Vehicle Lateral and Longitudinal Velocity Estimation Using Machine Learning Algorithms

by

Amir Fathazam

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2021

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Prediction and estimation of states are of great importance for vehicle control and safety. The conventional observers are designed and used vastly in the automotive industry to fulfill this objective. However, the vehicle behavior can be nonlinear or unpredictable, and it is difficult or impossible to use linear or low-degree nonlinear observers to estimate vehicle states. These observers may fail to estimate the states correctly in high slippery roads, combined slip situations, and maneuvers with intense steering inputs.

In this study, two kernel regression-based machine learning methods are used to estimate lateral and longitudinal velocities. In the estimations with kernel regression methods, only a limited number of reference points in the vicinity of estimation space are needed. The kernel-based methods do not need training and can be implemented for real-time applications. The estimation methods are capable of estimating lateral or longitudinal velocities with a frequency more than 50 Hz.

The suggested estimation methods can be applied for different vehicles without the need to be changed or modified. The proposed approach is capable of utilizing data of any vehicle by normalization. The resulting solution can be used for the state estimation of any vehicle.

Since the estimation methods rely on the local reference points, the lack of rich reference points may be a challenge for these estimation methods. Two healing algorithms are proposed to address this issue and make the reference points richer in the vicinity of the estimation space. The performance of the healing algorithms for different maneuvers is also studied in this thesis.

A series of simulation and experimental tests with various road conditions are utilized for validating the estimation performance. Results show that the proposed algorithm can estimate different vehicles' lateral and longitudinal velocities. The estimation methods can estimate the lateral velocity with an error of less than 0.1 m/s. The methods can estimate the longitudinal velocity with an error of less than 2 kph if the proper reference data is provided.

Acknowledgements

Without the guidance and help of several individuals, conducting this research would not be possible. In the first place, I would like to express my sincerest gratitude to my supervisor, Prof. Amir Khajepour, for his valuable insight, patience, support, and encouragement for my research and education during my MASc program at the University of Waterloo. I appreciate many great lessons I learned during this two year program from my supervisor.

Thank you to Dr. Ehsan Hashemi, Dr. Amin Habibnejad Korayem, and Ahmad Mozaffari for their contribution and guidance. In my opinion, the constant support of the research team was crucial to the success of this project. I thank my friend, Mehdi Zabihi, and colleagues at the Mechatronic Vehicle System Laboratory for all their support.

I would like to acknowledge the sponsors of this project, the Natural Sciences and Engineering Research Council of Canada (NSERC), and General Motors. Special thanks to Dr. Bakhtiar Litkouhi, Dr. Alireza Kasaizadeh, and Dr. Annie Zhao at the GM Research and Development Center in Warren, MI, USA for their technical support and helpful comments in improving my research.

I would like to also acknowledge and thank Professor Kristen Morris and Professor Behrad Khamesee that accepted to read my thesis and provide me with their valuable comments.

I appreciate and thank all the supports I received from my parents and my brother during my life. Last but not least, I would like to show my deep appreciation to my dear wife, Rana, for all of her love, support, and understanding during our journey.

Dedication

To Rana

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, the problem will be explained, and the motivation of the thesis will be discussed. Moreover, the general outline of the rest of the thesis will be described.

1.1 Problem Statement and Motivation

Prediction and estimations of states are of great importance for vehicle control and safety. The conventional observers are designed and used vastly in the automotive industry to fulfill this objective. However, the vehicle behaviour can be nonlinear or unpredictable, and it is difficult or impossible to use linear or low-degree nonlinear observers to estimate vehicle states. These observers may fail to estimate the states correctly, in case of high slippery roads, combined slip situations, and maneuvers with intense steering inputs.

The main drawback of conventional observers is the need for a model. The model needs to be comprehensive and simple at the same time. A complex model will increase the computational complexity. While a simple model will decrease the system's estimation accuracy. Also, due to the complicated system behaviour, the accurate model may not be available in some cases.

