Temporal and Spatial Models for Temperature Estimation Using Vehicle Data

Lisa Eriksson
Master of Science Thesis in Electrical Engineering

Temporal and Spatial Models for Temperature Estimation Using Vehicle Data:

Lisa Eriksson
LiTH-ISY-EX–19/5241–SE

Supervisor: Magnus Malmström
ISY, Linköping University

Lage Ragnarsson
NIRA Dynamics AB

Examiner: Fredrik Gustafsson
ISY, Linköping University

Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Copyright © 2019 Lisa Eriksson
Abstract

Safe driving is a topic of multiple factors where the road surface condition is one. Knowledge about the road status can for instance indicate whether it is risk for low friction and thereby help increase the safety in traffic. The ambient temperature is an important factor when determining the road surface condition and is therefore in focus.

This work evaluates different methods of data fusion to estimate the ambient temperature at road segments. Data from vehicles are used during the temperature estimation process while measurements from weather stations are used for evaluation. Both temporal and spatial dependencies are examined through different models to predict how the temperature will evolve over time. The proposed Kalman filters are able to both interpolate in road segments where many observations are available and to extrapolate to road segments with no or only a few observations. The results show that interpolation leads to an average error of 0.5 degrees during winter when the temperature varies around five degrees day to night. Furthermore, the average error increases to two degrees during springtime when the temperature instead varies about fifteen degrees per day.

It is shown that the risk of large estimation error is high when there are no observations from vehicles. As a separate result, it has been noted that the weather stations have a bias compared to the measurements from the cars.
Acknowledgments

First of all, I would like to thank my supervisor Lage Ragnarsson at NIRA Dynamics AB for all the support during this work. Your numerous ideas and enthusiasm have encouraged me to investigate different approaches and to see the solutions and results from multiple perspectives.

I would also like to thank my examiner Fredrik Gustafsson for all input along the way. Your helpfulness regardless time of the day has been truly valuable. Furthermore, thank you to my supervisor Magnus Malmström at Linköping University for all feedback regarding my thesis throughout these months.

Lastly, I would like to thank my family for always believing in me during all my years at the university.

Linköping, June 2019
Lisa Eriksson
Contents

Notation xi

1 Introduction 1
1.1 Background 1
1.2 Purpose 2
1.3 Problem Formulation 3
1.4 Related Work 3

2 Theory 5
2.1 State-space Model 5
  2.1.1 Sample Time Dependency 6
  2.1.2 Constant Velocity Model 6
  2.1.3 Singer Model 7
  2.1.4 Measurement Quantization 7
2.2 Kalman Filter 7
  2.2.1 Handling Missing Observations 8
  2.2.2 Square Root Implementation 8
  2.2.3 Extended Kalman Filter 11
2.3 Fusion of Multiple Observations 11
2.4 Road Representations in Map Services 12
  2.4.1 Segmentation of Maps 12
  2.4.2 Segmentation of Roads 13
2.5 Evaluation Methods 14
  2.5.1 Average Absolute Error 14
  2.5.2 Root Mean Square Error 14
  2.5.3 Evaluating Weather Forecasts 14

3 Method 15
3.1 Data Description 15
3.2 Time Discretization 16
  3.2.1 Timestamp Vector Generation 16
3.3 Dataset Generation 16
  3.3.1 Tile Level 17
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.2</td>
<td>Link Level</td>
<td>17</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Sublink Level</td>
<td>18</td>
</tr>
<tr>
<td>3.4</td>
<td>Temperature Models</td>
<td>18</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Constant Position Model</td>
<td>18</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Constant Velocity Model</td>
<td>18</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Constant Velocity Singer Model</td>
<td>20</td>
</tr>
<tr>
<td>3.5</td>
<td>Fusion of Multiple Observations</td>
<td>20</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Dependent Observations</td>
<td>20</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Independent Observations</td>
<td>20</td>
</tr>
<tr>
<td>3.6</td>
<td>Handling Missing Observations</td>
<td>21</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Complete Time Update</td>
<td>21</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Incomplete Time Update</td>
<td>22</td>
</tr>
<tr>
<td>3.7</td>
<td>Spatial Dependency Models</td>
<td>22</td>
</tr>
<tr>
<td>3.7.1</td>
<td>No Spatial Dependency</td>
<td>23</td>
</tr>
<tr>
<td>3.7.2</td>
<td>Distance Impact</td>
<td>23</td>
</tr>
<tr>
<td>3.7.3</td>
<td>Precalculated Correlation Matrix</td>
<td>24</td>
</tr>
<tr>
<td>3.8</td>
<td>Estimation Process</td>
<td>26</td>
</tr>
<tr>
<td>3.9</td>
<td>Modified Kalman Filtering Algorithm</td>
<td>28</td>
</tr>
<tr>
<td>3.10</td>
<td>Numerical Instability</td>
<td>30</td>
</tr>
<tr>
<td>3.11</td>
<td>Evaluation</td>
<td>32</td>
</tr>
<tr>
<td>3.11.1</td>
<td>Dataset Analysis</td>
<td>32</td>
</tr>
<tr>
<td>3.11.2</td>
<td>Extraction of Ground Truth Data</td>
<td>33</td>
</tr>
<tr>
<td>3.11.3</td>
<td>Error Metric</td>
<td>33</td>
</tr>
<tr>
<td>3.11.4</td>
<td>Consideration of Bias</td>
<td>33</td>
</tr>
<tr>
<td>4</td>
<td>Results</td>
<td>35</td>
</tr>
<tr>
<td>4.1</td>
<td>Time Period with Small Temperature Changes</td>
<td>35</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Temperature Models</td>
<td>35</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Datasets and Spatial Models</td>
<td>39</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Bias Consideration</td>
<td>41</td>
</tr>
<tr>
<td>4.2</td>
<td>Time Period with Large Temperature Changes</td>
<td>42</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Temperature Models</td>
<td>42</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Datasets and Spatial Models</td>
<td>46</td>
</tr>
<tr>
<td>5</td>
<td>Discussion</td>
<td>49</td>
</tr>
<tr>
<td>5.1</td>
<td>Results</td>
<td>49</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Temperature Models</td>
<td>49</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Datasets</td>
<td>50</td>
</tr>
<tr>
<td>5.1.3</td>
<td>Spatial Dependency Models</td>
<td>51</td>
</tr>
<tr>
<td>5.1.4</td>
<td>Consideration of Bias</td>
<td>52</td>
</tr>
<tr>
<td>5.2</td>
<td>Parameter Selection</td>
<td>53</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions</td>
<td>55</td>
</tr>
<tr>
<td>6.1</td>
<td>Answers to Problem Formulations</td>
<td>55</td>
</tr>
<tr>
<td>6.2</td>
<td>Future Work</td>
<td>56</td>
</tr>
<tr>
<td>A</td>
<td>Pearson Sample Correlation</td>
<td>61</td>
</tr>
<tr>
<td>Contents</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>B    Haversine Formula</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>C    Hour-wise Error per Temperature Model</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>D    Bias Investigation</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>Bibliography</td>
<td>71</td>
<td></td>
</tr>
</tbody>
</table>
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAE</td>
<td>Average absolute error</td>
</tr>
<tr>
<td>CP</td>
<td>Constant position model</td>
</tr>
<tr>
<td>CV</td>
<td>Constant velocity model</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman filter</td>
</tr>
<tr>
<td>EMD</td>
<td>Empirical mode decomposition</td>
</tr>
<tr>
<td>ID</td>
<td>Identification number</td>
</tr>
<tr>
<td>LS-SVM</td>
<td>Least squares support vector machine</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root mean squared error</td>
</tr>
</tbody>
</table>
| RWIS         | Road weather information system  
  (VViS, vägväderinformationssystem in Swedish) |
| SVD          | Singular value decomposition |
| SVM          | Support vector machine |
Introduction

This chapter introduces the background to the thesis, describes the purpose and presents the problem formulations. In addition, some related work in the area is mentioned.

1.1 Background

The issue of safe driving includes several components, where one regards the road condition [24]. During the cold months of the year there is an increased risk of worse road conditions. Water on the road that freezes or snowfall can lead to low friction between the tires and the road. If it is possible to determine road conditions, then this could contribute to increased safety in the traffic by making drivers aware of where it is low friction.

The road status is relevant within the area of autonomous driving as well. An autonomous car has to be aware of the road condition in order to drive safely. It is important that the speed is adjusted to the road friction and not just the traffic signs. Furthermore, distance to the car in front is another factor that has to be adjusted due to longer break path during low friction. Hence, determining the road condition is important to enable autonomous cars during all seasons [13].

Another use case of determining the road surface condition is within maintenance. Roads require snow clearing and salt spreading during the winter. If it exists information of which roads that have a higher risk of low friction, then these can be prioritized during the maintenance work. This could both improve the efficiency and prevent traffic accidents due to that the most dangerous roads are maintained first [13].
Low friction warnings can nowadays be generated using weather stations called RWIS (road weather information system) and knowledge about the climate. Some tests to combine these information sources with friction data from cars have also been initiated [22]. A RWIS measures for example the humidity, road temperature, ambient temperature and precipitation. There are only 775 RWIS around Sweden which restricts the coverage [23]. However, the RWIS are placed at locations where the temperature drops quickly to detect cold weather as fast as possible. It can for example be in places where shading is substantial, at bridges or in valleys [2]. The location of some RWIS around Stockholm are shown in Figure 1.1. If information from cars can be included to chart hazardous road surface conditions then there is a possibility to estimate a more precise and wider ranged road status than if only RWIS and general climate understanding are used.

1.2 Purpose

This work aims to evaluate different methods of data fusion to estimate the state of road segments using data from vehicles. The focus in this work is ambient temperature at road segments instead of friction which was mentioned in Section 1.1. This is the case because ambient temperature is an important factor when determining the road condition. In addition, there are several weather stations called RWIS [23] placed around Sweden which measurements then can be used as ground truth. However, the methods investigated are general and a continuation could be to use them on other signals than temperature, such as friction data.
To estimate temperature, it is helpful if it exists a model of temperature change over time. Therefore, three different approaches are examined in this work. These are combined with three different spatial dependency models. The spatial dependency models adjust how nearby measurements are impacting the temperature estimation of a specific road segment. The temperature models are described in Section 3.4 while the spatial dependency models are described in Section 3.7.

1.3 Problem Formulation

The problem formulations to be answered are the following:

1. What model can be used for temperature estimation?

2. Does a local dataset of measurements lead to a more accurate temperature estimation than a dataset containing measurements from a wider area?

3. Which spatial dependency model leads to the most accurate temperature estimation?

The first problem formulation aims to give an answer to which model that should be used to model temperature change over time. Which model contributes to as accurate temperature estimation as possible?

The second problem formulation regards which measurements that should be used during temperature estimation of a specific road segment. Is it preferable to use a few measurements close to the sublink which temperature is estimated? Or is it better to use a larger amount of measurements but with the constraint that some of these are located further away from the sublink?

The last problem formulation is about how the measurements in the chosen dataset shall impact the temperature estimation. Which spatial dependency model generates the most accurate result?

1.4 Related Work

The relationship between car measurements and data from RWIS has been investigated in [3]. Measurements from cars and RWIS have been plotted against each other to determine potential differences. The conclusions made were that the RWIS in general reports a temperature around one degree lower than the cars. Furthermore, there was a larger difference between the cars’ measurements and the data from RWIS during the summer when the temperature is higher than during the autumn.

A similar study of differences between measurements from cars and measurements from weather stations has been performed in [2]. In addition, this study includes a comparison of the measurements with respect to distance between RWIS and the car which reported a measurement. The conclusions were that the weather stations in general measures low ambient temperature compared to the
cars. However, car measurements close to the RWIS do not differ as many degrees as car measurements further away. In that work far away is defined as further than 1.6 kilometers away and close is defined as within 0.6 kilometers from the weather station.

When it comes to prediction of weather and more precisely temperature there are multiple approaches to solve the problem. Radhika and Shashi propose one method in [17] where they use support vector machines, SVM, to predict the maximum temperature per day. A period of five years was used to train the model and it was then evaluated for data received during approximately six months. Preprocessing of the data was performed to decrease the impact of missing data. According to that work, the model’s performance is vulnerable when it comes to missing data. If there is no temperature value available for a day then it is assigned the mean maximum temperature value for the regarding month.

Another approach to predict temperature has been proposed in [5]. It is based on both empirical mode decomposition, EMD, and least squares support vector machine, LS-SVM. In short, the method consists of three phases. First, the signal is decomposed using EMD. Second, multiple different forecast models, created using LS-SVM, process the decomposed signal which results in different predictions. The final phase makes a combined prediction based on all forecasts in step two.
This chapter presents the theory needed to understand the methods used in this work. All methods are based on the state space model presented in Section 2.1 and the Kalman filter introduced in Section 2.2. Appendices A and B contain further theory of some mathematical concepts for deeper understanding.

