of working with transition and emission matrices one could provide more general probability distribution functions like in [NK09].

We use the second technique from above and the suggestions for time depending transition and emission probabilities like proposed in [NK09]. We would like to point out that in the mentioned work the task was more general, since instead of finding the most

likely mapping onto a single shape, the task was to map the coordinates on a road network, including crossings and especially a much larger set of segments.

The authors modeled the GPS noise as zero-mean Gaussian, i.e.,

$$\mathbb{P}\left(z_k \mid X_k = S_i\right) \propto \exp\left(-\frac{1}{2}\left(\left\|z_k - \Pi_S^{\text{cpp}}(z_k, i)\right\|_{\text{gd}} / \sigma_z\right)^2\right).$$

The value σ_z represents the variance of the noise. We use this probability distribution to calculate the emission matrix $B_k = B(z_k)$ for the transition from time t_{k-1} to t_k as follows:

$$(B_k)_i = \mathbb{P}\left(z_k \mid X_k = S_i\right).$$

The transition probabilities for the transition from time t_{k-1} to t_k also depend on the current and last measurements and are given as follows:

$$\begin{aligned} \mathbb{P}\left(d_{k,ij}\right) &\propto \frac{1}{\beta}\exp\left(-d_{k,ij}/\beta\right), \\ d_{k,ij} &= \left|\left\|z_{k-1} - z_k\right\|_{\text{gd}} - \left\|\Pi_S^{\text{cpp}}(z_{k-1}, i) - \Pi_S^{\text{cpp}}(z_k, j)\right\|_{\mathbb{1}}\right|, \end{aligned}$$

where the value $\beta \in \mathbb{R}_{>0}$ is a design parameter. We obtain the transition matrix for the transition from time t_{k-1} to t_k again by normalizing:

$$(A_k)_{ij} = \mathbb{P}\left(d_{k,ij}\right).$$

The remaining question is how to find the maximizing sequence of states q^* in equation (A.1). To solve this problem we use the *Viterbi algorithm* (see [For73]). This *dynamic programming algorithm* uses on the one hand the variable $v_k(i)$, which gives the maximum probability to end up in state S_i after $k-1$ transitions transitioning through a sequence of length k and having the observations $z_1, \ldots, z_k$, i.e.,

$$v_k(i) = \max_{\substack{q \in X^k \\ q_k = S_i}} \left(\mathbb{P}\left(z_1, z_2, \ldots, z_k; q_1, q_2, \ldots, q_k \mid \lambda_Z\right)\right),$$

for a model with parameters λ_Z depending on the sequence of observations. On the other hand it uses the variable $\psi_k(i)$ which saves for each transition and each state the preceding state which was involved in creating the maximum value for $v_k(i)$. This idea is summarized in algorithm 1.

Algorithm 1 Viterbi algorithm [2]

1: $v_1(i) = (\mathbb{P}_I)_i \cdot (B_1)_i, \psi_1(i) = 0, \quad 1 \le i \le N_X$ ▷ Initialization
2: **for** $2 \le k \le N_Z$ **do** ▷ Recursion
3: $\quad v_k(i) = (B_k)_i \cdot \max\limits_{1 \le j \le N_X} \left((A_k)_{ji} \cdot v_{k-1}(j) \right), \quad 1 \le i \le N_X$
4: $\quad \psi_k(i) = \operatorname{argmax}\limits_{1 \le j \le N_X} \left((A_k)_{ji} \cdot v_{k-1}(j) \right), \quad 1 \le i \le N_X$
5: $\mathbb{P}\left(q^*; Z \mid \lambda\right) = \max\limits_{1 \le i \le N_X} (v_{N_Z}(i))$ ▷ Termination: probability
6: $i^*_{N_Z} = \operatorname{argmax}\limits_{1 \le i \le N_X} (v_{N_Z}(i))$ ▷ Termination: state sequence
7: $i^*_k = \psi_{k+1}\left(i^*_{k+1}\right), \quad 1 \le k < N_Z$ ▷ Determination of path
8: $q^* = \left(x_{i^*_1}, \ldots, x_{i^*_{N_Z}}\right)$ ▷ Maximizing sequence

We obtain the segment mapping by

$$\mathscr{M}_S^{\mathrm{HMM}}\colon \left\{ (p_0, p_1, \ldots, p_{N_T}) \in S_{\mathrm{earth}}^{N_T+1} \;\middle|\; N_T \in \mathbb{N} \right\} \to \bigcup_{N_T \in \mathbb{N}} \{1, \ldots, N_S\}^{N_T},$$

$$p_k \mapsto i^*_k \text{ (given by the } \textit{Viterbi} \text{ algorithm with } z_k = p_k), \quad k = 0, \ldots, N_T.$$

A nice feature of the Viterbi algorithm is the fact that it furthermore gives us the probability $P\left(q \mid Z; \lambda_Z\right)$ of the state sequence q for a given sequence of observations Z. This value could be used as one criterion to evaluate the quality of the shape projection. Furthermore we are able to give preference to monotonically increasing sequences of state indices by assigning more weights to the transitions $(A_k)_{i,j}$ with $i \le j$ than to those with $i > j$, like suggested in figure A.1. Furthermore the HMM approach is applicable to online data, i.e., we are able to use it with some modifications in an operational setting.

A.2 SIRI Data

The information given in this part about SIRI data is based on and partially taken from [Kno08] and the webpage `www.siri.org.uk`.

The abbreviation *SIRI* stands for *Service Interface for Real-Time Information*, which is an European data interface standard for exchanging information about performance of real-time public transport operations. In this thesis we are interested in its *Vehicle Monitoring service*. For this we use the *SIRI* dataset provided on `https://data.dublinked.ie/dataset/dublin-bus-gps-sample-data-from-dublin-city-council-insight-project`, which contains Dublin Bus GPS data across Dublin City for the time from the 6th of November 2012 to 30th of November 2012.

[2] based on `https://de.wikipedia.org/wiki/Viterbi-Algorithmus`

The data is given in a csv-file, and the columns of each row of this file contain the following entries:

1. **timestamp**: The timestamp is given in microseconds since the 1st of January 1970, 00:00:00 GMT. We refer to the MATLAB functions `datenum`, `datevec` and `datetime`.
2. **line ID**: The line ID is the information displayed to the passengers, like in our example *46A* . This does neither include information about the direction (inbound or outbound) nor about timing nor about the particular stop sequence, as this might vary for example at different daytimes.
3. **direction**: Each line can be operated in two directions: *inbound* (1) and *outbound* (0).
4. **journey pattern ID**: A journey pattern specifies a particular sequence of stops. This especially includes information about the direction. A journey pattern could include timings for covering the links between the stops, but does not include any other operational time information like a schedule. A line can have several patterns, as these might be changed during peak hours (e.g., short turns) or in late hours (e.g., combining of services; see figure 2.10). We would like to point to the mismatch between the *direction* field in the third column and the true direction indicated in the *journey pattern ID*. Thus we use the direction information given by the journey pattern ID.

   ▪ **Example A.1** Structure of a journey pattern ID: $\underbrace{046A}_{\text{line ID}}\ \underbrace{1}_{\text{direction}}\ \underbrace{001}_{\text{number}}$. ▪
5. **date frame**: The date frame of a vehicle journey is defined as the day of the start date of the journey.