A data-driven method is considered to address the described difficulties. The data-driven system does not need a model for estimation and only depends on the training (reference) data. In similar applications, neural network (NN) based methods have been used for state or parameter estimation. However, the NN-based methods require training a model, and this characteristic does not allow the estimation method to be modified online. Thus, a method that does not require training a model is of interest.

In this study, instead of black-box machine learning (ML) methods, non-parametric kernel regression methods are considered. In contrast to NN-based methods, these methods are interpretable. They also do not need to be trained in advance, and the computational complexity is less than NN-based methods. In order to make the algorithm more computationally efficient only the local data points need to be selected as reference points for each state estimation.

The train data set will be called the reference data set from now on, since no model needs to be trained. The reference data set contains all the sample points that can be used for the estimation. The method proposed in this study can perform state estimation for any vehicle using only one reference data set. The reference dataset does not need to be changed for different vehicles.

Another goal for this study is to make the estimation method explainable and interpretable. Hence, the reference data set can be enriched to improve the estimation accuracy.

1.2 Overall Estimation Structure

The overall estimation diagram utilized for this study is shown in Figure 1.1. This figure shows a general scheme of estimation approach used for both longitudinal and lateral velocity estimations.

As it is shown, the overall process consists of three main units. The input data unit takes the vehicle sensor measurements and normalizes them using the vehicle parameters. The reference dataset unit will generate the reference points for the estimation. And finally, the state estimation unit utilizes both other units outputs to estimate the intended state.

In the input data unit, measurements are collected from vehicle sensors and the desired feature set will be formed by utilizing vehicle parameters. The measurements include, but are not limited to, wheel speeds, lateral and longitudinal accelerations, yaw rate, and steering wheel angle. The input data unit will provide the input points for the kernel-based estimation.

The reference dataset unit provides the reference sample points for the kernel-based estimation. Normalized reference (sample) data will be used after the data reduction process. The process of building reference points and reducing them does not need to be performed online or repeated for each estimation. The normalized and reduced data set can be saved and used for the estimations.

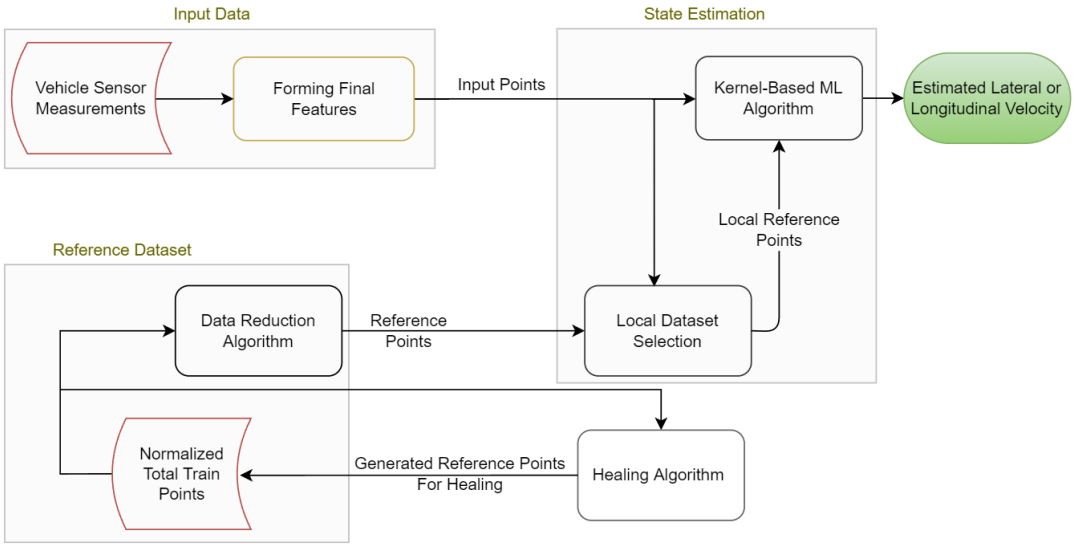

Figure 1.1: Diagram shows the overall procedures of this study.