## 2.1 State-space Model

A state-space model aims to describe the behavior of a system. The discrete state-space model with time variance is defined according to

\[
\begin{align*}
    x_k &= F_k x_{k-1} + G_k u_k + \epsilon_k \quad (2.1a) \\
    z_k &= H_k x_k + \delta_k \quad (2.1b) \\
    x_0 &= \mu_0 + \tau_0. \quad (2.1c)
\end{align*}
\]

Let \( \mathcal{N}(\mu, Q) \) denote a Gaussian distribution with mean \( \mu \) and covariance matrix \( Q \). Then let

\[
\begin{align*}
    \epsilon_k &\sim \mathcal{N}(0, Q_k) \\
    \delta_k &\sim \mathcal{N}(0, R_k) \\
    \tau_0 &\sim \mathcal{N}(0, P_0).
\end{align*}
\]

Variable \( x \) is the state vector, \( u \) is control actions and \( z \) is a vector with observations. Furthermore, \( F \) and \( G \) describe the transition from a previous state to the next and \( H \) defines the relationship between the actual state \( x \) and the observation \( z \). The first state is denoted \( x_0 \) and \( \mu_0 \) is the mean of the initial value. Noise is represented by \( \epsilon, \delta \) and \( \tau \), which are all normally distributed with zero mean, but with the different covariance matrices \( Q, R \) and \( P \) [21].
2.1.1 Sample Time Dependency

Consider a time-continuous model

\[ \dot{x}(t) = Ax(t) + Bu(t) \]  \hspace{1cm} (2.2a)
\[ z(t) = Cx(t) \]  \hspace{1cm} (2.2b)

This model can be discretized by introducing a sample time-dependency \[ T_{s,k} \]. This discrete, sample time dependent model is defined as

\[ x_k = F(T_{s,k})x_{k-1} + G(T_{s,k})u_k \]  \hspace{1cm} (2.3a)
\[ z_k = Hx_k \]  \hspace{1cm} (2.3b)
\[ T_{s,k} = t_k - t_{k-1} \]  \hspace{1cm} (2.3c)

where \( T_{s,k} \) is the time between two observations. Variable \( t_k \) is the time for the current sample and \( t_{k-1} \) is the time for the previous sample. Observe that \( T_{s,k} \) is not constant due to irregular sample times for the observations. Hence, the model is only updated for the time steps \( T_{s,k} \), which makes it discrete instead of continuous. The sample time dependency is also included in the calculation of the a priori covariance matrix \( \tilde{P} \) (see line 1 in Algorithm 1) by insertion in the process noise covariance matrix \( Q \). An example of how the sample time can be used to modify the process noise covariance matrix is presented in Section 2.1.2.

2.1.2 Constant Velocity Model

A model where the state vector is one-dimensional and represented by a position is called the constant position model, \( \text{cp} \). An extension of this one is the so-called constant velocity model, \( \text{cv} \), where the state vector \( x \) consists of a position and a constant velocity \[ \text{7} \].

The process noise covariance matrix \( Q_k \) represents the uncertainty of the system model \[ \text{7} \]. Covariance matrix \( Q_k \) has been derived by Reid \[ 18 \] under the assumption of a discrete constant velocity model impacted by Gaussian noise according to

\[ Q_k = \text{Var}(\epsilon) = E[\epsilon_k \epsilon_k^T] - E[\epsilon_k]E[\epsilon_k] = \begin{bmatrix} \epsilon_k \text{ Gaussian noise} \\ \text{with mean 0} \end{bmatrix} = \\ E[\epsilon_k \epsilon_k^T] = \cdots = \begin{bmatrix} T_{s,k}^3 \\ \frac{T_{s,k}^2}{2} \\ \frac{T_{s,k}}{2} \\ T_{s,k} \end{bmatrix} \sigma_Q^2. \]  \hspace{1cm} (2.4)

The method used was to integrate the process noise over a period of one sample time \( T_{s,k} \). The result shown in (2.4) uses the notation used in Section 2.2 instead of Reid’s. The notation \( \text{Var}(x) \) is the variance of variable \( x \) and \( E[x] \) denotes the expected value of \( x \). The resulting matrix depends on the process noise standard deviation \( \sigma_Q \) and the sample time \( T_{s,k} \) \[ 18 \].
2.1.3 Singer Model

Let the state vector $x$ consist of a position, velocity and acceleration. The idea of the Singer model is to include a regularization, $a$, to the acceleration term according to

$$\dot{a}(t) = -\alpha a(t) + w(t), \quad \alpha > 0. \quad (2.5)$$

Variable $a$ is the acceleration and $w$ is zero mean Gaussian noise [19].

2.1.4 Measurement Quantization

When the measurements, used as observations in $z$, are quantized it leads to a quantization noise. This noise’s variance, $\sigma_n^2$, can be modeled according to

$$\sigma_n^2 = \frac{\Delta^2}{12} \quad (2.6)$$

under the assumption that the noise is additive uniform. Variable $\Delta$ is the quantization step [10]. This quantization noise can be included in the system model by adding the variance $\sigma_n^2$ to the measurement noise covariance $R_k$ [10].

2.2 Kalman Filter

Kalman filters aim to predict a state given observations and a model of the system. The notation introduced by Thrun, Burgard and Fox [21] will be the basis when describing the Kalman filter algorithm.

**Algorithm 1:** Kalman filter algorithm

1. **Time update:**
   \[ \overline{x}_k = F_k x_{k-1} + G_k u_k \]
   \[ \overline{P}_k = F_k P_{k-1} F_k^T + Q_k \]

2. **Calculate Kalman gain:**
   \[ K_k = \overline{P}_k H_k^T (H_k \overline{P}_k H_k^T + R_k)^{-1} \]

3. **Measurement update:**
   \[ x_k = \overline{x}_k + K_k (z_k - H_k \overline{x}_k) \]
   \[ P_k = (I - K_k H_k) \overline{P}_k \]

   **Return:** $x_k$, $P_k$

The Kalman filter algorithm is presented in Algorithm 1. Predicted value $\overline{x}$ and covariance $\overline{P}$ are calculated for each time instance. Correction is then performed with help from the calculated Kalman gain, $K$, which results in a posteriori value $x$ and covariance $P$ [21].
2.2.1 Handling Missing Observations

Kalman filters rely on receiving observations for correcting its predictions. In a time variant system, where there is a risk for intermitted measurements, the filter has to be modified to handle this in a suitable way. Multiple proposals have been made [12], but the one in focus here is the Bernoulli process [15].

The Bernoulli process approach is to introduce the binary parameter $\gamma$. A value of one indicates that an observation has been received and value zero indicates no present observation for the time in question. This leads to a change in the calculation of a posteriori value $x$ and covariance $P$ in the Kalman algorithm (see lines 3-3 in Algorithm 1). The new equations become

$$x_k = \bar{x}_k + \gamma K_k (z_k - H_k \bar{x}_k)$$

$$P_k = (I - \gamma K_k H_k) \bar{P}_k.$$  \hspace{1cm} (2.7a) \hspace{1cm} (2.7b)

Insertion of $\gamma$ leads to that both $x$ and $P$ equal their corresponding predicted values when no observation is available for correction [15].

2.2.2 Square Root Implementation

The covariance matrix, $P$, created during each iteration of the Kalman filtering should be symmetric and positive definite [7]. A real matrix $A$ is positive definite if

$$u^T A u > 0$$

holds. Variable $u$ is a non-zero vector belonging to the complex vector space $\mathbb{C}^n$ of dimension $n$ [26]. When the Kalman filter is implemented and used in practice it can lead to that the covariance matrix ends up with negative variances, which is theoretically impossible, or that it does not become positive definite due to numerical issues. The square root implemented Kalman filter is more robust against these kinds of problems than the original Kalman filter presented in Section 2.2 [7].

The idea with the square root implemented Kalman filter is to propagate the square root of the covariance matrix $P$. This leads to more precision than if the covariance is being propagated. Let $P^{1/2}$ denote the square root of $P$. Then the relation $P = P^{1/2} P^{1/2}$ holds. The square root form of the Kalman filter can either be divided into a time update and a measurement update, like the original Kalman filter in Section 2.2, or these can be combined to only one step. The time update step is presented in Algorithm 2. The complete version of the square root Kalman filter is presented in Algorithm 3, where the time update step has been merged with the measurement update step. Note that a priori estimate $\bar{x}$ and covariance matrix $\bar{P}$ are returned from the time update of the square root Kalman filter. On the other hand, the a posteriori estimate $x$ and covariance matrix $P$ are returned from the measurement update of the square root Kalman filter and the complete square root Kalman filter [7].
Algorithm 2: Time update of the square root Kalman filter

1. Calculate the square root of $P_{k-1}$ and $Q_k$:
   \[ P_{k-1} = P_{k-1}^{1/2} P_{k-1}^{T/2} \]
   \[ Q_k = Q_k^{1/2} Q_k^{T/2} \]

2. Create the following matrix:
   \[ U_k = \begin{bmatrix} F_k P_{k-1}^{1/2} & G_k Q_k^{1/2} \end{bmatrix} \]

3. Perform QR factorization on $U_k$:
   \[ qr(U_k) = R_{qr,k}^T Q_{qr,k}^T = \begin{bmatrix} \bar{P}_k^{1/2} & 0 \end{bmatrix} Q_{qr,k}^T \]

4. Extract $\bar{P}_k^{1/2}$ from $R_{qr,k}$

5. Update the state estimate:
   \[ \bar{x}_k = F_k x_{k-1} \]

6. Create the covariance matrix from $\bar{P}_k^{1/2}$:
   \[ \bar{P}_k = \bar{P}_k^{1/2} \bar{P}_k^{T/2} \]

Return: $\bar{x}_k, \bar{P}_k$

The first stage of the square root Kalman filter in Algorithm 3 is to calculate the square roots of all covariance matrices, see step 1. Cholesky decomposition can be used to calculate the square roots due to that the covariance matrices are symmetric and positive definite. The Cholesky decomposition of matrix $A$ is given by

\[ A = LL^T \]  \hspace{1cm} (2.9)

where variable $L$ is a lower triangular matrix which represents the square root of matrix $A$ [25].

The second step is to form the matrix here denoted $U$. It contains the previously calculated square roots of the covariance matrices as well as the $F$, $G$ and $H$ matrices defined in Section 2.2 [7].

The third step is to perform QR factorization on $U$. This means that $U$ is decomposed into two matrices here denoted $R_{rq}$ and $Q_{rq}$. Matrix $R_{rq}$ is an upper triangular matrix while $Q_{rq}$ is an orthogonal matrix [14].

The fourth step is to identify the square root of the covariance matrix, $P_k^{1/2}$, the square root of the so-called innovation term, $S_k^{1/2}$, and the term which is the Kalman gain $K_k$ multiplied with $S_k^{1/2}$. All these are present in the $R_{qr,k}$ matrix which can be seen in the third step of Algorithm 3.
Algorithm 3: Square root Kalman filter

1. Calculate the square root of $P_{k-1}$, $Q_k$ and $R_k$:
   
   \[
   P_{k-1} = P_{k-1}^{1/2} P_{k-1}^{T/2} \\
   Q_k = Q_k^{1/2} Q_k^{T/2} \\
   R_k = R_k^{1/2} R_k^{T/2}
   \]

2. Create the following matrix:
   \[
   U_k = \begin{bmatrix}
   R_k^{1/2} & H_k F_k P_{k-1}^{1/2} & H_k G_k Q_k^{1/2} \\
   0 & F_k P_{k-1}^{1/2} & G_k Q_k^{1/2}
   \end{bmatrix}
   \]

3. Perform QR factorization on $U_k$:
   \[
   qr(U_k) = R_{qr,k}^T Q_{qr,k}^T = \begin{bmatrix}
   S_k^{1/2} & 0 & 0 \\
   K_k S_k^{1/2} & P_k^{1/2} & 0
   \end{bmatrix} Q_{qr,k}^T
   \]

4. Extract $S_k^{1/2}$, $K_k S_k^{1/2}$ and $P_k^{1/2}$ from $R_{qr,k}$

5. Update the state estimate:
   \[
   x_k = F_k x_{k-1} + K_k S_k^{1/2} S_k^{-1/2} (z_k - H_k F_k x_{k-1})
   \]

6. Create the covariance matrix from $P_k^{1/2}$:
   \[
   P_k = P_k^{1/2} P_k^{T/2}
   \]

Return: $x_k$, $P_k$

When all needed components have been extracted from $R_{qr,k}$, it is time for the state estimate update. This is the fifth step of the square root Kalman filter. The result from this part of the algorithm is the estimated state $x_k$. Note that the update of $x$ in the time update step and in the measurement update step, in the original Kalman filter (see Algorithm 1), have been combined here [7].