   ▪ **Example A.2** A journey that starts on the 5th of November 2012 at 23:00 and ends on the 6th of November at 01:00 would have the date frame *2012-11-05*. ▪
6. **vehicle journey ID**: A vehicle journey describes a scheduled journey on a public transport route. A *Dated Vehicle Journey* is a planned journey run on a specific date and is a unique identifier within a journey pattern.

pattern	date frame	vehicle journey ID
046A0001	2012-11-06	7468
046A1001	2012-11-06	7468

Table A.1: Example: Unique identifier in SIRI data

7. **operator**: Bus operator, not the driver.
8. **congestion**: 0 = no, 1 = yes.

9. **longitude**: One of two GPS coordinates in degree of arc (deg) w.r.t. WGS84 (see appendix A.5).
10. **latitude**: One of two GPS coordinates in degree of arc (deg) w.r.t. WGS84 (see appendix A.5).
11. **delay**: Value that represents the delay of the bus w.r.t. the schedule, given in seconds. If the bus is ahead of schedule, this value is negative.
12. **block ID**: A section ID of the journey pattern.
13. **vehicle ID**: ID of the corresponding operated vehicle.
14. **stop ID**: See appendix A.3, used in combination with the *at stop* field in column 15 to indicate if the bus is currently at a particular stop.
15. **at stop**: 0 = no, 1 = yes, referring to stop ID.

In this thesis we only use the dated vehicle journey IDs as unique identifiers of bus runs and the timestamped GPS coordinates from the SIRI dataset, i.e., the columns 1,4,5,6,9,10 of the above mentioned csv-file.

A.3 GTFS Data

The information given in this part about GTFS data is based on and partially copied from `https://developers.google.com/transit/gtfs/`.
The abbreviation *GTFS* stands for *General Transit Feed Specification Reference*, which is a format for infrastructure data of public transportation, like schedules and associated geographic information. The GTFS data for Dublin Bus is provided on `https://data.dublinked.ie/dataset/dublin-bus-gtfs-data` and is given by the following txt.files:

- **agency.txt**: One or more transit agencies that provide the data in this feed. In our case this is *Dublin Bus*.
- **stops.txt**: Coordinates and IDs of stops, i.e., individual locations where vehicles pick up or drop passengers.
- **routes.txt**: Transit routes. A route is a group of trips that are displayed to riders as a single service. The GTFS field *route* is similar to the SIRI field *line* in the sense that it is grouping services but without containing timetable or direction information. In this thesis we consider the route *46A* . The route type determines the vehicle used

route ID	route short name	route long name	route type
0-46A-y12-1	46A	Queen's Road - Phoenix Pk Gate	3

Table A.2: Example: Structure of 'routes' file

for the transportation. A list can be found on `https://sites.google.com/site/`

shape ID	point latitude [deg]	point longitude [deg]	point number in sequence	distance traveled in shape [m]
0-46A-y12-1.322.O	53.3516...	-6.2975...	1	0
0-46A-y12-1.322.O	53.3520...	-6.2975...	2	62.43
0-46A-y12-1.322.O	53.3521...	-6.2980...	3	68.76

Table A.3: Example: Structure of the 'shapes' file

`gtfschanges/proposals/route-type`. The route type *3* for example indicates that the transportation is done by bus.

- **trips.txt**: Trips for each route. A trip is a sequence of two or more stops that occurs at specific times. The GTFS field *trip* is similar to the *dated vehicle journey* of the SIRI dataset, as it identifies a particular run of the schedule and combines location information with certain points in time. It is also the unique identifier within the GTFS dataset for a particular scheduled bus run.
- **calendar.txt**: Dates for service IDs using a weekly schedule. It is specified when a service starts and ends, as well as days of the week where a service is available.
- **calendar_dates.txt**: Exceptions for the service IDs defined in the calendar.txt file.
- **shapes.txt**: This file contains the geographic information of a route. More specific each journey pattern has its own shape. A shape is represented by a set of points like in definition 2.1.1. The points are given by longitude and latitude coordinates in degree of arc (deg) w.r.t. WGS84 (see appendix A.5). A vehicle journey of a particular pattern is approximated by connecting the points of the corresponding shape to form a *geodetic polygon*. The value *distance traveled in shape* positions a shape point as a distance traveled along a shape from the first shape point. This is equivalent to the offset distance of this point w.r.t. to the shape described in subsection 2.1.1.
- **stop_times.txt**: Times at which a vehicle arrives at and departs from individual stops for each trip.

A.4 GPS Data and Geodetic Distance

All spatial coordinates in the given datasets are represented as *latitude* and *longitude* coordinates, provided by *Global Positioning Systems (GPS)*. GPS data are based on the standard *World Geodetic System 1984 (WGS84)* ([ElR02]). This one provides a *coordinate system* whose origin is located in the center of mass of the earth, a spheroidal reference surface, which we call *reference ellipsoid*, and the *geoid*, which is a mathematical model of

the earth's sea level and describes the shape that the surface of the oceans would take only considering the influence of gravitation and rotation of the earth. In [IA00] a description of the WGS84 is given. Especially the values for the *semi-major axis a* and the *semi-minor axis b* of the reference ellipsoid are given as follows:

$$\begin{aligned} a_{\text{earth}} &= 6\,378\,137.0\ [\text{m}], \\ b_{\text{earth}} &= 6\,356\,752.3142\ [\text{m}]. \end{aligned}$$

In this thesis we do not consider the *elevation coordinate*, i.e., a point's height above or below the reference geoid, and thus approximate the surface of the earth by the WGS84 reference ellipsoid. This allows us to represent the surface of the earth S_{earth} like in the following equation:

$$\begin{aligned} S_{\text{earth}} &= \left\{ \begin{pmatrix} x & y & z \end{pmatrix}^T \in \mathbb{R}^3 \;\middle|\; \frac{x^2}{a_{\text{earth}}^2} + \frac{y^2}{a_{\text{earth}}^2} + \frac{z^2}{b_{\text{earth}}^2} = 1 \right\} \\ &= \left\{ \begin{pmatrix} a_{\text{earth}} \cos(\phi)\cos(\lambda) \\ a_{\text{earth}} \cos(\phi)\sin(\lambda) \\ b_{\text{earth}} \sin(\phi) \end{pmatrix} \;\middle|\; \phi \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right), \lambda \in (-\pi, \pi) \right\}, \end{aligned}$$

where the parameter ϕ is called *(degree of) latitude* and λ is called *(degree of) longitude*. *Parallels of latitude* (German: Breitengrad) are formed by circles surrounding the earth and in planes parallel with that of the equator. *Meridians of longitude* (German: Längengrad) are formed with a series of imaginary lines, all intersecting at both the north and south poles, and crossing each parallel of latitude at right angles. This description is illustrated in figure A.2 and can be found beside further information in [Sny87].