The kernel-based ML method uses local points as reference points for the estimation. These points are selected by the local reference data set selection algorithm. This algorithm utilizes the feature importance to select the nearest and most effective set of reference points in the vicinity of the estimation point. In the state estimation unit, the local dataset is generated and the estimation is performed using the kernel-based method.

The proposed estimation methods use the local reference (sample) points for the estimation of each point. Therefore, the lack of proper sample points near the estimation point can reduce the estimation accuracy. So, by generating new sample points and adding them to the reference data set, there will be more sample points available near the estimation point, and it will enhance the accuracy of the estimation. This process will be called healing. A healing algorithm will be presented in this study, that will use a high fidelity simulation model to generate related vehicle data for the sparse regions of the reference dataset.

1.3 Thesis Outline

The rest of this thesis is organized as follows:

Chapter 2: A review of literature will be presented on data-driven methods used in vehicle systems-related subjects.

Chapter 3: In this chapter, the background information about the machine learning kernel-based methods, as well as other methods and their formulations, which are used in this study, will be presented. Kernel-based ML methods, Pearson correlation coefficient, data reduction algorithm, and local reference data selection will be explained.

Chapter 4: Feature selection for lateral and longitudinal velocity estimations will be explained and performed in this chapter. By considering the dynamic relations as well as correlation tests, the most efficient set of features will be selected for lateral and longitudinal velocity estimations. The data normalization method will be explained and implemented to make the estimation universal.

Chapter 5: This chapter is dedicated to longitudinal velocity estimation. The reference data will be normalized and three test results will be presented. Tests will be estimated using two kernel-based estimators, and the performance of the estimation algorithms will be studied for different driving scenarios and various road conditions.

Chapter 6: Lateral velocity vehicle-independent estimation will be performed in this chapter. First, the reference dataset will be made using the selected features, normalization, and data reduction. Then, Six test results, consisting of both simulation and experimental tests are presented and discussed. Different vehicle models, various driving scenarios, and diverse road conditions are used to observe the estimation performance. Two healing algorithms are proposed in this chapter and implemented to see their effects on estimation accuracy.

Chapter 7: The last chapter will be where the conclusions, as well as suggestions for future works, are made.

Chapter 2

Literature Review

In this chapter, a review of the related literature is presented. Since the main focus of this study is longitudinal and lateral velocity estimation using data-driven methods, the literature related to the data-driven vehicle parameter or state estimations will be reviewed.

Lateral and longitudinal velocity estimations have great and undeniable importance in vehicle safety and control systems. These estimations can be utilized in active safety control systems for driver assistance and crash avoidance. Some of them are anti-lock brake system (ABS), traction control system (TCS), and electronic stability program (ESP), to name a few. These states are not measured directly in vehicles because the sensors for measurements of these states are costly and not accurate. Thus, many model-based observers are designed for the estimation of mentioned states. However, model-based observers are not precise in many cases because of the non-linear and complex vehicle and tire behaviours [2, 3, 4, 5, 6, 7, 8, 9, 10].

Data-driven estimation methods are utilized to estimate vehicle parameters and states using the sensor measurements and input/output data sets. The data-driven systems take advantage of the captured vehicle dynamic attributes within the data. In contrast to the model-based vehicle estimations that need prior knowledge of the dynamic model of the system, the data-driven methods can be performed regardless of the availability of exact reference models [11, 12, 13, 14].

The following sections will present a review of methods used for a data-driven parameter or state estimation with application in vehicles.

2.1 Road Roughness Estimation

In [15], the sensor measurements from at least two connected cars are used to classify the road roughness category and roughness index using the artificial neural networks method. This study's test and train data are generated for two vehicles travelling with constant velocities. Features for estimation are selected using sensitivity analysis. The results are evaluated using numerical simulations, and they show a great performance.

In [16], road defects reconstruction and surface roughness classification is done by training an artificial neural network, using vehicle inertia measurement unit (IMU) data. One of the advantages of this method is the lack of need for excessive system characterization.

In [17], three ML methods of support vector machine (SVM), multi-layer perceptron (MLP), and random forest are utilized to estimate weighted longitudinal profile (WLP), which is an indicator of road roughness. The estimation performance and robustness are studied and compared. It is shown that although the three methods have nearly the same results, the neural network-based method, which is MLP, will lose its accuracy faster with decreasing the size of the feature train set.