The sixth and final step of the square root Kalman filter is to compute the covariance matrix $P_k$ with help from the square root. This procedure, when multiplying the square root with the transposed square root, generates a symmetric and positive definite matrix as long as the square root matrix is of full rank [11].
2.2.3 Extended Kalman Filter

A modification of the Kalman filter is the so-called extended Kalman filter, EKF. This variant allows non-linear relationships between state vectors \( \mathbf{x}_k \) and \( \mathbf{x}_{k-1} \) as well as between the observation \( \mathbf{z} \) and state \( \mathbf{x} \), see (2.10). The EKF linearizes the non-linearity [7].

\[
\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{e}_k) \quad (2.10a)
\]
\[
\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{d}_k) \quad (2.10b)
\]

These non-linear relationships lead to some modifications in the Kalman algorithm. Algorithm 4 presents the extended Kalman algorithm when only the relation between observation \( \mathbf{z} \) and estimated state \( \mathbf{x} \) is non-linear [7].

**Algorithm 4: Extended Kalman filter algorithm**

1. **Time update:**
   \[
   \mathbf{\bar{x}}_k = F_k \mathbf{x}_{k-1} + G_k \mathbf{u}_k
   \]
   \[
   \mathbf{P}_k = F_k \mathbf{P}_{k-1} F_k^T + Q_k
   \]

2. **Calculate Kalman gain:**
   \[
   \mathbf{K}_k = \mathbf{P}_k (h'(\mathbf{\bar{x}}_k))^T (h'(\mathbf{\bar{x}}_k) \mathbf{P}_k (h'(\mathbf{\bar{x}}_k))^T + \mathbf{R}_k)^{-1}
   \]

3. **Measurement update:**
   \[
   \mathbf{x}_k = \mathbf{\bar{x}}_k + \mathbf{K}_k (\mathbf{z}_k - h(\mathbf{\bar{x}}_k))
   \]
   \[
   \mathbf{P}_k = (I - \mathbf{K}_k h'(\mathbf{x}_k)) \mathbf{P}_k
   \]

   Return: \( \mathbf{x}_k, \mathbf{P}_k \)

2.3 Fusion of Multiple Observations

Gan and Harris [6] mention two different ways of measurement fusion. The first one increases the size of the observation vector \( \mathbf{z} \), the \( \mathbf{C} \) matrix and the measurement covariance matrix \( \mathbf{R} \) to include multiple observations in the filtering process. Equations (2.11a)-(2.11c) illustrate this method, where \( N \) denotes the number of observations for the time in question [6].

\[
\mathbf{z}(t) = [\mathbf{z}_1(t) \cdots \mathbf{z}_N(t)]^T \quad (2.11a)
\]
\[
\mathbf{C}(t) = [\mathbf{C}_1(t) \cdots \mathbf{C}_N(t)]^T \quad (2.11b)
\]
\[
\mathbf{R}(t) = \text{diag} (\mathbf{R}_1(t), \ldots, \mathbf{R}_N(t)) \quad (2.11c)
\]

Hence, this solution takes advantage of the sample time dependency model (see Section 2.1.1) and lets \( T_k = 0 \) for those samples which belong to the same time stamp.
The second measurement fusion method combines the measurements by a weighting process leading to only one measurement being used in the filtering process. Likewise, the $C$ matrices and measurement covariance matrices $R$ are also combined leading to only one of each. This method is summarized in (2.12), where $N$ is the number of observations for the regarding time. Note that this method is only valid if the measurements are independent [6].

$$z(t) = \left[\sum_{j=1}^{N} R_j^{-1}(t)\right]^{-1} \sum_{j=1}^{N} R_j^{-1}(t)z_j(t) \quad (2.12a)$$

$$C(t) = \left[\sum_{j=1}^{N} R_j^{-1}(t)\right]^{-1} \sum_{j=1}^{N} R_j^{-1}(t)C_j(t) \quad (2.12b)$$

$$R(t) = \left[\sum_{j=1}^{N} R_j^{-1}(t)\right]^{-1} \quad (2.12c)$$

### 2.4 Road Representations in Map Services

This section describes how a map can be divided into smaller parts. After that, the segmentation of roads is also explained.

#### 2.4.1 Segmentation of Maps

Maps can be divided into multiple smaller parts called *tiles*. There are multiple approaches to break a map into smaller parts and the algorithm created by HERE is used in this work [9].

The tiles are of different sizes depending on the *zoom level*. A higher level is received by dividing each of the current tiles into four new, equally sized tiles. These newly created tiles form the new zoom level. This procedure can be executed as long as wished, leading to smaller and smaller tiles. Each tile has a specific ID, which makes it possible to get a specific tile on the desired zoom level. An example of tiles on different levels can be seen in Figure 2.1 [9].
2.4 Road Representations in Map Services

2.4.2 Segmentation of Roads

Roads on maps can be modeled according to so-called links. A link is defined as a part of a road with one node in the beginning and one at the end. There are multiple factors that lead to an endpoint of a link. Change in a road attribute, for example when the allowed velocity is reduced or increased, is one reason. Another factor is if the link reaches an intersection. Each link has, just as the tiles, a unique ID [8]. It is possible to divide these links into smaller parts. These will here be called sublinks. Figure 2.2a illustrates the road model.

Observe that there can be multiple links, and thereby multiple sublinks, within a tile. Figure 2.2b illustrates this.

Figure 2.1: Example of different zoom levels. A map is divided into more and smaller parts the higher zoom level. These limited areas of a map are called tiles.

(a) Illustration of the road model. A road can be divided into links, which in turn can be divided into smaller parts called sublinks.
(b) Illustration of that there can be multiple links within one tile. A tile is a limited area of a map while a link is a part of a road.

Figure 2.2: Illustration of how roads can be divided into links which in turn can be divided into smaller parts called sublinks. Furthermore, it is illustrated that multiple links can be located within a tile, which is a limited area of a map.
2.5 Evaluation Methods

In [28] different metrics for evaluating models are investigated. Here two different ones are presented. Furthermore, a common way of evaluating weather forecasts is presented in Section 2.5.3.

2.5.1 Average Absolute Error

The error metric called average absolute error, AAE, measures how well a model’s predicted values represent the ground truth data. More precisely, AAE is the mean difference between the estimated values and the ground truth values. It is calculated according to

$$\text{AAE} = \frac{1}{N} \sum_{i=1}^{N} |\tilde{x}_i - x_i|$$

(2.13)

where $\tilde{x}$ is the predicted value, $x$ is the ground truth and $N$ denotes the total number of error pairs. The lower value of AAE, the better the model [28].

2.5.2 Root Mean Square Error

The root mean square error, RMSE, is calculated according to [27]

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\tilde{x}_i - x_i)^2}.$$  

(2.14)

The predicted value is denoted $\tilde{x}$ while the ground truth is denoted $x$. Variable $N$ is the total number of error pairs. As for AAE presented in Section 2.5.1, the lower RMSE value the better.

2.5.3 Evaluating Weather Forecasts

Weather has a tendency to vary slowly from day to day. Hence, when forecasting of weather should be performed it is reasonable to predict that the weather will be similar to the previous state. This type of forecasting, when the next state is assumed to be the same as the previous, is the so-called persistence model. Due to that this model is the easiest one, any other model for weather prediction should be compared to this one. This comparison makes it possible to argue whether another model is a potential substitute for the persistence model or not [16].
This chapter explains the methods used in this work. First, the data and division of datasets are outlined. It is followed by a description of the temperature models, how multiple and missing measurements are handled and the spatial dependency models. Last, the temperature estimation process is presented together with the evaluation method.

### 3.1 Data Description

Measurements are continuously reported from cars and stored in a database from where different datasets can be extracted. Every observation contains several data fields. The information from the observations used in this work is summarized in Table 3.1.

**Table 3.1:** Data fields of an observation registered by a car.

<table>
<thead>
<tr>
<th>Data field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Timestamp of when the observation was registered. Format: YYYYY-MM-DD HH:MM:SS</td>
</tr>
<tr>
<td>Tile ID</td>
<td>The tile the observation belongs to.</td>
</tr>
<tr>
<td>Link ID</td>
<td>The link the observation belongs to.</td>
</tr>
<tr>
<td>Sublink ID</td>
<td>The sublink the observation belongs to.</td>
</tr>
<tr>
<td>Report ID</td>
<td>The car that reported the observation. Observe that a car changes report ID after three minutes.</td>
</tr>
<tr>
<td>Latitude</td>
<td>Latitude coordinate of the observation.</td>
</tr>
<tr>
<td>Longitude</td>
<td>Longitude coordinate of the observation.</td>
</tr>
<tr>
<td>Temperature</td>
<td>Temperature value of the observation. Quantized to half degrees.</td>
</tr>
</tbody>
</table>
3.2 Time Discretization

Measurements from the cars are received irregularly in time. Each observation is connected to a sublink and due to the unevenness in measurements, some sublinks have lots of measurements for a specific time interval and some none. The states of the sublinks are desired to be estimated regularly. Only running the Kalman filtering when there are available observations can lead to several hours between the estimations. Due to this, a fix time discretization of one hour has been decided, leading to that all measurements with the same hour in their timestamp will be fused together and used for the same time index $k$. Hence, discrete time index $k$ represents hours. The method of fusing multiple observations is further described in Section 3.5.

3.2.1 Timestamp Vector Generation

The temperature estimation is performed on a predetermined dataset leading to a predetermined number of discretized time steps $k$. The estimation process is performed hour-wise on the dataset, leading to a timestamp vector, $s$, according to

$$s = \begin{bmatrix} 2019-02-04 & 22 \\ 2019-02-04 & 23 \\ 2019-02-05 & 00 \end{bmatrix}.$$ (3.1)

Note that $s$ contains all hours between the first and the last timestamp in the dataset.

3.3 Dataset Generation

The aim with this project is to estimate the temperature of a sublink. This can be achieved by letting the sublink depend on nearby sublinks’ observations. Datasets containing different amounts of sublinks have been used to evaluate the different spatial dependency models described in Section 3.7. The datasets are presented in this section, where the coarsest dataset with the most sublinks is described first. It is followed by datasets of finer and finer scale.

Observe that only the sublinks which has at least one observation registered during the time interval, for which data is extracted, are known as sublinks within the tile or link. Hence, not all sublinks within an area are certain to be represented in the generated dataset. Different time intervals will probably lead to that different sublinks are included in the dataset.

Section 3.2 mentioned that the time is discretized in hours. Therefore, the information of minutes and seconds, which can be seen in the timestamp in Table 3.1, is redundant and hence neglected during creation of the datasets.
3.3 Dataset Generation

3.3.1 Tile Level

This model uses a dataset with all observations from a tile of the approximate size of four square kilometers. All measurements belonging to the sublinks within this tile are used as observations during the Kalman filtering. This means that all sublinks’ observations for the discrete time index \( k \) are gathered in an observation vector, \( z \), created according to

\[
\mathbf{z}_k = \begin{bmatrix}
\text{obs}_{k,n} \\
\text{obs}_{k,n+i} \\
\vdots \\
\text{obs}_{k,N}
\end{bmatrix}
\]  

where each row represents different sublinks. Note that the \( z \) vector can be of different lengths due to that there are not observations for all sublinks at all \( k \). However, it has a max length of \( N \) rows, where \( N \) is the number of sublinks in the tile. This is the case because multiple measurements for the same sublink and time instance \( k \) are fused according to Section 3.5, leading to only one observation per sublink and hour. Observation for sublink \( n \) at time index \( k \) is denoted \( \text{obs}_{k,n} \). Variable \( i \) represents an integer showing that not all sublinks are present in the observation vector.

A vector, \( \mathbf{R}_k \), containing measurement variances belonging to \( \mathbf{z}_k \) is also created according to

\[
\mathbf{R}_k = \begin{bmatrix}
\text{r}_{k,n} \\
\text{r}_{k,n+i} \\
\vdots \\
\text{r}_{k,N}
\end{bmatrix}
\]  

The initial measurement noise variance, \( \sigma^2_{\mathbf{R}} \), is modified for respective sublink, with respect to the number of observations at hour \( k \), according to Section 3.5. Hence, the element at row \( n \) in \( \mathbf{R}_k \), here denoted \( \text{r}_{k,n} \), is the corresponding variance of the observation for sublink \( n \).