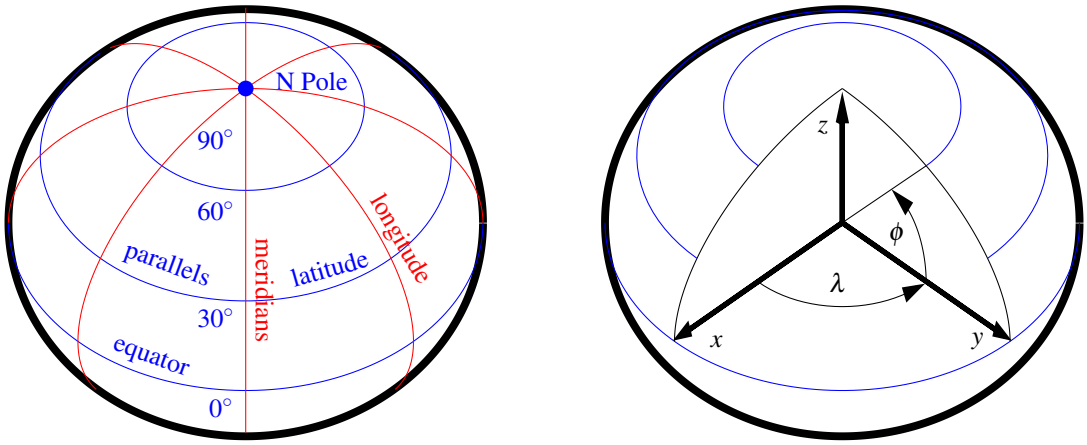


Figure A.2: Earth's coordinate system

For our application we are interested in computing the geodetic distance $\left\|P-\hat{P}\right\|_{\text{gd}}$ between two points $P,\hat{P} \in S_{\text{earth}}$ on the earth's surface, which is defined as the length of the shortest curve on the surface connecting the two points. This problem is known as *inverse geodetic problem* and includes the solution of elliptical integrals, which can be approximated by an iterative method called *Vincenty's formulae* (see [Vin75]).

A.5 Map Projection

This subsection is based and partially taken from [Sny87].

In order to visualize, for example, the locations of buses on a map we need to transform the three-dimensional coordinates describing the location as a point on the earth's surface S_{earth} into two-dimensional coordinates in a "reasonable" way. This transformation is called *map projection* and describes a systematic representation of a part of a surface of a round body, for instance, the surface of the earth, on a plane. The term "reasonable" depends on the particular scope of the transformation as there are several characteristics regarding map projections like *area, shape, scale, direction, method of construction,* etc.. In this thesis we consider the map projection of coordinates of a 30 [km] $\times$ 30 [km] square on the earth's surface around the Dublin area. In order to obtain a suitable visualization, for example, of a route's shape we are especially interested in a method that is *conformal*, i.e., the relative local angles about every point on the map are shown correctly and thus small features of shapes are mapped essentially correctly. We choose a very well known conformal method called *Mercator projection*, which is also used in a slightly modified way by services like *Google Maps*[3] and thus meets our requirements.

The Mercator projection is a cylindrical projection like shown in figure A.3[4]and uses the following formula for the projection of coordinates of the surface of the reference ellipsoid to the plane:

$$\begin{aligned} x_{\text{merc}} &= a_{\text{earth}} \cdot (\lambda - \lambda_0), \\ y_{\text{merc}} &= a_{\text{earth}} \cdot \log\left(\tan\left(\frac{\pi}{4}+\frac{\phi}{2}\right) \cdot \left(\frac{1-e_{\text{earth}}\sin(\phi)}{1+e_{\text{earth}}\sin(\phi)}\right)^{\frac{e_{\text{earth}}}{2}}\right) - y_0, \\ e_{\text{earth}} &= \sqrt{1-\frac{b_{\text{earth}}^2}{a_{\text{earth}}^2}}, \qquad \text{(eccentricity of the ellipsoid)} \end{aligned}$$

where λ_0 defines the "Zero-Meridian" and y_0 shifts the latitude of the map's origin. In this thesis we choose as origin of the map the *Millenium Spire* in Dublin. For further reading we refer to [Fen06] and [Sny87].

[3] https://en.wikipedia.org/wiki/Google_Maps

[4] source: http://geographx.co.nz/_wp/wp-content/uploads/2012/01/cylindrical.jpg

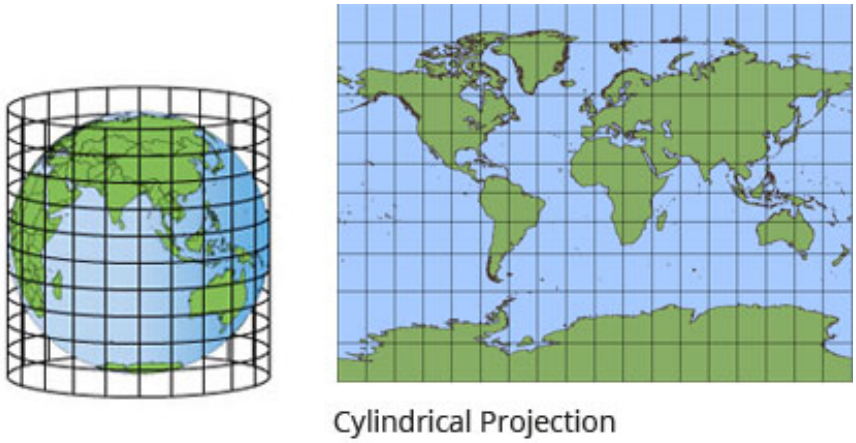


Figure A.3: Cylindrical projection [4]

B. Appendix to chapter 3

B.1 Headway Preprocessing

We briefly describe two very common preprocessing functions, called filters, to reduce the effect of noise disturbing our headway trajectories. For this let $h = (h_k)_{k=1}^{N_h}$ be a headway trajectory of a bus pair with equidistant timestamps $T = (t_k)_{k=1}^{N_h}$.

B.1.1 Moving Average

The *weighted moving average filter* replaces the value h_k at time t_k by the weighted average over all values of h with indices $\{k-w_l,\dots,k,\dots,k+w_u\}$ for the window limits $w_l, w_u \in \mathbb{N}_{\geq 0}$, i.e.,

$$\begin{aligned}
\hat{h}_k &= \sum_{i=b_l}^{b_u} \omega_{k-i}^{(k)} \cdot h_i, \\
b_l &= \max\{1, k-w_l\}, \\
b_u &= \min\{N_h, k+w_u\}, \\
\sum_{i=b_l}^{b_u} \omega_{k-i}^{(k)} &= 1, \\
\omega_{k-i}^{(k)} &\geq 0, \quad i = b_l,\dots,b_u.
\end{aligned}$$

For the *centered moving average filter* it holds $w_l = w_u$, so in our context it can only be applied in an offline case, like in the sections 3.4 and 3.5 for the smoothing of the reference and training trajectories, since at time t_k when we need to evaluate the filter the values

$\{h_{k+1},\dots,h_{k+w_u}\}$ are not accessible yet.

For this reason the *(standard) moving average filter* with $w_u = 0$ has been introduced, which can also be applied in an operational setting. In here we chose linear decreasing weights $\omega_{k-i}^{(k)} = \frac{w-i}{w(w+1)/2}$, with $w = k - b_l + 1$ and $i \in \{b_l,\dots,k\}$, and $\omega_{k-i}^{(k)} = 0$ otherwise for the moving average and equal weights $\omega_i^{(k)} = \frac{1}{w}$, with $w = b_u - b_l + 1$ and $i \in \{b_l,\dots,b_u\}$, and $\omega_i^{(k)} = 0$ otherwise for the centered moving average. The centered moving average with equal weights is implemented as the MATLAB function `smooth`.

This description and further details can be found in most text books on *statistical analysis* such as [DM57].