A deep neural network (DNN) is formed using sparse auto-encoder and softmax regression in [18], for road excitation classification. This classification is used for a semi-active suspension system.

2.2 Power Management Estimation

With the increasing demand for electric vehicles and hybrid electric vehicles, the power management has gotten more attention. In [19], a battery state of charge (SOC) estimation method is proposed for electric vehicles. The proposed algorithm uses a subtractive clustering-based neuro-fuzzy system for this estimation. The estimation results using simulation data are compared to the results of two other methods, backpropagation (BP) and Elman neural networks. Furthermore, in [20] a method for real-time estimation of remaining useful life (RUL) of the battery is proposed that combines classification and regression attributes of multi-layer support vector machine (SVM) technique.

In [21] a deep reinforcement learning method is used for the task of power management between the engine and second energy source for hybrid electric vehicles (HEV). Also, a deep neural network system is used as a function estimator for approximating a policy for even not-visited states. It is shown that the proposed algorithm can adapt to new environments even trained in other environments; however, the algorithm may not be

fast enough because of the need to train a model. The performance of this algorithm is comparable to the optimization method of dynamic programming in the presence of prior knowledge.

Also, in [22, 23] a new combination of optimization algorithms, dynamic programming, and machine learning methods, neural networks are offered for HEV that is utilized for training online intelligent energy controllers. Two neural networks are used to classify the driving trends and predict the next steps of engine speed and optimal battery power. The networks will be considering driving ambient, roadway types, and traffic congestion. The methods offered in these papers rely on training a model offline using different driving scenarios.

2.3 Side Slip Angle Estimation

In [24], an integration of model-based and data-driven methods is used for sideslip angle estimation, using inertia measurement unit (IMU) measurements. An artificial neural network (ANN) is used to estimate a pseudo-side slip angle, which will be used in an unscented Kalman filter (UKF). The UKF implemented in the study is based on the kinematic model of the sensor instead of the vehicle dynamics. Also, another algorithm is proposed that enables the ANN to discover the best network architecture in a numerical environment.

In [25], a non-linear algorithm based on the neural network method is presented for sideslip angle estimation. The method does not need to know vehicle parameters to run the black-box regression models. It only uses the sensor measurements used in the electronic stability control (ESC) unit for estimation. This method can provide accurate estimation results within the domains and constraints of available train sets. The train set is vehicle-dependent, and the model needs to be trained again for different vehicles. This will increase the complexity and costs of this method.

Similarly, another method for sideslip angle is proposed in [26], using general regression neural network (GRNN) and driver-vehicle closed-loop system. It is shown that the GRNN has good accuracy and has a faster convergence rate than BPNN. Also, paths with a higher degree of complexity will result in greater errors.

In [27], also an RNN method is combined with the kinematic vehicle model and implemented for sideslip angle estimation. It is shown how a simplified vehicle model utilized with the RNN can result in competitive outcomes. However, the trained model in the

study was not generic and could not be implemented on other vehicles before retraining and considering their models.

2.4 Mass Estimation

Data-driven methods have been used in the literature for mass estimation. In [28] an estimation method for static gross or axle weight estimation is proposed that works with general feedforward neural network (FFNN). This method improves the multiple-sensor weigh-in-motion system by taking suspension deflections into account.

Closely in [29], another FFNN is used for road grade and vehicle mass estimating. The proposed FFNN works based on the measurements of velocity, acceleration, and engine torque in the current and last steps.

By considering the longitudinal dynamic model, a recursive least square algorithm is presented in [30]. Also, the driveline efficiency and road grade angle are estimated for implementation in the mass estimation problems.

There are other works in the literature that use the recursive least square algorithm for mass and road grade estimation, like [31]. The recursive algorithm is implemented using multiple forgetting factors to respond to different variation rates.