3.3.2 Link Level

This dataset contains all observations from a link. Hence, a fewer amount of sublinks are taken into account when building the observation vector \( \mathbf{z}_k \), illustrated in (3.2), than when sublinks in a whole tile are used. This leads, in most cases, to fewer observations per time instance, but these observations are closer, with respect to distance, to the sublink which temperature is to be estimated. A vector, \( \mathbf{R}_k \), with variances belonging to the observations is also formed in the same way as described in Section 3.3.1.
3.3.3 Sublink Level

This dataset only contains observations from the sublink which temperature is to be estimated. Hence, no sublinks nearby are contributing with observations. This leads to that the observation vector $z_k$, see (3.2), in this case only consists of one value $z$. Similarly, the measurement variance vector $R_k$, see (3.3), does only contain one variance value.

3.4 Temperature Models

Change of temperature over time is modeled through a state space model (see Section 2.2). Three different temperature models will be compared. The first one is based on the constant position model while the other two are based on the constant velocity model.

3.4.1 Constant Position Model

The persistence model introduced in Section 2.5.3 is realized using the CP model mentioned in Section 2.1.2. Neglection of temperature change leads to a state vector with only the state temperature, $t_k$. Hence, it becomes a scalar and is therefore denoted $x_k$ instead of $x_k$. Consequently, all matrices become scalars. The components of this temperature model’s state space model can be seen in (3.4). Note that the process noise covariance $Q_k$ is here initialized with $\sigma_{Q,p}$ while the state space model in Section 3.4.2 uses $\sigma_{Q,v}$. Index Q represents the process noise covariance matrix $Q$ and $p$ shows that it belongs to the CP model.

\[
\begin{align*}
    x_k &= t_k \\
    F_k &= 1 \\
    Q_k &= \sigma^2_{Q,p} \\
    P_0 &= \sigma^2_p \\
    H_k &= 1 \\
    R_k &= \sigma^2_R
\end{align*}
\]

3.4.2 Constant Velocity Model

This temperature model considers temperature change when updating the temperature estimate. Hence, the states represented in $x$ are temperature of a pre-determined sublink and temperature change. The temperature of the sublink, for time step $k$, is here denoted $t_k$ while the temperature change is denoted $\delta t$. Initial value of the sublink’s temperature is the mean temperature value over all measurements belonging to the first timestamp in the dataset that has been chosen. An illustration of the state vector can be seen in (3.5a).
Temperature change is initially set to zero and then updated due to noise. The covariance matrix $P_0$ for the initial state is set to a diagonal matrix containing standard deviation $\sigma_0$. The sublink’s temperature value is changed over time by adding the temperature change value multiplied with the sample time $T_{s,k}$. Neglection of the term $G_k u_k$, see (2.1a), is possible due to the assumption that there are no control actions. Process noise covariance matrix $Q$ is formed according to (2.4) with the sample time $T_{s,k}$ and the standard deviation $\sigma_{Q,v}$. Index $Q$ represents the process noise covariance matrix $Q$ and $v$ shows that it belongs to the CV model. State vector $x$, initial state covariance matrix $P_0$, the $F$ matrix and the process noise covariance matrix $Q$ are illustrated in (3.5).

\[
x_k = \begin{bmatrix} t_k \\ \frac{d t}{dt} \end{bmatrix} \quad (3.5a)
\]

\[
F_k = \begin{bmatrix} 1 & T_{s,k} \\ 0 & 1 \end{bmatrix} \quad (3.5b)
\]

\[
Q_k = \begin{bmatrix} \frac{T_{s,k}^3}{3} & \frac{T_{s,k}^2}{2} & \frac{T_{s,k}}{2} & \frac{T_{s,k}}{2} \\ \frac{T_{s,k}^2}{2} & \frac{T_{s,k}}{2} & \frac{T_{s,k}}{2} & \frac{T_{s,k}}{2} \\ \frac{T_{s,k}}{2} & \frac{T_{s,k}}{2} & \frac{T_{s,k}}{2} & \frac{T_{s,k}}{2} \\ \frac{T_{s,k}}{2} & \frac{T_{s,k}}{2} & \frac{T_{s,k}}{2} & \frac{T_{s,k}}{2} \end{bmatrix} \sigma_{Q,v}^2 \quad (3.5c)
\]

\[
P_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \sigma_0^2 \quad (3.5d)
\]

Only one observation at the time will be used as input to the Kalman filtering, leading to that the $H$ matrix is a row matrix. Furthermore, a one dimensional observation vector leads to that the measurement noise covariance matrix $R$ only consists of one element, the measurement variance. It is therefore denoted $R_k$ instead and initiated with the standard deviation $\sigma_R$. Note that the measurement variance is changed if multiple measurements are fused. This process is described in Section 3.5. The $H$ matrix and the measurement variance $R$ are illustrated in (3.6).

\[
H_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad \forall \ k \quad (3.6a)
\]

\[
R_k = \sigma_R^2 \quad (3.6b)
\]

Two different variants of this temperature model are examined. The difference is how the case of missing measurements is handled. More information about the approaches is found in Section 3.6.
3.4.3 Constant Velocity Singer Model

This model is inspired by the Singer model introduced in Section 2.1.3. The difference is that a regularization term \( \alpha \) is applied to the velocity instead of the acceleration. This model uses the same matrices as the CV model described in Section 3.4.2, besides the \( F \) matrix. Variable \( \alpha \) is inserted according to

\[
F_k = \begin{bmatrix} 1 & T_{s,k} \\ 0 & \alpha \end{bmatrix},
\]

leading to a regularization of the temperature change \( \delta t \). The value of \( \alpha \) can be found in Table 3.2. This model is shortened CV-Singer.

3.5 Fusion of Multiple Observations

This section describes the process of fusing multiple observations. There are two different scenarios that have to be handled. The first case is when the same car has reported multiple measurements for the same hour and sublink. The second case is when different cars have reported multiple measurements for the same hour and sublink. Every measurement is linked to a so-called report ID. This ID represents which car that sent the measurement. However, as stated in Table 3.2, a car changes report ID after three minutes. This leads to the risk of observations with different report ID could have been reported by the same car.

3.5.1 Dependent Observations

When the same car has registered multiple measurements for the same hour and sublink these measurements are fused without combining the corresponding measurement noise variances. All measurements belonging to the same hour, sublink and that have the same report ID are fused by calculating the mean value of these observations. Due to that the measurement noise variances are equal, the \( R \) corresponding to the fused measurement is simply equal to the one before the fusion took place.

3.5.2 Independent Observations

When multiple measurements for the same hour and sublink exist, but there are different cars beyond these, they are considered independent. Hence, the second approach for measurement fusion, see (2.12), mentioned in Section 2.3 is used to handle this fusion.

Due to the assumption that all sensors have the same covariance matrix \( R \), Equation (2.12c) is rewritten according to

\[
R(t) = \left[ \sum_{j=1}^{N} R_j^{-1}(t) \right]^{-1} = \left\{ R_1 = R_2 = \ldots = R_N := R \right\} = \left[ NR^{-1} \right]^{-1} = \frac{1}{N} R. \quad (3.8)
\]
This leads to that (2.12a) and (2.12c) can be rewritten as

\[
z(t) = \left[ \sum_{j=1}^{N} R_j^{-1}(t) \right]^{-1} \sum_{j=1}^{N} R_j^{-1}(t) z_j(t) = \left\{ \frac{1}{N} R \sum_{j=1}^{N} R_j^{-1}(t) z_j(t) = \right\} (3.8) = \frac{1}{N} R \sum_{j=1}^{N} R_j^{-1}(t) z_j(t) = \left\{ R_1 = R_2 = \ldots = R_N := R \right\} = \frac{1}{N} RR^{-1} \sum_{j=1}^{N} z_j(t) = \frac{1}{N} \sum_{j=1}^{N} z_j(t)
\]

(3.9)

respective

\[
C(t) = \left[ \sum_{j=1}^{N} R_j^{-1}(t) \right]^{-1} \sum_{j=1}^{N} R_j^{-1}(t) C_j(t) = \left\{ \frac{1}{N} R \sum_{j=1}^{N} R_j^{-1}(t) C_j(t) = \right\} (3.8) = \frac{1}{N} R \sum_{j=1}^{N} R_j^{-1}(t) C_j(t) = \left\{ R_1 = R_2 = \ldots = R_N := R \right\} = \frac{1}{N} RR^{-1} C = C.
\]

(3.10)

The measurements belonging to the same hour and sublink, but have different report ID are fused using (3.8)-(3.10).

### 3.6 Handling Missing Observations

The time discretization mentioned in Section 3.2 leads to that some sublinks does not have any observations for some hours. However, state estimation is still desired to be performed for every time steps, which leads to that some kind of prediction still has to be performed. Two different approaches are tested and these are described in this section.

#### 3.6.1 Complete Time Update

The first approach is to use the Bernoulli parameter \( \gamma \) presented in Section 2.2.1. If there are no observations available for the current discretized time step, then the Kalman filtering is performed with modification. Only the time update step is run and the measurement update phase is neglected via the Bernoulli parameter. This leads to that the predicted temperature value is used as an estimate, instead of the corrected one which otherwise is received after the measurement update.
3.6.2 Incomplete Time Update

The second approach is, just like the method described in Section 3.6.1, also a modification of the Kalman algorithm. Similarly, the measurement step is disregarded. However, only the covariance matrix $P$ is updated during the time update step. This means that the covariance values will grow continuously for the time instances where no observations are available. On the contrary, the state vector will remain unchanged until another observation is accessible.

3.7 Spatial Dependency Models

Section 3.3 mentioned that nearby sublinks can contribute with observations during temperature estimation of a specific sublink. Further modification of sublinks’ impact on each other will be introduced here in the form of spatial dependency models. Three different spatial dependency models are presented.

The variables used during all tests of different combinations of spatial dependency models, datasets and temperature models can be seen in Table 3.2. Initial temperature change, $\delta t_0$, and process noise standard deviation, $\sigma_{Q,v}$, are used for the CV model. Note that this includes both the regular CV model and the CV-Singer model. However, regularization term $\alpha$ is only included for the CV-Singer model. Similarly, process noise standard deviation, $\sigma_{Q,p}$, is only used when the CP model is the temperature model. The rest of the parameters, initial state standard deviation $\sigma_0$ and maximal covariance value $l$ are used for all temperature models.

Table 3.2: Parameters used for all spatial dependency models

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial temperature change, $\delta t_0$</td>
<td>0.0</td>
</tr>
<tr>
<td>Process noise standard deviation (when using CV model), $\sigma_{Q,v}$</td>
<td>0.5</td>
</tr>
<tr>
<td>Process noise standard deviation (when using CP model), $\sigma_{Q,p}$</td>
<td>0.8</td>
</tr>
<tr>
<td>Regularization term (when using CV-Singer model), $\alpha$</td>
<td>0.55</td>
</tr>
<tr>
<td>Measurement noise standard deviation, $\sigma_R$</td>
<td>3.0</td>
</tr>
<tr>
<td>Initial state standard deviation, $\sigma_0$</td>
<td>2.0</td>
</tr>
<tr>
<td>Maximal covariance value, $l$</td>
<td>144.0</td>
</tr>
</tbody>
</table>
3.7 Spatial Dependency Models

3.7.1 No Spatial Dependency

This model does not consider spatial dependencies when creating observation vector $z_k$ and associated measurement variance vector $R_k$. The observation vector and the variance vector are created according to Section 3.3 and then used in the temperature estimation process described in Section 3.8. Hence, measurements at all sublinks in the dataset contribute equally to the temperature estimation.

3.7.2 Distance Impact

The idea with this model is to include some spatial dependency, meaning that measurements at sublinks close in the spatial dimension have a larger contribution to each other's temperature estimation than sublinks far away.

The observation vector $z_k$ does still consist of observations from all sublinks in the dataset (as described in Section 3.3), but their corresponding measurement variance values are adjusted with respect to distance to the sublink which temperature is to be estimated. Values in the measurement noise variance vector, $R_k$, are multiplied with weights decided by the function

$$f(d) = 0.01d + 1.$$  \hspace{1cm} (3.11)

The function in (3.11) makes the weights increase with distance. Weights for some example distances are found in Table 3.3.

Latitude and longitude coordinates for measurements belonging to the same sublink can vary due to that the location of the car decides these values. Hence, the mean values of the latitude and longitude are calculated respectively for all measurements in a dataset that belongs to a sublink. These mean values are then used during the distance calculation.

The distance between two sublinks is calculated using the Haversine formula (see Appendix B). This results in a distance in meters between two latitude and longitude coordinates. The distance between sublink $n$, which temperature is being estimated, and sublink $m$, which contributes with an observation, is denoted $d_{n,m}$. This distance is then inserted into the weight function to receive the correct weight. The row corresponding to sublink $m$ in the measurement noise variance vector $R$, see (3.3), is multiplied with the weight $w_m$. These new variance values are then used, together with their corresponding observations, in the temperature estimation process described in Section 3.8. This modification leads to that sublinks close in terms of distance to the sublink, which temperature is to be estimated, will have larger impact than sublinks farther away.