B.1.2 Kalman Filter

The *Kalman filter* is an application of the more general framework of *Bayesian filtering*. This description is based on [Che03] and [WB06]. It is a probabilistic approach, i.e., instead of one exact filtered value we obtain a probability distribution of values, which is called *belief*. In general we distinguish between the state $x_k \in \mathbb{R}^{N_X}$, the external input $u_k \in \mathbb{R}^{N_U}$ and the observation $z_k \in \mathbb{R}^{N_Z}$ corresponding to the equidistant times $(t_k)_{k=1}^{N_T}$. The belief is then defined as

$$\mathrm{bel}(k) = \mathbb{P}\left(x_k \;\middle|\; u_{1:k}, z_{1:k}\right).$$

It relies on the *Markov assumption*, which states that the state is a complete summary of the past. Furthermore we assume a linear state space model of the following form:

$$\begin{aligned}
x_k &= A_k x_{k-1} + B_k u_k + \varepsilon_k, && \text{(motion model)}\\
z_k &= C_k x_k + \delta_k, && \text{(measurement model)}\\
\varepsilon_k &\sim \mathcal{N}(0, R_k),\\
\delta_k &\sim \mathcal{N}(0, Q_k),
\end{aligned}$$

with $A_k \in \mathbb{R}^{N_X \times N_X}$, $B_k \in \mathbb{R}^{N_X \times N_U}$, $C_k \in \mathbb{R}^{N_Z \times N_X}$ and the positive definite matrices $R_k \in \mathbb{R}^{N_X \times N_X}$ and $Q_k \in \mathbb{R}^{N_Z \times N_Z}$, called *process noise* and *measurement noise*, and $\mathcal{N}$ denotes the normal distribution.

For a mean value $\mu_{k-1} \in \mathbb{R}^{N_X}$ and covariance matrix $\Sigma_{k-1} \in \mathbb{R}^{N_X \times N_X}$ at time index $k-1$

one iteration of the Kalman filter is given by the following equation:

$$\begin{aligned}
\hat{\mu}_k &= A_k \mu_{k-1} + B_k u_k, \\
\hat{\Sigma}_k &= A_k \Sigma_{k-1} A_k^T + R_k, \\
K_k &= \hat{\Sigma}_k C_k^T \left(C_k \hat{\Sigma}_k C_k^T + Q_k \right)^{-1}, \\
\mu_k &= \hat{\mu}_k + K_k \left(z_k - C_k \hat{\mu}_k \right), \\
\Sigma_k &= \left(I - K_k C_k \right) \hat{\Sigma}_k, \\
\text{bel}\left(x_k \right) &\sim \mathcal{N}\left(\mu_k, \Sigma_k \right),
\end{aligned}$$

where $I \in \mathbb{R}^{N_X \times N_X}$ denotes the identity matrix. In our case the states x_k as well as the observations z_k correspond to the entries in the headway trajectory h_k and we do not consider any control inputs u_k. Using a *constant motion model* we obtain the matrices $A_k = C_k = 1 \in \mathbb{R}$ for all k. Also the process noise R_k and measurement noise Q_k become scalar values.

We would like to point out the following observation regarding parameter tuning, illustrated for the one-dimensional case:

$$\begin{aligned}
&R_k = \text{const}, \quad Q_k \to 0 \quad \Rightarrow K_k \to \frac{1}{C_k} = 1 \quad \Rightarrow \mu_k \to z_k, \\
&R_k = \text{const}, \quad Q_k \to \infty \quad \Rightarrow K_k \to 0 \qquad\quad \Rightarrow \mu_k \to \hat{\mu}_k = A_k \mu_{k-1}.
\end{aligned}$$

This means by fixing the values of R_k we can control the 'trust' in the measurement by changing the value of Q_k. If we choose large values for Q_k we can consider this as trusting the motion model more than the measurement model. Reversely if we choose small values for Q_k we can consider this as trusting the measurement model more than the motion model.

The advantage of the Kalman filter is on the one hand that it is an *online* algorithm, i.e., as far as we get the value of h_k for some time t_k we are able to compute the filtered value for it and do not need to wait until succeeding points in time are available. Furthermore we do not only obtain a filtered value $\hat{h}_k = \mu_k$ online, but also with Σ_k a probabilistic confidence value, that could be used to mark outliers in the filtering process what could be applied in the prediction part.

B.1.3 Neural Network

Formally a neural network can be seen as a directed, L-partite graph (with $L \in \mathbb{N}_{\geq 2}$) with vertex set V and edge set E. Each partition L_k of V contains $m_k \in \mathbb{N}$ vertices and is called *layer* of the neural network. Each neural network has one *input layer*, one *output layer* and $L-2$ *hidden layers*. In the special case of a multilayer perceptron (MLP) the layers are ordered in such a way, that there are only edges between vertices of the $(k-1)$th layer to

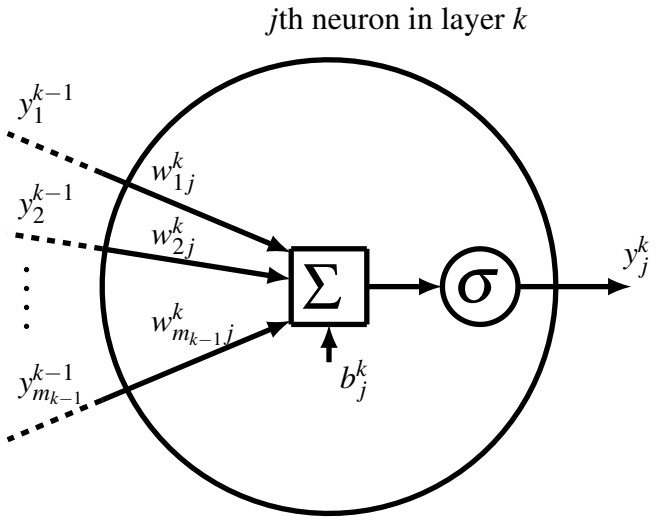


Figure B.1: Single neuron

vertices of the kth layer, where the first layer is the input layer and the last layer the output layer. Especially there are no *recurrent connections* and with that the MLP is a *feedforward* neural network. All edges are weighted, where the weight of the edge connecting the ith vertex in layer $k-1$ with the jth vertex in layer k is denoted by $w_{ij}^k \in \mathbb{R}$. Each vertex is called *neuron* (illustrated in figure B.1) and can be seen as a function that maps the outputs of the previous layer to a real number, which is used as the output of this vertex. For the jth vertex of the kth layer there is a real value $b_j^k \in \mathbb{R}$, which is called *bias*, and real valued *activation function* $\sigma_j^k \colon \mathbb{R} \to \mathbb{R}$. The neuron function f_j^k for the jth vertex of the kth layer with $k > 1$ is then defined as

$$f_j^k \colon \mathbb{R}^{m_{k-1}} \to \mathbb{R},$$
$$\begin{pmatrix} x_1 \\ \vdots \\ x_{m_{k-1}} \end{pmatrix} \mapsto \sigma_j^k \left(\sum_{i=1}^{m_{k-1}} w_{ij}^k x_i + b_j^k \right),$$

and for $k = 1$ as

$$f_j^1 : \mathbb{R}^{m_{\text{IN}}} \to \mathbb{R},$$
$$\begin{pmatrix} x_1 \\ \vdots \\ x_{m_{\text{IN}}} \end{pmatrix} \mapsto x_j,$$

where $\begin{pmatrix} x_1 \cdots x_{m_{\text{IN}}} \end{pmatrix}$ is the input vector to our network. In our case the activation functions are chosen for all vertices of the corresponding layer as follows:

1. hidden layer, $k < L$: $\sigma_j^k : \mathbb{R} \to \mathbb{R} : x \mapsto \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ (see fig B.2),
2. output layer: $\sigma_j^L = \text{id}_{\mathbb{R}}$.