2.5 Other Applications

In [32], a nonlinear autoregressive network with exogenous inputs (NARX) is utilized for assisting the dynamic system identification algorithm, in a path tracking problem. NARX algorithm is used to predict the system response over the prediction horizon, and it will be implemented as future inputs for non-linear MPC. Also, another trajectory tracking problem is studied in [33]. The machine learning method of local recurrent neural networks (RNN) is used for online system identification. The proposed algorithm is applied for a remotely operated vehicle (ROV). A comparison is made between local RNN and three-layer NN. It is concluded that the local RNN learns faster and has the required accuracy.

In [34], a review on some of the vehicle lidar signal processing methods and their different features presented, which are applicable in autonomous vehicles. It is explained that the estimation of moving object location and estimation of signal, parameters can be performed by superresolution algorithms, such as multiple signal classification (MUSIC)

[35], and rotational invariance technique (ESPRIT) [36, 37]. They have also used the signal model's 2-D fast Fourier transform (FFT) to get the joint estimation of distance and angle. The ML algorithms proposed in this article offer a great resolution in exchange for increasing the computational costs.

A 3-layer BP neural network is utilized to predict the road inclination as well as real-time optimizing the rollover model in [38]. The road inclination prediction is being used in a 3-DOF rollover model to study the proposed new rollover index. In another work, the rollover of a liquid-carrying tank truck is modelled and predicted using BP neural network [39]. The model is trained using simulation data, and it is shown the proposed algorithm has an accurate estimation within the train set boundaries.

In [40], two supervised, and unsupervised learning methods are used to classify the braking intensity. An unsupervised Gaussian mixture model is used for clustering and determining different levels based on the braking pressure. A random forest model is used for classification. Also, another feedforward NN is used that utilizes CAN bus signals to quantitatively analyze the brake intensity.

In [41], an RNN is trained for estimation of roll angle, and it is integrated with a model predictive control. The data-driven algorithm performs the estimation task using steering wheel angle input, lateral acceleration, and control system outputs. Similarly, [42] an ANN method is deployed for real-time roll angle estimation. The algorithm inputs are selected based on the direct measurements of available states.

A comparison is made [43] between four statistical learning methods for internal combustion engine torque estimation. Four methods of linear least squares, linear and non-linear NNs and support vector machines are used to estimate the engine torque using available measurements of accelerator pedal position, engine rotational speed, and vehicle speed. While linear models had difficulties estimating the torque, especially at lower gears, the non-linear models of SVM and non-linear NN showed accurate results for torque estimation.

In [44], integration of GPS and INS systems using an ANN is presented. The proposed algorithm helps to improve the location estimation in case of GPS signal loss or NS signal errors.

2.6 Conclusion

Considering the referenced studies in this chapter, the data-driven methods are widely considered helpful for vehicle state or parameter estimation problems. In some cases,

the data-driven methods can perform without prior knowledge of exact system models. In others, a system model, which is not necessarily exact, contributes to the estimation process. In contrast to the data-driven methods, all the model-based approaches depend on the availability of a model describing the system.

One of the most popular groups of algorithms utilized for this purpose is neural network-based algorithms. Although NN algorithms can perform the estimations or classifications in real-time, training a model is time-consuming. So, it is not possible to refine or change the train set and perform estimation simultaneously.

Furthermore, the black-box algorithms, such as NN algorithms, are not easy to interpret. Also, they offer a global solution considering all the train data.

In this study, three main factors are considered. First, the estimation is going to be performed using traditional machine learning algorithms, which allows analyzing the features, test set, train set, and other aspects of the estimation. Second, the estimation is carried out with local data. So, only the local reference points are considered for each point estimation. Third, the estimation is made universal and applicable for different vehicles. It means that the estimator does not need to be modified or changed if the vehicle is changed. The generic algorithm can be implemented in different vehicles estimation if the data is normalized properly.

There are not many works found in the literature related to the application of traditional machine learning algorithms for state or parameter estimation to the vehicle state or parameter estimation. The vehicle-independent scheme that is used in this study is a novel approach that has not been used before.