Table 3.3: Example of weights generated by the weight function. This function is only used for the spatial dependency model that considers distance impact. Weights are calculated with respect to distances between the sublinks and these weights are then used to scale the measurement variances.

<table>
<thead>
<tr>
<th>Distance between two sublinks [m]</th>
<th>0</th>
<th>25</th>
<th>50</th>
<th>75</th>
<th>100</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>1.0</td>
<td>1.25</td>
<td>1.5</td>
<td>1.75</td>
<td>2</td>
<td>6</td>
<td>11</td>
<td>21</td>
</tr>
</tbody>
</table>
3.7.3 Precalculated Correlation Matrix

The idea with this model is that sublinks with similar characteristics should have large impact on each other. This means that the sublinks do not necessarily need to be located close to each other in terms of distance, there could be other circumstances leading to similar behavior. Note that this model does not capture potential biases between sublinks. It only discovers similar behavior, opposite behavior or if there is no correlation in behavior. Due to this more complex dependency between sublinks, these relationships have to be calculated on historical data before the temperature estimation process takes place.

This model builds on a precalculated correlation matrix over the sublinks in the dataset. Elements at position \( [n, m] \) respective \( [m, n] \) are the correlation between sublink \( n \) and sublink \( m \). First step is to choose a dataset from historical data. A dataset that spans over 12 weeks, from November to January, is created by extraction of data for the chosen tile during this time interval. Note that the time interval consists of dates before those of the evaluation sets.

The second step is to sort all observations hour-wise. An observation list,

\[
o_k = \begin{bmatrix}
  [obs_{k,n'}^1, \ldots, obs_{k,n}^M] \\
  [obs_{k,n+i}^1] \\
  \vdots \\
  [obs_{k,N', obs_{k,N}^2}]
\end{bmatrix}^T,
\]  

is created in a similar way as the observation vector described in Section 3.3.1. The difference here is that multiple measurements for the same sublink and hour \( k \) are not fused if the observations have been reported from different cars. These observations are stored in a list which in turn is stored in the observation list. The variable \( M \) is the number of observations for sublink \( n \) at time instance \( k \). Observe that the inner lists can be of different sizes, due to that the sublinks have different amounts of observations. Similarly, the observation lists, \( o_k \), can have different length from time to time due to the fact that some sublinks do not have any observations for some \( k \). Observation lists for all time instances, meaning all hours between the first and last timestamp in the dataset, are extracted.

The sample correlation (see Appendix A) is calculated with respect to temperature, which can vary depending on date and hour of day. Consequently, only observations close in time are paired. Observations from respective sublink that have been registered the same date and hour plus/minus one hour are paired as correlation pairs on the form \((n_i, m_i)\) (see Appendix A). Variable \( n_i \) represents an observation for sublink \( n \) and \( m_i \) represents an observation for sublink \( m \). Index \( i \) shows that the observations belong to the same time interval that spans over three hours. A minimum limit of eight correlation pairs has been decided for calculation of correlation between two sublinks.
Due to the grouping of correlation pairs with respect to time, also means and standard deviations for respective time interval have to be calculated. A vector with mean values for each sublink is generated according to

\[
\mathbf{m} = \begin{bmatrix} m_0 \\ \vdots \\ m_I \end{bmatrix}. \quad (3.13)
\]

The row index of the vector represents the time interval \( i \). Index \( I \) denotes index of the last time interval. Equally, vectors with the standard deviations are created. Once enough correlation pairs have been created, the sample correlation is calculated almost according to (A.1). The modified equation is given by

\[
r_{xy} = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} (x_{ij} - \bar{x}_i)(y_{ij} - \bar{y}_i)}{\frac{1}{I} \sum_{i=1}^{I} \sqrt{\sum_{j=1}^{J} (x_{ij} - \bar{x}_i)^2} \cdot \frac{1}{J} \sum_{j=1}^{J} \sqrt{\sum_{i=1}^{I} (y_{ij} - \bar{y}_i)^2}}. \quad (3.14)
\]

The time intervals of three hours during the pairing process leads to an outer summation which goes through all intervals \( i \). The variable \( I \) denotes the total number of three hour intervals. Each time interval has specific means \( \bar{x}_i \) and \( \bar{y}_i \). The inner summation handles the different correlation pairs for each time interval. Variable \( J \) denotes the total number of correlation pairs for a specific time interval. Note that the number of correlation pairs varies between the time intervals. Furthermore, note that the outer sums in the denominator are divided by \( I \) leading to mean values with respect to the number of time intervals.

Section 3.3.1 describes the creation of observations vector \( \mathbf{z}_k \) and associated measurement variance vector \( \mathbf{R}_k \). This spatial dependency model modifies the measurement variance vector. Each row index in \( \mathbf{z}_k \) represents a specific sublink. The measurement variance vector is built in the same way, making it possible to pair observation and variance belonging to the same sublink. Similarly, the correlation between two sublinks, \( n \) and \( m \), is represented by index \([n, m]\) in the precalculated correlation matrix. The correlation between the sublink which temperature is to be estimated and all the sublinks which are represented in the observation vector are extracted from the precalculated correlation matrix. These correlation values are then mapped to measurement variance values using the following function

\[
f(x) = \min \left( 16, \frac{4}{x} + 1 \right). \quad (3.15)
\]

The function is also illustrated in Figure 3.1. However, the correlation values are between minus one and one and to get the desired mapping all these values have to be adjusted so the new interval is between zero and two instead. Therefore, one is added to all correlation values before the mapping is performed. As can be seen in the figure, the highest possible shifted correlation value two is mapped to the variance value three while correlation values below 0.25 are all set to value sixteen. These calculated variance values are then inserted on respective row, in the measurement variance vector \( \mathbf{R}_k \), that represents the sublink that contributed
3.8 Estimation Process

All spatial dependency models described in Section 3.7 are tested through the temperature estimation process. The process is described briefly in Algorithm 5.

The first step in the estimation process is the initialization. First of all, a dataset needs to be created. All observations in the database that fulfill the criteria of either a specific tile ID, link ID or sublink ID between two different dates are extracted. These observations form the dataset. Next, a timestamp vector $s$ is formed according to the method described in Section 3.2.1 and the state space model is initiated, according to Section 3.4, with the parameters in Table 3.2. The final step of the initialization is to set the first sample time, $T_{s0}$, to one.

with an observation. The observation vector and the new variance vector are then used in the temperature estimation process described in Section 3.8.

Observe that the sublinks represented in the precalculated correlation matrix might not be the same as the ones represented in the evaluation dataset. This is the case due to that different time intervals are used when extracting data for the training dataset, used when creating the precalculated correlation matrix, and for the evaluation dataset. This issue was further explained in Section 3.3. This can lead to the problem that the correlation between two sublinks is requested but has not been calculated. Hence, if that is the case, the maximal variance value in the measurement variance vector $R_k$ is used.

Figure 3.1: The function that maps shifted correlation values to variance values. The shifted correlation values are between zero and two instead of minus one to one. A maximum value of sixteen has been decided for the corresponding variance value while the minimum value is three.
3.8 Estimation Process

Algorithm 5: Estimation process

Output: x, P

1. Initialization:
   (a) Create a dataset using a tile ID/link ID/sublink ID and a start and end date
   (b) Create a timestamp vector, s
   (c) Initiate the state space model
   (d) Let $T_{s,0} = 1$

2. Main loop:
   for $k$ in $s$ do
     Create observation vector $z_k$ and belonging covariance matrix $R_k$, see (3.2) and (3.3)
     if no observations in $z_k$:
       $\gamma = 0$
       Run modified Kalman filtering, see Algorithm 6
     else:
       for $i$, obs in $z_k$ do
         $z = obs$
         $r = R_k[i] + \sigma^2$
         $\gamma = 1$
         if $i$ is 1:
           $T_{s,k} = 0$
           Run modified Kalman filtering, see Algorithm 6
           $T_{s,k} = 0$
           $T_{s,k+1} = T_{s,k} + 1$
     Return: x, P

The main loop contains the actual temperature estimation. Temperature estimation is performed for each discretized time instance $k$. Due to that the base time has been decided to hours, $k$ represents hours and the temperature estimation is hence performed hour-wise. Every iteration, an observation vector $z_k$ together with a measurement noise variance vector $R_k$ are created. Note that the measurement noise variance vector varies depending on chosen spatial dependency model. Details of composition of the $R_k$ vector is found in Section 3.7. Information for all sublinks within the dataset, for time instance $k$, can be found row-wise in $z_k$ and $R_k$.

There are two possible branches of the main loop. The first is when there are no observations for any sublink, leading to an empty $z_k$ vector. In this case, the Kalman filtering is still performed, but with Bernoulli parameter $\gamma = 0$ leading to that only the time update within the Kalman filtering is performed.
The second branch is when there is at least one valid observation in the observation vector $z_k$. In that case, the Kalman filtering should be performed for every observation, leading to an inner loop. One by one, the observations are extracted from the observation vector $z_k$ and stored in the variable $z$. The corresponding variance value is also extracted and stored in variable $r$. Observe that the measurement quantization is taken into account by addition of $\sigma^2_q$ (see Section 2.1.4). Due to that the measurements are quantized to half degrees, the quantization step $\Delta$ is 0.5 which gives a quantization noise variance of $\sigma^2_q = 0.5^2/12$. Due to existing measurements in this case, the Bernoulli parameter is set to $\gamma = 1$. Since multiple observations are to be used as input to the Kalman filtering for the same time instance $k$, the sample time $T_{s,k}$ has to be set to zero for all observations except the first one. This will prevent the time update step.

The resulting a posteriori vector $x_k$ and covariance matrix $P_k$ are stored in the result vectors $x$ and $P$ for each timestamp. More details about the modified Kalman filtering is available in Section 3.9.

### 3.9 Modified Kalman Filtering Algorithm

The Kalman filter algorithm presented in Section 2.2 has been modified to meet the system requirements. Sample time dependency as well as the Bernoulli process approach have been included in the filtering procedure. Furthermore, this modified version is adjusted to only be performed for one time instance at the time. The modified Kalman algorithm is presented in Algorithm 6.

The first step in the modified Kalman filter algorithm is to add the sample time, $T_{s,k}$, to the $F$ matrix. However, due to that the filtering process is performed for the even time step of one hour, the sample time will either be one or zero. A sample time of one makes the state proceed to the next time instance, while a sample time of zero prevents the time update. This is desired when the filtering should be performed for multiple observations for the same time instance $k$. Note that this step is only performed if the CP model (see Section 3.4.1) is not used.

The second step is to add the sample time to the process noise covariance matrix $Q$. This is performed according to (3.5c), leading to that the variances of the temperature and the temperature change are affected differently by the sample time $T_{s,k}$.

The third step, the time update, is equal to the one in Algorithm 1. However, note that the time update is only performed for the first observation, if there are multiple during the same hour, for the CP model. Hence, the time update is neglected if $T_{s,k}$ is zero in this case.
Algorithm 6: Modified Kalman filter algorithm

Output: \( x_k, P_k \)

Input: \( F_k, H_k, Q_k, r_k, x_{k-1}, P_{k-1}, T_{s,k}, \gamma_k \)

1. Add \( T_{s,k} \) to the \( F \) matrix if not \( CP \)

2. Add \( T_{s,k} \) to the \( Q \) matrix

3. Time update:
   (a) \( \bar{x}_k = F_k x_{k-1} \)
   (b) \( \bar{P}_k = F_k P_{k-1} F_k^T + Q_k \)

4. Measurement update:
   \( K_k = \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + r_k)^{-1} \)
   \( x_k = \bar{x}_k + \gamma_k K_k (z_k - H_k \bar{x}_k) \)
   \( P_k = (I - \gamma_k K_k H_k) \bar{P}_k \)

5. Force \( P \) to be positive definite and to have values below or equal to \( l \):
   (a) Perform SVD of \( P_k \): \( P_k = USV^T \)
   (b) Set all singular values in \( S \) that are less or equal to zero to \( \eta \)
   (c) Only allow singular values in \( S \) lower or equal to \( l \) by assigning values over the limit to \( l \)
   (d) Compose \( P_k \) according to (5a) with the modified \( S \)

Return: \( x_k, P_k \)

The fourth step in the modified Kalman filtering process is the measurement update. This step has included the Bernoulli parameter \( \gamma \) to enable the filtering process to run even when there are no observations available. The resulting products after the measurement update is the a posteriori vector \( x_k \) and belonging covariance matrix \( P_k \). These are returned each time the filtering is performed.