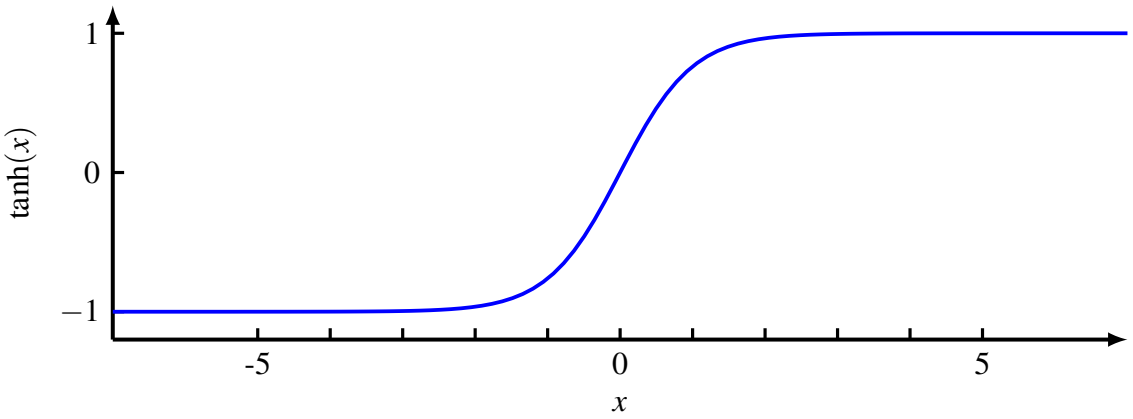


Figure B.2: Activation function of hidden layers: tanh

The input layer is used as a formal way to present the argument $(x_1 \cdots x_{m_1})^T$ to the *network function*, which is defined as

$$f_{\text{net}} : \mathbb{R}^{N_{\text{IN}}} \to \mathbb{R}^{N_{\text{OUT}}},$$
$$\begin{pmatrix} x_1 \\ \vdots \\ x_{N_{\text{IN}}} \end{pmatrix} \mapsto \begin{pmatrix} y_1^L \\ \vdots \\ y_{N_{\text{OUT}}}^L \end{pmatrix},$$

$$y_j^k = \sigma_j^k \left(\sum_{i=1}^{m_{k-1}} w_{ij}^k y_i^{k-1} + b_j^k \right), \quad j \in \{1, \ldots, m_k\}, k \in \{2, \ldots, L\},$$

where $N_{\text{IN}} = m_1$ is defined as the *number of inputs* of the neural network, $N_{\text{OUT}} = m_L$ is the *number of outputs* of the neural network and $y_j^k \in \mathbb{R}$ is the output of the jth neuron in the kth layer. An illustration of a MLP is shown in figure 3.7.

B.2 MATLAB : Neural Network Toolbox

For the numerical experiments in this thesis we make use of MATLAB's *Neural Network Toolbox*. To this we have to prepare the headway trajectories in the following way:

1. As a first step we need to create a network structure. For example, the MATLAB command `net = feedforwardnet([10,5])` creates a feedforward neural network (MLP) with two hidden layers, where the first hidden layer contains 10 and the second hidden layer contains 5 neurons.
2. As a second step we need to create two matrices $X \in \mathbb{R}^{(HH+1)\times N}$ (input) and $T \in \mathbb{R}^{1\times N}$ (target/output), where N is the number of input-output samples, like defined in equation (3.2).
3. We need to subdivide the sample set (X,T) into a *training set*, a *validation set* and a *test set*. For the training and validation we used all headway trajectories of the 14th and 21st of November 2012, and for the testing some trajectories of the 28th of November 2012. In order to prepare the dataset the `net` structure provides the objects
   - `net.divideFcn = 'divideind'`,
   - `net.divideParam.trainInd`,
   - `net.divideParam.valInd`,
   - `net.divideParam.testInd`.

   We chose 60 % of the data as training set, 20% of the data as validation set, and the testing was performed on the remaining 20% of the dataset. In here we chose these ratios for the data of the 14th and 21st of November 2012, and the results presented in the table 3.4 are computed using the trained network applied only on headway trajectories of the 28th of November.
4. The training of the network is done by the command `net = train(net,X,T)`.
5. The evaluation of the network function for a input matrix X_{test} is then done by the command `net(X_test)`.

   We would like to point out here that it is common to normalize the data before using it for prediction, like described in [Bis95]. This is done automatically by MATLAB's training and evaluation functions.

B.3 MATLAB : ARMA/ARMAX Model

For our numerical experiment we made use of MATLAB's *System-Identification Toolbox*. This one provides the functions `iddata`, which transforms a headway trajectory, given as an equidistantly timestamped vector, into a data format that can be further processed by

the toolbox. The function for fitting an AR/ARX model to the given data is `ar`/`arx`, which gets the data and the AR order p or both the AR order p and the number of exogenous inputs N_U respectively as inputs. The function `armax` fits an ARMAX model to the data. The output of these functions is an `idpoly model`, which can be plugged into the function `forecast` together with the data needed for predicting the future values. The function `forecast` performs the multi-step-ahead prediction and returns the predicted values as output.

Bibliography

[AF12] Eric Albright and Miguel Andres Figliozzi. “Analysis of the impacts of transit signal priority on bus bunching and performance”. In: (2012) (cited on page 11).

[Alt12] Hans Wilhelm Alt. *Lineare Funktionalanalysis: Eine anwendungsorientierte Einführung*. Springer-Verlag, 2012 (cited on pages 16, 17).

[And14] Matthias Andres. “Identifikation linearer Systeme”. bachelorthesis. Technische Universität Kaiserslautern, Mar. 2014 (cited on page 34).

[Bar74] Arnold Barnett. “On controlling randomness in transit operations”. In: *Transportation Science* 8.2 (1974), pages 102–116 (cited on page 10).

[BE12] John J Bartholdi and Donald D Eisenstein. “A self-coördinating bus route to resist bus bunching”. In: *Transportation Research Part B: Methodological* 46.4 (2012), pages 481–491 (cited on page 9).

[BG10] Giuseppe Bellei and Konstantinos Gkoumas. “Transit vehicles’ headway distribution and service irregularity”. In: *Public transport* 2.4 (2010), pages 269–289 (cited on page 8).

[BWL15] Simon J Berrebi, Kari E Watkins, and Jorge A Laval. “A real-time bus dispatching policy to minimize passenger wait on a high frequency route”. In: *Transportation Research Part B: Methodological* 81 (2015), pages 377–389 (cited on page 10).

[Bia+11] James Biagioni, Tomas Gerlich, Timothy Merrifield, and Jakob Eriksson. "Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones". In: *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. ACM. 2011, pages 68–81 (cited on page 5).

[Bil+98] Jeff A Bilmes et al. "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models". In: *International Computer Science Institute* 4.510 (1998), page 126 (cited on page 89).

[BZB06] Yu Bin, Yang Zhongzhen, and Yao Baozhen. "Bus arrival time prediction using support vector machines". In: *Journal of Intelligent Transportation Systems* 10.4 (2006), pages 151–158 (cited on page 6).

[Bis95] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995 (cited on pages 42, 43, 103).

[Bro+12] Ilja N Bronstein, Juraj Hromkovic, Bernd Luderer, Hans-Rudolf Schwarz, Jochen Blath, Alexander Schied, Stephan Dempe, Gert Wanka, Siegfried Gottwald, Eberhard Zeidler, et al. *Taschenbuch der Mathematik*. Volume 1. Springer-Verlag, 2012 (cited on pages 17, 33, 65).

[CD03] FW Cathey and DJ Dailey. "A prescription for transit arrival/departure prediction using automatic vehicle location data". In: *Transportation Research Part C: Emerging Technologies* 11.3 (2003), pages 241–264 (cited on page 5).