Chapter 3

Background

In this chapter, the background knowledge used in this thesis will be briefly discussed. In the following sections Nadaraya-Watson (NW) method, GPR method, and Gasser & Müller method will be briefly explained. The Pearson correlation coefficient will be discussed in the next section. Also, the data reduction algorithm used in this project will be explained, afterward.

3.1 Nadaraya-Watson (NW)

Nadaraya-Watson Kernel Regression (NWKR) is an estimation method that interpolates the output value for a given point, based on the observations around that area. This algorithm works based on the weighting average. The weights used in the NW method are calculated using a kernel function. In this kernel function, based on the distance between the point of interest x and observations around that point (reference data) X_i a value will be assigned. [45][46] NWKR uses the following equation to estimate $\hat{y}$ for the point of interest x.

$$\hat{y} = \frac{\sum_i^n K(x, X_i) y_i}{\sum_i^n K(x, X_i)} \tag{3.1}$$

In Equation (3.1) X_i is the i^{th} observation and x is an arbitrary point that is to be estimated. Also, $K(x, X_i)$ that is used as weights in this equation shows the kernel function. This kernel function should satisfy three conditions:

- $K(x, X_i) \geq 0$
- $\int_{-\infty}^{+\infty} xK(x, 0)dx = 0$
- $0 < \int_{-\infty}^{+\infty} x^2 K(x, 0)dx < \infty$

In the NWKR method, the controlling factor lies within the kernel function used in the equations. For example, for Gaussian kernels, the bandwidth and smoothing factors can be used to tune the algorithm. Since the NWKR method performs based on the weighting average, does not need to be trained in advance. Because of the fact that weights are calculated based on the distance, this method is robust to the outliers.

Furthermore, this method relies on reference data in the vicinity of the test point to estimate the test point output. This property results in the fact that adding more data to reference data will help the algorithm to have more observations available to make decisions based on.

Another main characteristic of this method is the ability to be applied locally. This feature enables the estimation algorithm to perform with a limited number of points in the neighbourhood. So, the estimation will be local and the global trend will be less effective on the estimation. The local estimation will also help the algorithm to perform computationally efficiently. Hence, working with a big database will rise no issues for this method.

One of the functions that can be used here to assign weights in the NWKR method, is **Gaussian Kernel** , as in Equation (3.2).

$$K_\sigma(x, X_i) = \frac{1}{\sqrt{2\pi}} exp(\frac{-(x - X_i)^2}{2\sigma^2}) \tag{3.2}$$

Where σ shows the bandwidth factor, which controls the smoothness of the kernel function. The Gaussian kernel K_σ is a function of the square norm of the distance between the sample point and the point of interest. The closest observations to the test point x will get the greatest weights in averaging.

The Gaussian kernel function can perform in one dimension. However, for most of the estimations, more than one feature is contributing. Each sample point can be assumed as a point with d features. The i^{th} sample point around a test point can be represented by

X_i:

$$X_i = \begin{bmatrix} X_i^1 \\ X_i^2 \\ \vdots \\ X_i^d \end{bmatrix} \qquad i = 1, 2, \cdots, n \tag{3.3}$$

There are two solutions suggested for this scenario. Using a product of univariate Gaussian kernels, and using multivariate Gaussian kernel.

3.1.1 Product of Univariate Gaussian Kernels

One of the approaches to deal with the multi-features of each point is to assume d different univariate kernel functions for d features [47]. Each of these kernel functions can have its own smoothness factor σ_p. In this case, the Equation (3.1) will be as follows:

$$\hat{y} = \frac{\sum_i^n y_i (\prod_{p=1}^d K_{\sigma_p}(x^p, X_i^p))}{\sum_i^n (\prod_{p=1}^d K_{\sigma_p}(x^p, X_i^p))} \tag{3.4}$$

In this method, the kernel functions are working separately and it is assumed that the kernel distributions are independent. Hence, correlations between features are not captured, completely. This is the main reason that this method is not ideal to apply to the vehicle state estimation. Since the production of all kernel functions is used, in case one kernel function has a small value, the overall weight will be very small. So, in this formulation, only those observations will have a significant influence on the estimation that all of their features are very close to the respective feature of the estimation point. This method can be used and implemented in scenarios, where features are completely independent.