The fifth step of the modified Kalman algorithm handles the requirement that the covariance matrix must be symmetric positive definite. This is achieved by performing a singular value decomposition, SVD, of \( P \). The decomposition leads to that \( P \) is divided into three different matrices according to step 5a in Algorithm 6. Matrices \( U \) and \( V \) contain singular vectors of \( P \) while the diagonal matrix \( S \) contains the singular values. All singular values in \( S \) that are less or equal to zero are assigned the value \( \eta = 10^{-6} \). Furthermore, this step also prevents the values of the covariance matrix values to increase undesirably. No singular values larger than the decided limit \( l \) are allowed and therefore these are assigned value \( l \). The determined limit can be seen in Table 3.2.
3.10 Numerical Instability

During the implementation and testing of the spatial model that uses a precalculated correlation matrix, there arose a problem with the original Kalman filter in terms of numerical instability. The numerical instability was detected due to negative variances in the calculated covariance matrix $P$.

As a solution to this problem, the square root implementation of the Kalman filter (see Section 2.2.2) is used instead. However, another problem not related to numerical instability was still present. The square root Kalman filter implementation gives a symmetric positive definite $P$ if $P^{1/2}$ is of full rank. However, $P^{1/2}$ was not always of full rank leading to an invalid covariance matrix. To cope with this problem, the square root algorithm is slightly modified to prevent the system from creating invalid covariance matrices. The modified square root implementation of the Kalman filter is presented in Algorithm 7.

As can be seen in Algorithm 7, the original square root implementation of the Kalman filter has been extended from six to seven steps. All steps reused from the original square root Kalman filter are performed in the same way as described in Section 2.2.2. However, note that the Kalman filter is only performed for one observation at the time, leading to that the measurement covariance matrix $R_k$ becomes a scalar $r_k$ as well as observation vector $z_k$ becomes a scalar $z_k$.

The seventh and final step of the modified square root Kalman filter handles the problem with a non-positive definite covariance matrix. Furthermore, it also restricts the elements in $P_k$. It is the same approach as in the modified Kalman algorithm (see Algorithm 6). This makes it possible to perform Cholesky decomposition on the covariance matrix in the next iteration due to that $P_k$ now is symmetric positive definite.

Note that the modified square root Kalman filter is only used when there is an observation available. In the case when there is no observation only the time update of the square root Kalman filter (Algorithm 2) is performed with the additional step (7) from the modified square root Kalman filter, see Algorithm 7.
Algorithm 7: Modified square root Kalman filter

1. Calculate the square root of $P_{k-1}$, $Q_k$ and $r_k$:
   
   $\begin{align*}
   P_{k-1} &= P_{k-1}^{1/2} P_{k-1}^{1/2} \\
   Q_k &= Q_k^{1/2} Q_k^{T/2} \\
   r_k^{1/2} &= \sqrt{r_k}
   \end{align*}$

2. Create the following matrix:
   
   $U_k = \begin{bmatrix} r_k^{1/2} & H_k F_k P_{k-1}^{1/2} & H_k G_k Q_k^{1/2} \\
   0 & F_k P_{k-1}^{1/2} & G_k Q_k^{1/2} \end{bmatrix}$

3. Perform QR factorization on $U_k$:
   
   $qr(U_k) = R_{qr,k}^T Q_{qr,k}^T = \begin{bmatrix} S_k^{1/2} & 0 & 0 \\
   K_k S_k^{1/2} & P_k^{1/2} & 0 \end{bmatrix} Q_{qr,k}$

4. Extract $S_k^{1/2}$, $K_k S_k^{1/2}$ and $P_k^{1/2}$ from $R_{qr,k}$

5. Update the state estimate:
   
   $x_k = F_k x_{k-1} + K_k S_k^{1/2} S_k^{-1/2}(z_k - H_k F_k x_{k-1})$

6. Create the covariance matrix from $P_k^{1/2}$:
   
   $P_k = P_k^{1/2} P_k^{T/2}$

7. Force $P$ to be positive definite and to have values below or equal to $l$:
   
   (a) $P_k = USV^T$
   (b) Set all singular values in $S$ less or equal to zero to $\eta$
   (c) Only allow singular values in $S$ lower or equal to $l$ by assigning values over the limit to $l$
   (d) Compose $P_k$ according to 7a with the modified $S$

Return: $x_k, P_k$
3.11 Evaluation

This section presents the evaluation process. Temperature estimation is performed on multiple tiles to enable evaluation of more than one sublink. Furthermore, the temperature estimation is evaluated on two different time periods, one that has a temperature change of around five degrees day to night and one that varies around fifteen degrees between day and night. The first time period uses measurements from the end of January to the beginning of February while the second time period uses measurements from April. How tiles are chosen is described in Section 3.11.1. Once a set of tiles have been chosen, ground truth data has to be extracted from the weather stations within these tiles. The extraction of ground truth data is described in Section 3.11.2. Lastly, the error metrics used to compare the different models are presented in Section 3.11.3.

3.11.1 Dataset Analysis

All tiles containing a weather station were examined to enable a decision about which ones that should be included during evaluation. Three different requirements decide whether a tile is suitable. These requirements consider number of measurements, number of unique report ID and variance of temperature.

The first requirement is that there should be a minimum number of measurements registered to the sublink which temperature is to be estimated. A minimum of ten measurements has been decided. Section 3.7.3 describes the spatial model that uses precalculated correlation values to adjust the measurement variances. It was there stated that at least eight correlation pairs should be used to calculate the correlation between two sub-links. Due to this requirement, tiles with multiple different report ID per hour are desired. If a sublink has at least two different report ID during one hour, then at least one correlation pair can be created for that specific hour and sublink. Hence, the number of unique report ID for the sublink, which is desired to estimate the temperature for, is calculated hour-wise. The tiles which sublink has at least five hours with a minimum of two unique report ID are considered suitable for evaluation. This because a minimum of two different report ID per hour and sublink indicates that those tiles are probable to lead to valid correlation calculations. Note that this requirement is tested for a dataset created over the time period for which the precalculated correlation matrix is supposed to be calculated.

The last requirement is also related to the spatial dependency model which uses precalculated correlation values. A tile with large variance when it comes to temperature values, within the same hour, indicates that there are different conditions in the tile that lead to different temperatures. Hence, there are sublinks within the tile that vary a lot in temperature although the temperature has been measured under the same hour and day. Sublinks which behave differently when it comes to the temperature values should not have large impact on each other during the temperature estimation. This was stated in Section 3.7.3. The spatial dependency model which calculates correlation values between all sublinks
within the tile could discover these variations. Tiles with an hour-wise temperature variance larger than 0.8 degrees are therefore used for evaluation.

### 3.11.2 Extraction of Ground Truth Data

Measurements from weather stations, called RWIS, are used as ground truth data. Measurements from available RWIS within the specified area, where the temperature is estimated, are extracted for the chosen timestamps. A RWIS measures the temperature around every tenth minute. However, due to the discretization of time in terms of hours for the estimation process, only one measurement per hour from each RWIS is used as ground truth data. To achieve as regular intervals as possible between the ground truth measurements, measurements around every half hour within each hour are chosen as ground truth for that specific hour.

### 3.11.3 Error Metric

The precision of the temperature estimation for a sublink is measured using the average absolute error, AAE, and the root mean square error, RMSE. These were presented in (2.13) respective (2.14). These metrics are calculated over all hours where there exists both an estimated temperature value and a ground truth value from the RWIS.

### 3.11.4 Consideration of Bias

After all temperature models, datasets and spatial models have been evaluated once using measurements from a specific week, the result is analyzed to find potential bias between the observations from the cars and the measurements from the RWIS. Then the temperature estimation is performed again, but this time with the bias included.

The extended Kalman filter algorithm, see Algorithm 4, is used for the second evaluation round. This is to include the bias, $b$, in the relation between the observation $z$ and the state $x$ according to

$$z_k = h(x_k, \delta_k) = Hx_k + b + \delta_k.$$  \hspace{1cm} (3.16)

Besides from the addition of $b$, the state space model is unchanged from how it was described in Section 2.2.
This chapter presents the results from the temperature estimations. The temperature estimation has been evaluated for both temperature varying five and fifteen degrees per day.

4.1 Time Period with Small Temperature Changes

This section presents the temperature estimation results when all configurations of models have been evaluated for a time period where the temperature changes around five degrees per day.

4.1.1 Temperature Models

Different temperature models lead to different behavior in the temperature estimation. This is illustrated in Figures 4.2-4.3. The blue dots in the figures are observations received from cars while the green crosses are measurements from the RWIS. The confidence bands have been formed by adding respective subtracting the belonging standard deviation to each estimated temperature value. The variance of each temperature estimation is generated during the Kalman filtering and stored in the covariance matrix $P$. Hence, the square root of the variance, which is extracted from $P$ for each hour, is calculated and used to form the confidence interval.

Various behavior leads in turn to different AAE and RMSE. To summarize which temperature model that contributes to the most accurate result, the AAE and the RMSE were calculated using estimates and ground truth from all sixteen tiles, all different datasets and all spatial models. Table 4.1 presents the comparison between the temperature models. Figure 4.1 shows the error, defined
as the difference between estimate $\tilde{x}$ and ground truth $x$, per hour for respective temperature model. Note that this is result for a specific sublink. The gray background in the figure represents hours without any observations. The no spatial dependency model has been used to create this graph. For the same type of graphs but for the other spatial dependency models see Appendix C.

Table 4.1 shows that the CV model with complete time update performs the worst. All other models receive about the same errors. Because of the similar performances, the CP model is chosen to be used when evaluating the different datasets and spatial models in the following sections due to its lower model complexity.

Table 4.1: The AAE and RMSE for the different temperature models when evaluating over all sixteen tiles, all datasets and all spatial dependency models. The datasets have been created using measurements from a time period where the temperature varies around five degrees between day and night.

<table>
<thead>
<tr>
<th>Temperature model</th>
<th>AAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV - complete time update</td>
<td>1.269</td>
<td>2.144</td>
</tr>
<tr>
<td>CV - incomplete time update</td>
<td>1.062</td>
<td>1.673</td>
</tr>
<tr>
<td>CV-Singer</td>
<td>1.059</td>
<td>1.677</td>
</tr>
<tr>
<td>CP</td>
<td>1.048</td>
<td>1.659</td>
</tr>
</tbody>
</table>

Figure 4.1: The error, defined as the difference between estimate $\tilde{x}$ and ground truth $x$, per temperature model and hour for a specific sublink. Gray background represents hours without any observations from cars. All observations in the dataset sublink have been used to generate this graph. The dataset has been created using measurements from a time period where the temperature varies around five degrees between day and night. Furthermore, the no dependency spatial model has been used.
4.1 Time Period with Small Temperature Changes

Figure 4.2: Illustration of the behavior of the CV model with complete respective incomplete time update when estimating temperature of a sublink. The sublink dataset and the spatial model called no spatial dependency have been used for these examples. The dataset has been created using measurements from a time period where the temperature varies around five degrees between day and night. The observations are measurements from cars while the ground truth is measurements from RWIS.
Figure 4.3: Illustration of the behavior of the CV-Singer model and the CP model when estimating temperature of a sublink. The sublink dataset and the spatial model called no spatial dependency have been used for these examples. The dataset has been created using measurements from a time period where the temperature varies around five degrees between day and night. The observations are measurements from cars while the ground truth is measurements from RWIS.
4.1.2 Datasets and Spatial Models

The AAE and RMSE were calculated using estimates and ground truth from all sixteen tiles for all configurations of datasets and spatial dependency models when using the CP temperature model. The result is presented in Table 4.2. As can be seen in Table 4.2, the dataset that uses all measurements within the tile generates the lowest AAE and RMSE. Due to this, the following results are shown with respect to the tile dataset.

Figure 4.4 shows the probability distributions of the difference between estimated temperature $\tilde{x}$ and corresponding ground truth value $x$ for respective spatial dependency model. The CP temperature model and the dataset that includes all measurements within a tile have been used when generating these histograms. Furthermore, these histograms have been created using estimates $\tilde{x}$ and ground truth $x$ for all sixteen tiles that the temperature estimation has been evaluated on. All probability distributions have the peak around the same value, 0.65, which is marked with a green line. This value was chosen visually. Additionally, Gaussian distributions have been adjusted to fit the histograms. It is illustrated by the orange dashed line in each figure.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Spatial dependency model</th>
<th>AAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tile</td>
<td>No spatial dependency</td>
<td>0.843</td>
<td>1.242</td>
</tr>
<tr>
<td></td>
<td>Distance impact</td>
<td>0.859</td>
<td>1.301</td>
</tr>
<tr>
<td></td>
<td>Precalculated correlation matrix</td>
<td>0.848</td>
<td>1.259</td>
</tr>
<tr>
<td>Link</td>
<td>No spatial dependency</td>
<td>1.052</td>
<td>1.657</td>
</tr>
<tr>
<td></td>
<td>Distance impact</td>
<td>1.076</td>
<td>1.699</td>
</tr>
<tr>
<td></td>
<td>Precalculated correlation matrix</td>
<td>1.046</td>
<td>1.637</td>
</tr>
<tr>
<td>Sublink</td>
<td>No spatial dependency</td>
<td>1.261</td>
<td>2.007</td>
</tr>
<tr>
<td></td>
<td>Distance impact</td>
<td>1.261</td>
<td>2.007</td>
</tr>
<tr>
<td></td>
<td>Precalculated correlation matrix</td>
<td>1.253</td>
<td>1.997</td>
</tr>
</tbody>
</table>
(a) The probability distribution of the difference between estimated temperature $\hat{x}$ and ground truth value $x$ when evaluating the no spatial dependency model on all tiles.