[Cha+10] Hyunho Chang, Dongjoo Park, Seungjae Lee, Hosang Lee, and Seungkirl Baek. "Dynamic multi-interval bus travel time prediction using bus transit data". In: *Transportmetrica* 6.1 (2010), pages 19–38 (cited on page 6).

[Cha00] Chris Chatfield. *Time-series forecasting*. CRC Press, 2000 (cited on pages 31, 42, 47).

[Che+04] Mei Chen, Xiaobo Liu, Jingxin Xia, and Steven I Chien. "A dynamic bus-arrival time prediction model based on APC data". In: *Computer-Aided Civil and Infrastructure Engineering* 19.5 (2004), pages 364–376 (cited on page 7).

[Che03] Zhe Chen. "Bayesian filtering: From Kalman filters to particle filters, and beyond". In: *Statistics* 182.1 (2003), pages 1–69 (cited on page 99).

[CK03] Steven I-Jy Chien and Chandra Mouly Kuchipudi. "Dynamic travel time prediction with real-time and historic data". In: *Journal of transportation engineering* 129.6 (2003), pages 608–616 (cited on page 5).

[CPC11] Cathal Coffey, Alexei Pozdnoukhov, and Francesco Calabrese. “Time of arrival predictability horizons for public bus routes”. In: *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Computational Transportation Science*. ACM. 2011, pages 1–5 (cited on page 6).

[Cor+10] Cristián E Cortés, Doris Sáez, Freddy Milla, Alfredo Núñez, and Marcela Riquelme. “Hybrid predictive control for real-time optimization of public transport systems’ operations based on evolutionary multi-objective optimization”. In: *Transportation Research Part C: Emerging Technologies* 18.5 (2010), pages 757–769 (cited on page 10).

[Cyb89] George Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4 (1989), pages 303–314 (cited on page 42).

[Dag09] Carlos F Daganzo. “A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons”. In: *Transportation Research Part B: Methodological* 43.10 (2009), pages 913–921 (cited on pages 9, 21, 52, 53, 55, 62, 63, 67–69, 74).

[DP11] Carlos F Daganzo and Josh Pilachowski. “Reducing bunching with bus-to-bus cooperation”. In: *Transportation Research Part B: Methodological* 45.1 (2011), pages 267–277 (cited on pages 10, 55).

[DAW99] Matthew D’Angelo, Haitham Al-Deek, and Morgan Wang. “Travel-time prediction for freeway corridors”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1676 (1999), pages 184–191 (cited on page 5).

[DMG12] Felipe Delgado, Juan Carlos Munoz, and Ricardo Giesen. “How much can holding and/or limiting boarding improve transit performance?” In: *Transportation Research Part B: Methodological* 46.9 (2012), pages 1202–1217 (cited on page 10).

[Del+09] Felipe Delgado, Juan Muñoz, Ricardo Giesen, and Aldo Cipriano. “Real-time control of buses in a transit corridor based on vehicle holding and boarding limits”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2090 (2009), pages 59–67 (cited on pages 8, 10).

[DM57] Wilfrid J Dixon and Frank J Massey Jr. “Introduction to statistical analysis .” In: (1957) (cited on pages 30, 99).

[EWB01] Xu Jun Eberlein, Nigel HM Wilson, and David Bernstein. “The holding problem with real–time information available”. In: *Transportation science* 35.1 (2001), pages 1–18 (cited on pages 8, 9).

[FM09] Wei Fan and Randy Machemehl. “Do transit users just wait for buses or wait with strategies? Some numerical results that transit planners should see”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2111 (2009), pages 169–176 (cited on pages 9, 62).

[FF11a] Wei Feng and Miguel Figliozzi. “Empirical findings of bus bunching distributions and attributes using archived AVL/APC Bus Data”. In: *Proc., 11th Int. Conf. of Chinese Transportation Professionals (ICCTP)*. ASCE Reston, VA. 2011 (cited on page 9).

[FF11b] Wei Feng and Miguel Figliozzi. “Using archived AVL/APC bus data to identify spatial-temporal causes of bus bunching”. In: *Compendium of papers of 90th Transportation Research Board, Annual Meeting, Washington, DC*. 2011 (cited on page 7).

[Fen06] Donald Fenna. *Cartographic science: a compendium of map projections, with derivations*. CRC Press, 2006 (cited on page 96).

[For73] G David Forney Jr. “The viterbi algorithm”. In: *Proceedings of the IEEE* 61.3 (1973), pages 268–278 (cited on page 90).

[FK98] Luc Frechette and Ata Khan. “Bayesian regression-based urban traffic models”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1644 (1998), pages 157–165 (cited on page 6).

[Fur+03] Peter Gregory Furth, Brendon J Hemily, T Muller, and James G Strathman. *Uses of archived AVL-APC data to improve transit performance and management: Review and potential*. Transportation Research Board Washington, DC, USA, 2003 (cited on page 5).

[GYW16] Vikash V Gayah, Zhengyao Yu, and Jonathan S Wood. “Estimating Uncertainty of Bus Arrival Times and Passenger Occupancies”. In: (2016) (cited on page 7).

[GP09] Carlos Gershenson and Luis A Pineda. “Why does public transport not arrive on time? The pervasiveness of equal headway instability”. In: *PloS one* 4.10 (2009), e7292 (cited on page 11).

[Han+15] Etienne Hans, Nicolas Chiabaut, Ludovic Leclercq, and Robert L Bertini. "Real-time bus route state forecasting using particle filter and mesoscopic modeling". In: *Transportation Research Part C: Emerging Technologies* 61 (2015), pages 121–140 (cited on pages 6, 78).

[He15] Sheng-Xue He. "An anti-bunching strategy to improve bus schedule and headway reliability by making use of the available accurate information". In: *Computers & Industrial Engineering* 85 (2015), pages 17–32 (cited on pages 10, 11).

[Her+15] Daniel Hernández, Juan Carlos Muñoz, Ricardo Giesen, and Felipe Delgado. "Analysis of real-time control strategies in a corridor with multiple bus services". In: *Transportation Research Part B: Methodological* 78 (2015), pages 83–105 (cited on page 11).

[Hic01] Mark D Hickman. "An analytic stochastic model for the transit vehicle holding problem". In: *Transportation Science* 35.3 (2001), pages 215–237 (cited on page 10).

[HVV09] CP IJ van Hinsbergen, JWC Van Lint, and HJ Van Zuylen. "Bayesian committee of neural networks to predict travel times with confidence intervals". In: *Transportation Research Part C: Emerging Technologies* 17.5 (2009), pages 498–509 (cited on page 7).

[Hoc98] Sepp Hochreiter. "The vanishing gradient problem during learning recurrent neural nets and problem solutions". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02 (1998), pages 107–116 (cited on page 43).

[IK74] Edward Ignall and Peter Kolesar. "Optimal dispatching of an infinite-capacity shuttle: Control at a single terminal". In: *Operations Research* 22.5 (1974), pages 1008–1024 (cited on page 10).

[IA00] National Imagery and Mapping Agency. *Department of Defense World Geodetic System 1984 - Its Definition and Relationships with Local Geodetic Systems*. 2000 (cited on page 95).

[IA02] Sherif Ishak and Haitham Al-Deek. "Performance evaluation of short-term time-series traffic prediction model". In: *Journal of Transportation Engineering* 128.6 (2002), pages 490–498 (cited on page 5).

[Jam+13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*. Volume 6. Springer, 2013 (cited on page 88).

[Jeo05] Ran Hee Jeong. “The prediction of bus arrival time using automatic vehicle location systems data”. PhD thesis. Texas A&M University, 2005 (cited on page 6).