3.1.2 Multivariate Gaussian Kernel

Multivariate Gaussian kernel is another function that captures the correlation between all the features by taking the covariance matrix into account. In the multivariate Gaussian kernel function, a positive definite matrix H will be used to reflect the correlation between different features. This positive definite matrix can also be an arbitrary diagonal matrix with equal positive diagonal elements. In this case, the result of the multivariate kernel

function will be the same as the product of univariate Gaussian kernels [48]. The kernel function is

$$K(x, X_i) = \frac{1}{(2\pi)^{d/2} |H|^{1/2}} exp(\frac{-(x - X_i)^T H^{-1}(x - X_i)}{2}) \tag{3.5}$$

3.2 Gaussian Process Regression

Gaussian Process Regression(GPR), also known as Kriging, is a method of interpolation based on the Gaussian process in the presence of prior covariances. With appropriate assumptions for the priors, GPR can result in the best linear unbiased prediction (BLUP). This method can also be studied as a form of Bayesian inference. In the first step a prior distribution over functions, in the form of a Gaussian Process, will be assumed. By observing a set of reference points, the estimation function will be updated and modified. This method combines the Gaussian prior with the Gaussian likelihood function for all of the reference points. The posterior which is also Gaussian will be used as the estimator to estimate the test points [49].

As it is shown in [50], one of the applications of the Gaussian Processes is to define distributions over functions. These distributions will get updated with training samples. Equation (3.6) assumes that the prior function is distributed as a GP with mean function m and covariance function k. The posterior function f will be a normal distribution with the mean function μ and covariance function Σ for the corresponding x's

$$\begin{aligned} f_{prior} &\sim \mathcal{GP}(m, k) \\ f_{posterior} &\sim \mathcal{N}(\mu, \Sigma) \end{aligned} \tag{3.6}$$

If the known function values for training points are given as f and the f_* is a set of function values corresponding to the test points X_*, the joint distribution can be written as

$$\begin{bmatrix} f \\ f_* \end{bmatrix} = \mathcal{N}(\begin{bmatrix} \mu \\ \mu_* \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma_* \\ \Sigma_*^\top & \Sigma_{**} \end{bmatrix}) \tag{3.7}$$

in which the terms are representing: $\mu = m(x_i), i = 1, ..., n$ for training means, and, in the same manner, the μ_* for the test means. The covariances are also used as Σ for the training set, Σ_* for the training-test covariances, and Σ_{**} for the test set covariances.

In the GPR method, the conditional distribution of f_* given f will be calculated using the joint distribution. As it can be seen, this method will produce a Gaussian process and update it in light of training points. Therefore, one of the main properties of this method is the need of updating the mean and covariance functions in case of additional training points. This will make this method slower than some other kernel-based methods like the Nadaraya-Watson Kernel Regression method. However, because of the fact that the GPR method will calculate a distribution for each estimation, a confidence region, e.g. 95% confidence interval, can be extracted. Despite other kernel-based methods that only

calculate a final value for the test point, the GPR method, in addition to the expected value, can present intervals for a given level of confidence.

The GPR estimations mentioned in this study, are performed using a MatLab function [51]. The following line of code is used to automatically fit a GPR model to the reference points. The input features will be given to the function as `x` and the output as `y`.

```
gprModel = fitrgp(x,y,'KernelFunction','squaredexponential');
```

The squared exponential kernel function, which is also known as the exponentiated quadratic kernel, is also used in Equation (3.2). The default kernel parameters are used for this estimation. In this study, every GPR test will be performed using a small set of local reference data. Because of time constraints in online vehicle state estimation, a small group of reference points around each test point will be selected to be used in the GPR method. These reference points will be the nearest points to the test point. Based on trial and error to maintain the estimation accuracy as well as keep the run time low, 30 nearest reference points are selected in the following tests.

3.3 Gasser&Müller

Similar to NWKR, Gasser&Müller(G&M) is another kernel-based local regression method that can be applied to a wide variety of problems. This method utilizes the data around the estimation point and assigns each of them a specific weight to calculate the output using a weighting average algorithm [52].