(b) The probability distribution of the difference between estimated temperature $\hat{x}$ and ground truth value $x$ when evaluating the spatial dependency model called distance impact on all tiles.

(c) The probability distribution of the difference between estimated temperature $\hat{x}$ and ground truth value $x$ when evaluating the spatial dependency model that uses a precalculated correlation matrix on all tiles.

Figure 4.4: The probability distributions of the difference between estimated temperature $\hat{x}$ and ground truth value $x$ when evaluating respective spatial dependency model on all tiles. The datasets have been created using measurements from a time period where the temperature varies around five degrees between day and night. Furthermore, the CP temperature model has been used. The orange dashed lines are Gaussian distributions adjusted to fit respective histogram. The green line is at error 0.65 in all three plots to illustrate that their peaks are around the same value.
4.1 Time Period with Small Temperature Changes

4.1.3 Bias Consideration

This section presents the results for the temperature estimation when the bias has been included in the state-space model as described in Section 3.11.4. The bias was determined as the value of the green line in Figure 4.4, hence 0.65 degrees.

Table 4.3 shows the AAE and RMSE for all datasets and spatial models when using the CP temperature model with and without bias. Furthermore, Figure 4.5 shows the probability distributions of the difference between estimated temperature $\tilde{x}$ and corresponding ground truth value $x$ for respective spatial dependency model when the bias has been included in the estimation process.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Spatial dependency model</th>
<th>No bias included in estimation process</th>
<th>Bias included in estimation process</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AAE</td>
<td>RMSE</td>
<td>AAE</td>
</tr>
<tr>
<td>Tile</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>No spatial dependency</td>
<td>0.843</td>
<td>1.242</td>
</tr>
<tr>
<td></td>
<td>Distance impact</td>
<td>0.859</td>
<td>1.301</td>
</tr>
<tr>
<td></td>
<td>Precalculated correlation matrix</td>
<td>0.848</td>
<td>1.259</td>
</tr>
<tr>
<td>Link</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>No spatial dependency</td>
<td>1.052</td>
<td>1.657</td>
</tr>
<tr>
<td></td>
<td>Distance impact</td>
<td>1.076</td>
<td>1.699</td>
</tr>
<tr>
<td></td>
<td>Precalculated correlation matrix</td>
<td>1.046</td>
<td>1.637</td>
</tr>
<tr>
<td>Sublink</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>No spatial dependency</td>
<td>1.261</td>
<td>2.007</td>
</tr>
<tr>
<td></td>
<td>Distance impact</td>
<td>1.261</td>
<td>2.007</td>
</tr>
<tr>
<td></td>
<td>Precalculated correlation matrix</td>
<td>1.253</td>
<td>1.997</td>
</tr>
</tbody>
</table>

Table 4.3: Comparison of the AAE and RMSE when the temperature estimation is performed with and without bias. The evaluation of all configurations of models has been performed for a time period where the temperature varies around five degrees between day and night.
4.2 Time Period with Large Temperature Changes

This section presents the temperature estimation results when all configurations of models have been evaluated for a time period with large temperature changes. Large temperature change is here defined as a temperature change of around fifteen degrees between day and night. Due to the improved results by the bias, seen in Table 4.3, it is included during all temperature estimations for this time period.

4.2.1 Temperature Models

All temperature models were combined with all configurations of datasets and spatial models for the temperature estimation. The results from all these combinations, for respective temperature model, have been compiled and the resulting AAE and RMSE are presented in Table 4.4. The table shows that the CV model with complete time update has the highest AAE at around 4 degrees. The other temperature models all get a lower AAE of approximately the same value. Similar relation is seen between the RMSE values. However, note that the CP model has a slightly lower RMSE value, which is highlighted using bold font. Due to this, the CP model will be used when presenting the results for the datasets and spatial models in Section 4.2.2.

As mentioned in Section 4.1.1, different temperature models have different behavior. Figures 4.6-4.7 illustrate the temperature models’ behavior when the temperature varies around fifteen degrees between day and night. The no spatial dependency model has been used when generating these graphs. The blue dots in the figures are observations received from cars while the green crosses are temperature measurements from the RWIS. As in Section 4.1.1, the confidence bands are formed using the temperature estimates’ variance values found in the covariance matrix P created during the Kalman filtering.

Table 4.4: The AAE and RMSE for the different temperature models when evaluating over all sixteen tiles, all datasets and all spatial dependency models. The datasets were created using measurements from a time period where the temperature varies around fifteen degrees day to night.

<table>
<thead>
<tr>
<th>Temperature model</th>
<th>AAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV - complete time update</td>
<td>4.113</td>
<td>7.874</td>
</tr>
<tr>
<td>CV - not complete time update</td>
<td>2.697</td>
<td>3.877</td>
</tr>
<tr>
<td>CV-Singer</td>
<td>2.628</td>
<td>3.874</td>
</tr>
<tr>
<td>CP</td>
<td>2.655</td>
<td>3.788</td>
</tr>
</tbody>
</table>
4.2 Time Period with Large Temperature Changes

Figure 4.5: The probability distributions of the difference between estimated temperature $\hat{x}$ and ground truth value $x$ when evaluating respective spatial dependency model on all tiles while including the bias in the temperature estimation. The datasets have been created using measurements from a time period where the temperature varies around five degrees between day and night. Furthermore, the CP temperature model has been used. The orange dashed lines are a Gaussian distributions adjusted to fit respective histogram.
Figure 4.6: Illustration of the behavior of the CV model with complete respective incomplete time update when estimating temperature of a sublink. The sublink dataset and the spatial model called no spatial dependency have been used for these examples. The dataset was created using measurements from a time period where the temperature varies around fifteen degrees day to night. The observations are measurements from cars while the ground truth is measurements from RWIS.
(a) Illustration of the behavior of the CV-Singer model when estimating the temperature of a sublink.

(b) Illustration of the behavior of the CP model when estimating the temperature of a sublink.

Figure 4.7: Illustration of the behavior of the CV-Singer model and the CP model when estimating temperature of a sublink. The sublink dataset and the spatial model called no spatial dependency have been used for these examples. The dataset was created using measurements from a time period where the temperature varies around fifteen degrees day to night. The observations are measurements from cars while the ground truth is measurements from RWIS.
4.2.2 Datasets and Spatial Models

Table 4.5 presents a comparison between all configurations of datasets and spatial models. As in Section 4.1.2, the CP model has been used when generating these results. This is the case due to that the CP model generates the smallest RMSE and has the lowest model complexity. Figure 4.8 shows the probability distributions of the difference between estimated temperature $\tilde{x}$ and corresponding ground truth value $x$ for respective spatial dependency model when the dataset on tile level has been used together with the CP model. The histograms have been generated as described in Section 4.1.2.

The relation between measurements from the cars and measurements from the RWIS for all sixteen tiles is illustrated in Figure 4.9. The red line and the green curve in each plot are adjusted to fit the data points using the least square method. Note that different datasets lead to different relations between the measurements from cars and measurements from weather stations. For the same type of graphs, but for the time period with small temperature changes see Appendix D.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Spatial dependency model</th>
<th>AAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tile</td>
<td>No spatial dependency</td>
<td>2.277</td>
<td>3.082</td>
</tr>
<tr>
<td></td>
<td>Distance impact</td>
<td>2.063</td>
<td>2.916</td>
</tr>
<tr>
<td></td>
<td>Precalculated correlation matrix</td>
<td>2.334</td>
<td>3.133</td>
</tr>
<tr>
<td>Link</td>
<td>No spatial dependency</td>
<td>2.638</td>
<td>3.847</td>
</tr>
<tr>
<td></td>
<td>Distance impact</td>
<td>2.640</td>
<td>3.831</td>
</tr>
<tr>
<td></td>
<td>Precalculated correlation matrix</td>
<td>2.671</td>
<td>3.875</td>
</tr>
<tr>
<td>Sublink</td>
<td>No spatial dependency</td>
<td>3.124</td>
<td>4.377</td>
</tr>
<tr>
<td></td>
<td>Distance impact</td>
<td>3.124</td>
<td>4.377</td>
</tr>
<tr>
<td></td>
<td>Precalculated correlation matrix</td>
<td>3.092</td>
<td>4.392</td>
</tr>
</tbody>
</table>
4.2 Time Period with Large Temperature Changes

(a) The probability distribution of the difference between estimated temperature $\tilde{x}$ and ground truth value $x$ when evaluating the no spatial dependency model on all tiles.

(b) The probability distribution of the difference between estimated temperature $\tilde{x}$ and ground truth value $x$ when evaluating the spatial dependency model called distance impact on all tiles.

(c) The probability distribution of the difference between estimated temperature $\tilde{x}$ and ground truth value $x$ when evaluating the spatial dependency model that uses a precalculated correlation matrix on all tiles.

Figure 4.8: The probability distributions of the difference between estimated temperature $\tilde{x}$ and ground truth value $x$ when evaluating respective spatial dependency model on all tiles while including the bias in the temperature estimation. The datasets have been created using measurements from a time period where the temperature varies around fifteen degrees day to night. Furthermore, the CP temperature model has been used. The orange dashed lines are Gaussian distributions adjusted to fit respective histogram.
Figure 4.9: The measurements from the cars, within the same tile, link respective sublink as the RWIS, plotted with respect to the corresponding measurements from the RWIS. The datasets have been created using measurements from a time period where the temperature varies around fifteen degrees day to night. The least square method has been used to fit the red lines and the green curves to the data points.
5 Discussion

This section contains discussions about the result that was presented in Chapter 4 and some discussion about the number of parameters and their values for the different methods.

5.1 Results

This section begins to discuss the different temperature models’ performances. It is followed by a discussion about the dataset impact and lastly an analysis of the spatial dependency models.

5.1.1 Temperature Models

When the temperature only varies a few degrees from day to night it is reasonable that a model that predicts the same state as previous time instance performs well. This coincides with the result in Table 4.1. Consequently, it is also reasonable that the CV model with complete time update performs the worst. This is due to the fact that it allows constantly increasing or decreasing temperature when there are no observations available although this time period has not significant temperature changes hour to hour. The errors per temperature model and hour for this time period are illustrated in Figure 4.1. In this figure it is visible that the CV model with complete time update has an error that wanders significantly more than the other models when there are missing observations.
When the temperature varies more aggressively, around fifteen degrees from day to night, the CP model, the CV model without complete time update and the CV-Singer model still perform the best. It is reasonable that the CP model and the CV model without complete time update generate similar results due to their similar behavior. The most significant difference between these models is the behavior of the confidence bands, which can be seen in Figures 4.2b and 4.3b. The CV-Singer model allows a restricted variation of the temperature when there are missing observations. A theory could be that this model should perform better than models that only predict the same state as previous time instance due to the larger temperature changes over time in this case. When only comparing the AAE values, the CP model and the CV-Singer model do perform similarly. However, the difference in RMSE is larger, see Table 4.4. The RMSE punishes large errors more than AAE does. Hence, this indicates that the CV-Singer model has more large errors than the CP model.

In summary, the CP model, the CV model without complete time update and the CV-Singer model performed the best no matter which time period they were evaluated for. However, the CP model generated a lower RMSE value for the time period with temperature varying around fifteen degrees from day to night, see Table 4.4. This indicates that the CP model has less large errors than the rest of the models. In addition, one advantage of the CP model is that it has only scalars in its state-space model while the CV models use matrices. This leads to an advantage in computational complexity for the CP model. Hence, the CP model could be to prefer of the evaluated temperature models.

5.1.2 Datasets

No matter for which time period the temperature estimation was evaluated, the dataset with all measurements within a tile generated the lowest AAE and RMSE. It seems to be preferable to use widespread information rather than local. The reason might be a combination of that temperature does not vary significantly many degrees over the tile area of two by two kilometers and the limited amount of car measurements for some links and sublinks. If there are several hours missing measurements for the dataset on link or sublink level, then it is a higher risk that the prediction deviates compared to the true value than if some observations are available. This is visible in Figure 4.1.
5.1.3 Spatial Dependency Models

When the temperature varies around five degrees between day and night all spatial dependency models perform equally for respective dataset, see Table 4.2. On sublink level the distance impact model and the no spatial dependency model have exactly the same error which is reasonable due to their equal behavior for this dataset. The reason why all spatial dependency models generate about the same error, no matter which dataset that is used, might be due to the possibility of similar temperatures around the tile because of the small temperature variations. If the temperature is about the same value within the tile, then it does not matter how these values are processed during the estimation process. However, as already stated in Section 5.1.2, it seems preferable to use as many observations as possible due to the lowest model errors for the tile dataset. This is reasonable when the observations have about the same value.