[JR05] Ranhee Jeong and Laurence Rilett. “Prediction model of bus arrival time for real-time applications”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1927 (2005), pages 195–204 (cited on page 6).

[JR04] Ranhee Jeong and Laurence R Rilett. “Bus arrival time prediction using artificial neural network model”. In: *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. IEEE. 2004, pages 988–993 (cited on page 6).

[Jia+03] Rui Jiang, Mao-Bin Hu, Bin Jia, and Qing-Song Wu. “Realistic bus route model considering the capacity of the bus”. In: *The European Physical Journal B-Condensed Matter and Complex Systems* 34.3 (2003), pages 367–372 (cited on page 10).

[JH75] JK Jolliffe and TP Hutchinson. “A behavioural explanation of the association between bus and passenger arrivals at a bus stop”. In: *Transportation Science* 9.3 (1975), pages 248–282 (cited on page 9).

[Kno08] Nicholas Knowles. *SIRI Handbook & Functional Service Diagrams*. Kizoom Limited,109-123 Clifton Street, London EC2A 4LD, 2008 (cited on page 91).

[KKP06] SB Kotsiantis, D Kanellopoulos, and PE Pintelas. “Data preprocessing for supervised leaning”. In: *International Journal of Computer Science* 1.2 (2006), pages 111–117 (cited on page 31).

[Kri07] David Kriesel. “A brief Introduction on Neural Networks”. In: (2007) (cited on pages 42, 43).

[LB15] Hoang Thanh Lam and Eric Bouillet. “Flexible sliding windows for kernel regression based bus arrival time prediction”. In: *Machine Learning and Knowledge Discovery in Databases*. Springer, 2015, pages 68–84 (cited on page 6).

[LRT15] Alfred Leick, Lev Rapoport, and Dmitry Tatarnikov. *GPS satellite surveying*. John Wiley & Sons, 2015 (cited on page 17).

[LR09] James Lin and M Ruan. “Probability-based bus headway regularity measure”. In: *Intelligent Transport Systems, IET* 3.4 (2009), pages 400–408 (cited on page 8).

[LB04] Wei-Hua Lin and Robert L Bertini. “Modeling schedule recovery processes in transit operations for bus arrival time prediction”. In: *Journal of Advanced Transportation* 38.3 (2004), pages 347–365 (cited on page 6).

[LZ99] Wei-Hua Lin and Jian Zeng. “Experimental study of real-time bus arrival time prediction with GPS data”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1666 (1999), pages 101–109 (cited on page 5).

[Liz+14] Pedro Lizana, Juan Carlos Muñoz, Ricardo Giesen, and Felipe Delgado. “Bus control strategy application: case study of Santiago transit system”. In: *Procedia Computer Science* 32 (2014), pages 397–404 (cited on page 11).

[MNT04] Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. “Methods for non-linear least squares problems”. In: (2004) (cited on page 43).

[MK04] Kavoussanos Manolis and Daskalakis Kwstis. “Intelligent transportation systems-travelers’ information systems the case of a medium size city”. In: *Mechatronics, 2004. ICM’04. Proceedings of the IEEE International Conference on*. IEEE. 2004, pages 200–204 (cited on page 5).

[MHD99] Ailin Mao, Christopher GA Harrison, and Timothy H Dixon. “Noise in GPS coordinate time series”. In: *Journal of Geophysical Research: Solid Earth* 104.B2 (1999), pages 2797–2816 (cited on page 17).

[MC84] Philippe HJ Marguier and Avishai Ceder. “Passenger waiting strategies for overlapping bus routes”. In: *Transportation Science* 18.3 (1984), pages 207–230 (cited on pages 9, 62).

[MQ13] Qiang Meng and Xiaobo Qu. “Bus dwell time estimation at bus bays: A probabilistic approach”. In: *Transportation Research Part C: Emerging Technologies* 36 (2013), pages 61–71 (cited on page 7).

[Mil08] Martin Milkovits. “Modeling the factors affecting bus stop dwell time: use of automatic passenger counting, automatic fare counting, and automatic vehicle location data”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2072 (2008), pages 125–130 (cited on page 7).

[Mor+16] Luís Moreira-Matias, Oded Cats, João Gama, João Mendes-Moreira, and Jorge Freire de Sousa. “An online learning approach to eliminate Bus Bunching in real-time”. In: *Applied Soft Computing* 47 (2016), pages 460–482 (cited on pages 6, 10).

[Mor+12] Luís Moreira-Matias, Carlos Ferreira, João Gama, João Mendes-Moreira, and Jorge Freire de Sousa. "Bus bunching detection by mining sequences of headway deviations". In: *Advances in Data Mining. Applications and Theoretical Aspects*. Springer, 2012, pages 77–91 (cited on page 7).

[Mor+14] Luís Moreira-Matias, Joao Gama, Joao Mendes-Moreira, and Jorge Freire de Sousa. "An incremental probabilistic model to predict bus bunching in real-time". In: *Advances in intelligent data analysis XIII*. Springer, 2014, pages 227–238 (cited on pages 5, 6).

[Nai+14] Rahul Nair, Eric Bouillet, Yiannis Gkoufas, Olivier Verscheure, Magda Mourad, Farzin Yashar, Rosie Perez, Joel Perez, Gerald Bryant, and Miami Dade Transit. "Data as a resource: real-time predictive analytics for bus bunching". In: (2014) (cited on pages 6, 14, 31, 32, 38, 56).

[New77] GF Newell. "Unstable Brownian motion of a bus trip". In: *Statistical Mechanics and Statistical Methods in Theory and Application*. Springer, 1977, pages 645–667 (cited on page 8).

[New74] Gordon Frank Newell. "Control of pairing of vehicles on a public transportation route, two vehicles, one control point". In: *Transportation Science* 8.3 (1974), pages 248–264 (cited on page 9).

[NP64] Gordon Frank Newell and Renfrey Burnard Potts. "Maintaining a bus schedule". In: *Australian Road Research Board (ARRB) Conference, 2nd, 1964, Melbourne*. Volume 2. 1. 1964 (cited on page 7).

[NK09] Paul Newson and John Krumm. "Hidden Markov map matching through noise and sparseness". In: *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. ACM. 2009, pages 336–343 (cited on pages 18, 24, 25, 87, 89).

[ON72] EE Osuna and GF Newell. "Control strategies for an idealized public transportation system". In: *Transportation Science* 6.1 (1972), pages 52–72 (cited on page 9).

[Pad+10] RPS Padmanaban, K Divakar, Lelitha Vanajakshi, and Shankar C Subramanian. "Development of a real-time bus arrival prediction system for Indian traffic conditions". In: *Intelligent Transport Systems, IET* 4.3 (2010), pages 189–200 (cited on pages 5, 7).

[PCB04] Jayakrishna Patnaik, Steven Chien, and Athanassios Bladikas. "Estimation of bus arrival times using APC data". In: *Journal of public transportation* 7.1 (2004), page 1 (cited on page 6).

[Pil09] Joshua Michael Pilachowski. “An approach to reducing bus bunching”. In: *University of California Transportation Center* (2009) (cited on pages 8, 10, 20).

[Pyl99] Dorian Pyle. *Data preparation for data mining*. Volume 1. Morgan Kaufmann, 1999 (cited on page 31).

[ElR02] Ahmed El-Rabbany. *Introduction to GPS: the global positioning system*. Artech House, 2002 (cited on page 94).

[Rab89] Lawrence R Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition”. In: *Proceedings of the IEEE* 77.2 (1989), pages 257–286 (cited on pages 87, 88).

[RV04] John Rice and Erik Van Zwet. “A simple and effective method for predicting travel times on freeways”. In: *Intelligent Transportation Systems, IEEE Transactions on* 5.3 (2004), pages 200–207 (cited on page 6).

[Sáe+12] Doris Sáez, Cristián E Cortés, Freddy Milla, Alfredo Núñez, Alejandro Tirachini, and Marcela Riquelme. “Hybrid predictive control strategy for a public transport system with uncertain demand”. In: *Transportmetrica* 8.1 (2012), pages 61–86 (cited on page 10).

[SKW16] GE Sánchez-Martínez, HN Koutsopoulos, and NHM Wilson. “Real-time holding control for high-frequency transit with dynamics”. In: *Transportation Research Part B: Methodological* 83 (2016), pages 1–19 (cited on page 10).

[Sch+16] Jan-Dirk Schmöcker, Wenzhe Sun, Achille Fonzone, and Ronghui Liu. “Bus bunching along a corridor served by two lines”. In: *Transportation Research Part B: Methodological* 93 (2016), pages 300–317 (cited on page 11).

[SF04] Amer Shalaby and Ali Farhan. “Prediction model of bus arrival and departure times using AVL and APC data”. In: *Journal of Public Transportation* 7.1 (2004), page 3 (cited on page 6).

[Sin+12] Mathieu Sinn, Ji Won Yoon, Francesco Calabrese, and Eric Bouillet. “Predicting arrival times of buses using real-time GPS measurements”. In: *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE. 2012, pages 1227–1232 (cited on pages 6, 38).

[Sny87] John Parr Snyder. *Map projections–A working manual*. Volume 1395. US Government Printing Office, 1987 (cited on pages 95, 96).

[Son+04] Bongsoo Son, Hyung Jin Kim, Chi-Hyun Shin, and Sang-Keon Lee. “Bus arrival time prediction method for ITS application”. In: *Knowledge-Based Intelligent Information and Engineering Systems*. Springer. 2004, pages 88–94 (cited on page 6).

[Son13] Eduardo D Sontag. *Mathematical control theory: deterministic finite dimensional systems*. Volume 6. Springer Science & Business Media, 2013 (cited on page 63).

[SH05] Aichong Sun and Mark Hickman. “The real–time stop–skipping problem”. In: *Journal of Intelligent Transportation Systems* 9.2 (2005), pages 91–109 (cited on page 11).

[Sun+07] Dihua Sun, Hong Luo, Liping Fu, Weining Liu, Xiaoyong Liao, and Min Zhao. “Predicting bus arrival time on the basis of global positioning system data”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2034 (2007), pages 62–72 (cited on page 7).

[TJ08] Dalia Tiesyte and Christian S Jensen. “Similarity-based prediction of travel times for vehicles traveling on known routes”. In: *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. ACM. 2008, page 14 (cited on page 6).

[VR07] Lelitha Vanajakshi and Laurence Rilett. “Support vector machine technique for the short term prediction of travel time”. In: *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE. 2007, pages 600–605 (cited on page 6).

[VDE16] David Verbich, Ehab Diab, and Ahmed El-Geneidy. “Have they bunched yet? An exploratory study of the impacts of bus bunching on dwell and 1 running times 2”. In: *running times* 2 (2016), page 3 (cited on page 9).

[Vin75] Thaddeus Vincenty. “Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations”. In: *Survey review* 23.176 (1975), pages 88–93 (cited on page 96).

[WB06] Greg Welch and Gary Bishop. *An introduction to the kalman filter. Department of Computer Science, University of North Carolina*. 2006 (cited on page 99).

[XAD11] Yiguang Xuan, Juan Argote, and Carlos F Daganzo. “Dynamic bus holding strategies for schedule reliability: Optimal linear control and performance analysis”. In: *Transportation Research Part B: Methodological* 45.10 (2011), pages 1831–1845 (cited on pages 9, 10, 52, 69–71).

[YLT11] Bin Yu, William HK Lam, and Mei Lam Tam. “Bus arrival time prediction at bus stop with multiple routes”. In: *Transportation Research Part C: Emerging Technologies* 19.6 (2011), pages 1157–1170 (cited on pages 5, 7).

[Yu+10] Bin Yu, Zhong-Zhen Yang, Kang Chen, and Bo Yu. “Hybrid model for prediction of bus arrival times at next station”. In: *Journal of Advanced Transportation* 44.3 (2010), pages 193–204 (cited on pages 5, 7).

[ZDB06] Jiamin Zhao, Maged Dessouky, and Satish Bukkapatnam. “Optimal slack time for schedule-based transit operations”. In: *Transportation Science* 40.4 (2006), pages 529–539 (cited on page 8).

[ZZL12] Pengfei Zhou, Yuanqing Zheng, and Mo Li. “How long to wait?: predicting bus arrival time with mobile phone based participatory sensing”. In: *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM. 2012, pages 379–392 (cited on page 5).

[Zho+16] Yuyang Zhou, Lin Yao, Yanyan Chen, and Yi Gong. “Bus Arrival Time Calculation Model Based on Smart Card Data”. In: *Transportation Research Board 95th Annual Meeting*. 16-5049. 2016 (cited on page 5).

[ZAJ04] Saeed Zolfaghari, Nader Azizi, and Mohamad Y Jaber. “A model for holding strategy in public transit systems with real-time information”. In: *International Journal of Transport Management* 2.2 (2004), pages 99–110 (cited on page 9).

Web Ressources

Dublin Bus. Retrieved Apr 2016, from `https://www.dublinbus.ie/RTPI/Sources-of-Real-Time-Information/`

Dublinked. Retrieved Apr 2016, from `http://dublinked.ie/`

Dublinked. Retrieved Apr 2016, from `https://data.dublinked.ie/dataset/dublin-bus-gps-sample-data-from-dublin-city-council-insight-project`

Dublinked. Retrieved Apr 2016, from `https://data.dublinked.ie/dataset/dublin-bus-gtfs-data`

General Transit Feed Spec Changes. Google Sites. Retrieved Apr 2016, from `https://sites.google.com/site/gtfschanges/proposals/route-type`

Geographx. Retrieved May 2016, from `http://geographx.co.nz/_wp/wp-content/uploads/2012/01/cylindrical.jpg`

Google Developers. Retrieved Apr 2016, from https://developers.google.com/transit/gtfs/

GPS. Retrieved May 2016, from http://www.gps.gov/systems/gps/

Latex Templates. Retrieved Jul 2016, from http://www.latextemplates.com/

Mathworks. Retrieved Apr 2016, from http://www.mathworks.com/matlabcentral/fileexchange/278-arrow/content/arrow.m

Mathworks. Retrieved Apr 2016, from http://de.mathworks.com/matlabcentral/fileexchange/5379-geodetic-distance-on-wgs84-earth-ellipsoid/content/vdist.m

Mathworks. Retrieved Apr 2016, from http://www.mathworks.com/matlabcentral/fileexchange/22022-matlab2tikz-matlab2tikz

SIRI. Retrieved Apr 2016, from www.siri.org.uk

The Journal. Retrieved Aug 2016, from http://www.thejournal.ie/dubiin-bus-strike-2919690-Aug2016

Wikipedia. Retrieved May 2016, from https://de.wikipedia.org/wiki/Viterbi-Algorithmus

Wikipedia. Retrieved Apr 2016, from https://en.wikipedia.org/wiki/Google_Maps

Wikipedia. Retrieved Jun 2016, from https://en.wikipedia.org/wiki/Universal_approximation_theorem

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W