When the temperature varies more, around fifteen degrees from day to night, there is a larger difference between the spatial dependency models for the tile level. This can be seen in Table 4.5. This time the distance impact model performs the best and its errors are highlighted using bold numbers in the table. The no spatial dependency model and the precalculated correlation matrix model perform similarly.

During a period where the temperature varies a lot there is a possibility that some areas within the tile increase/decrease in temperature quicker than others due to for example shading. This could be an explanation to why the distance impact model generates the lowest AAE and RMSE in this case. The nearby sublinks should have a similar environment and therefore have similar temperatures.

The reason why the no spatial dependency model and the precalculated correlation matrix model performs equally for the time period with large variations in temperature could be due to need of improvement during the creation of the correlation matrix. As mentioned in Section 3.7.3, the precalculated correlation matrix is created using measurements from November to January. The temperature does not vary as much at that time as it does around spring time in April, which was used during evaluation in this case. Hence, large variations in temperature within the tile might not have been captured in the preprocessing phase leading to a similar behavior as the no spatial dependency model which uses a mean value as estimate. An alternative, that might improve the precalculated correlation matrix method, could have been to choose another time period for the preprocessing phase, where larger temperature variations are included.

In conclusion, all spatial dependency models generated about the same errors for the time period with small temperature variations. However, in the case of larger variations the distance impact model performed better than the other two. Hence, this model could be to prefer before the other two. One aspect to consider though, is that this model requires a preprocessing phase where the distances between all sublinks are calculated and then converted to weights. The no spatial dependency model does not require any preprocessing and has no hyperparameters which is an advantage. The improvements in performance can therefore be compared to the model complexity during the process of choosing model.
5.1.4 Consideration of Bias

Section 4.1.3 presents the results when the bias between measurements from cars and measurements from RWIS was considered during the temperature estimation process. Note that this was for the time period when the temperature had small variations, meaning the temperature varies about five degrees between day and night. It can be seen in Table 4.3 that the bias contributes with an improvement for all spatial dependency models on all dataset levels. This is also illustrated in Figure 4.5 where the peaks of the probability distributions of the difference between estimated temperature $\tilde{x}$ and corresponding ground truth value $x$ are more centered around zero than the probability distributions in Section 4.1.2 where no bias was considered. Based on these results it seems that an included bias leads to improved results.

Section 4.2 presents the result, for a time period with large temperature change between day and night, when the bias has been included in the estimation process. The probability distributions of the difference between estimated temperature $\tilde{x}$ and corresponding ground truth value $x$ for respective spatial dependency model can be seen in Figure 4.8. These histograms are not centered around zero although the bias is included. This could be an indication that the bias found in Section 4.1.2 does not exist in the general case. Looking at Figures 4.9a-4.9c there are comparisons between measurements from cars and measurements from RWIS. The fitted lines' imply different biases depending on which dataset that is used. It seems to be a larger bias for this time period than for the previous one. In addition, it seems like the bias is distance dependent. Car measurements further from the RWIS seem to be more overestimated compared to car measurements closest to the RWIS.

There are multiple possible reasons for overestimation of the temperature further away from the RWIS. One explanation could be that the RWIS are placed at high-velocity roads where many cars drive frequently. High speed leads to an increased air flow through the hood compared to when the car is driven more slowly. This leads to a decreased risk for the hood becoming warm and impacting the temperature measurements than if the car is standing still or moving slower. Hence, this could be an explanation to why the temperature is overestimated on tile level compared to sublink level if some measurements on tile level have been registered on roads with lower speed limits. However, this discussion is not as reasonable when taking the link level in consideration. All measurements from the same link as the RWIS are received from the same road with the same speed limit. Hence, the temperature of the cars’ hoods should be about the same temperature for the whole link. Therefore, this theory might not be so probable.

Another hypothesis of the overestimated temperatures is that the RWIS are placed on meteorologically interesting locations. This was mentioned in [2]. With meteorologically interesting means places where the temperature drops the quickest within the tile. It could be reasonable to place the RWIS in that way to detect risk of low friction as fast as possible and thereby have the possibility to warn drivers before the slipperiness has spread around the tile. This theory supports the fact that cars report higher temperatures on sublinks within the same
link due to that the environment around the road can change which might lead to different temperatures. Shading by buildings or trees could for instance be one factor.

In conclusion, there seems like there is a bias between the measurements from the cars and the RWIS that depends on the time of the year. Furthermore, it seems to be distance dependent.

5.2 Parameter Selection

There are several parameters to be chosen for the models and this is a factor that impacts the results. The chosen parameters for the temperature models, presented in Table 3.2, and the parameters for respective spatial dependency model, presented in Section 3.7, have all been set manually.

When it comes to the temperature models there are some parameters that are shared and some model specific ones. To make an as fair model comparison as possible the parameters shared between all models are set to the same values. The \( \text{CV} \) models and the \( \text{CP} \) model have different process noise standard deviation and the \( \text{CV} \)-Singer model has the additional regularization parameter \( \alpha \). How is it certain that the present parameter values work in the general case?

When it comes to the spatial dependency models there are some model specific parameters that can be tuned. The distance impact model scales the measurement noise variances with weights decided by a function of distance between sublinks. The function is presented in (3.11) and it can be modified to give measurements far away more or less impact during the temperature estimation. The spatial dependency model that uses a precalculated correlation matrix has multiple hyperparameters. The first one is how many weeks that should be used when forming the correlation pairs. Secondly, the number of required pairs to calculate the correlation between two sublinks is tune-able. Furthermore, the function that maps correlation to variance is tune-able. All these hyperparameters could be changed to try to improve the estimation result.

In summation, there are multiple hyperparameters that can be tuned to try to improve the result. One possible way to optimize the parameters could be to use an outer loop during the estimation process where different configurations of parameters are tested to investigate which one that generates the most accurate result. However, due to the amount of tune-able parameters this could be time-consuming. How many values of each parameter is it worth evaluating? This is a trade-off between time needed to investigate this further and the possible improvements it will result in.
This chapter answers the problem formulations presented in Section 1.3. Some proposals for future work are also presented.

6.1 Answers to Problem Formulations

This work aimed to investigate different methods of data fusion to estimate the ambient temperature at road segments. Four ways of modeling temperature change over time were evaluated together with the datasets on tile, link and sub-link level. Furthermore, three spatial dependency models, which each handles the measurements in the datasets in different ways, were evaluated. All configurations of temperature models, datasets and spatial dependency models were evaluated for two time periods with varying temperature change. Using the results in Chapter 4 and the discussion in Chapter 5 this section answers the problem formulations presented in Section 1.3.

1. What model can be used for temperature estimation?

All temperature models presented in this work can be used for temperature estimation. However, the constant velocity model with complete time update performed worse and should not be prioritized before the other ones. The other ones performed similarly, but the CP model has the advantage of lower computational complexity. Therefore, this is the one to prefer.
2. Does a local dataset of measurements lead to a more accurate temperature estimation than a dataset containing measurements from a wider area?

No, according to the result in this work it seems preferable to use widespread information rather than local. Tables 4.2 and 4.5 show that, no matter what spatial dependency model or time period, the temperature estimation performed for the dataset on tile level always generates the smallest AAE and RMSE. The reason might be that hours with missing observations are more occurring for the link and sublink dataset. It is in the cases of missing measurements that it is most probable that the temperature estimation deviates from the ground truth as can be seen in Figure 4.1. Due to the fact that measurements within the tile do not vary many degrees compared to the ones at the sublink, which temperature is to be estimated, it is better to use a distant measurement than none at all.

3. Which spatial dependency model leads to the most accurate temperature estimation?

Which spatial dependency model that generates the lowest errors depends on what time period they are evaluated for. It has been stated that it is the best to use measurements from the tile dataset no matter which spatial dependency model that is used and the following statements are therefore built on that the tile dataset is used. For a time period with small changes in temperature the three spatial dependency models perform about the same. However, when the temperature has larger variations the distance impact model generates lower errors than the other models. In conclusion, the distance impact model is to prefer if the same model should be used for all time periods.

6.2 Future Work

The background to this work is how to estimate road conditions to improve the safety in traffic. The focus in this work has been ambient temperature due to the importance of this signal when trying to determine the road condition. However, the models presented are general and could be used for other signals. Hence, a natural progression of this work would be to investigate the models further but this time for other data than temperature, for example friction data.

Possible improvements on the current work could be to perform more intense hyperparameter optimization, which was mentioned in Section 5.2. The current parameter values are hand-picked without further investigation of whether there might be more suited ones.
Another model specific modification could be to create another precalculated correlation matrix, but this time for a time period where the temperature varies such as the one illustrated in Figures 4.6-4.7. This might lead to an improvement in the performance for the precalculated correlation matrix spatial dependency model.

The result in this work states that there is a bias between the vehicle measurements and the measurements from the weather stations. This bias seems to vary depending on time of the year and the distance between the car measurement and the location of the weather station. The knowledge about the bias could be used in future work to improve the temperature models. For example by modifying the relation between the observation $z$ and the state $x$.

When it comes to the extraction of ground truth data from the RWIS it can be performed in several ways. To consider is whether another approach could represent the temperature at a specific sublink better than the current one. A mean value of all reported temperature values, for respective hour, could for instance has been used as ground truth instead of only picking one value to represent the whole hour.
Appendix
The sample correlation between two sets of samples, denoted \(X\) and \(Y\), is calculated according to [20]

\[
r_{xy} = \frac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{N} (y_i - \bar{y})^2}}. \tag{A.1}
\]

Set \(X\) consists of samples \([x_0, \ldots, x_N]\) and set \(Y\) of samples \([y_0, \ldots, y_N]\). Hence, there are \(N\) samples in each set. The mean value of \(x\) respective \(y\) is denoted \(\bar{x}\) respective \(\bar{y}\). Let the sample from set \(X\) and set \(Y\) belonging to the same index \(i\) form a pair. It is denoted a correlation pair.

A correlation value close to one or minus one means that the sets are correlated but in different ways. On the other hand, a value close to zero represents that the two sets are uncorrelated [20].
The Haversine formula is used to calculate the distance between two latitude and longitude coordinates. When using the Haversine formula the distance, \( d \), is calculated according to

\[
d = 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\phi_2 - \phi_1}{2} \right) + \cos (\phi_1) \cos (\phi_2) \sin^2 \left( \frac{\theta_1 - \theta_2}{2} \right)} \right). \quad (B.1)
\]

Variable \( r \) is the radius of Earth, \( \phi_1 \) and \( \phi_2 \) are latitudes and \( \theta_1 \) and \( \theta_2 \) are longitudes. The index represents which latitude and longitude that form a coordinate [4].
Figures C.1 and C.2 show the error per hour for all temperature models for the distance impact respective the precalculated correlation matrix spatial dependency model. Note that these figures have been created using results for a specific sublink. The gray background in the figures represents hours without any observations. The sublink dataset has been used to generate this graph. For the same type of graph but for the no spatial dependency model see Section 4.1.2.
Figure C.1: The error, defined as the difference between estimate $\hat{x}$ and ground truth $x$, per temperature model and hour for a specific sublink. Gray background represents hours without any observations from cars. All observations in the dataset sublink have been used to generate this graph. The dataset has been created using measurements from a time period where the temperature varies around five degrees between day and night. Furthermore, the distance impact model has been used.
Figure C.2: The error, defined as the difference between estimate $\hat{x}$ and ground truth $x$, per temperature model and hour for a specific sublink. Gray background represents hours without any observations from cars. All observations in the dataset sublink have been used to generate this graph. The dataset has been created using measurements from a time period where the temperature varies around five degrees between day and night. Furthermore, the spatial dependency model that uses a precalculated correlation matrix model has been used.
The relation between measurements from the cars and measurements from the RWIS for all sixteen tiles is illustrated in Figure D.1. The red line and the green curve in each plot are adjusted to fit the data points using the least square method. Note that different datasets lead to different relations between the measurements from cars and measurements from weather stations. Furthermore, note that these graphs have been created using data from the time period where the temperature varies around five degrees day to night.
Figure D.1: The measurements from the cars, within the same tile, link respective sublink as the RWIS, plotted with respect to the corresponding measurements from the RWIS. The datasets have been created using measurements from a time period where the temperature varies around five degrees day to night. The least square method has been used to fit the red lines and the green curves to the data points.
Bibliography