Assuming that the measurements are available at $X_1, \cdots, X_n \quad (0 \leqslant X_1 \leqslant \cdots \leqslant X_n \leqslant 1)$, the governing equation and modifications of G&M method for univariate situation is presented as following:

$$\hat{y}(x) = \frac{\sum_i^n y_i(\int_{s_{i-1}}^{s_i} K(x,u).du)}{\sum_i^n(\int_{s_{i-1}}^{s_i} K(x,u).du)} \tag{3.8}$$

In Equation (3.8), $K(x,u)$ is a kernel function that assigns weights. The intervals of integrals are selected based on the fact that $0 = s_0 \leqslant s_1 \leqslant \cdots \leqslant s_n = 1$. Also, for every X_i there should be $s_{i-1} \leqslant X_i \leqslant s_i$.

The G&M method resembles the NWKR method. However, there are some differences, when the algorithms are used to estimate in multivariate space. Since, in the G&M method, integral intervals are needed to be assigned based on the value of features, a complexity will happen in multi-dimensional space, where features are not necessarily arranged.

For the G&M method, the factor that can affect and control the estimation lies within the kernel function used in the equations. Although it seems the integral intervals can be another controlling factor, the intervals should satisfy:

$$\max_j |s_j - s_{j-1}| = O(\frac{1}{n})$$

Since the G&M method performs based on the weighting average, it is not associated with the need of training a model. The same as NWKR, this method uses a weighting average, and weights are calculated based on the distance, so this method is robust to the outliers, too.

This method is also considered a local estimation method and adding more data in the vicinity of the estimation point will empower the method for a more accurate estimation. Moreover, considering the fact that this method does not need to train a model to estimate, it is very useful for local estimation using the available data in the neighbourhood.

3.4 Pearson Correlation Coefficient Test

Based on the Pearson Correlation Coefficient (PCC), the linear correlation between two features can be calculated statistically [53, 54]. The PCC can be also used as a measurement of the linear correlation between two variables. So, two variables can be related but not linearly correlated. It is because of the fact that the correlation shows only the linear relationship. In PCC, based on the covariance matrix of features, the correlation coefficient will be calculated. In Equation (3.9), ρ, which is known as the population Pearson correlation coefficient, is presented.

$$\rho_{ij} = \frac{Cov(x_i, x_j)}{\sqrt{Var(x_i)Var(X_j)}} \tag{3.9}$$

When the correlation test is applied to a sample set, it will be shown by r. If the sample set is consist of n pairs of (x_i, y_i) and the sample mean values for x and y variables are given by $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. The sample Pearson correlation coefficient can be calculated as:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3.10}$$

In this equation, n is the number of sample points, and r_{xy} will be the sample Pearson correlation coefficient of x and y.

3.5 Data Reduction

Since the methods going to be used in this study are data-driven methods, the reference data has great importance. reducing the size of reference data as well as removing duplicated data, not only reduces the computational complexity but also improves the estimation accuracy and prevents bias errors.

3.5.1 Reduction Algorithm

The general idea of the data reduction algorithm is to detect overcrowded regions and remove some of the least informative points in that region. To implement this idea, the K-means method will be used to partition data into k clusters. The clusters with more points than a certain threshold number will be reduced by removing the least informative points or the points within tight clusters to meet the threshold.

Considering $[x_1, x_2, x_3, ..., x_n]$ a set of points that need to be divided into $k(\leq n)$ clusters by K-means method. the K-means algorithm will divide the n points into k clusters, shown as $s = \{S_1, S_2, S_3, ..., S_k\}$, in order to minimize the within-cluster sum of squares as:

$$\arg\min_{s} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 \tag{3.11}$$

In the above equation, μ_i is the mean of points in S_i. This optimization problem will be solved using built-in functions in MatLab for this study.

In this study, the number of clusters will be set to 50 for data sets with more than 2000 points. For data sets with fewer points, the number of clusters will be less. The maximum number of points in a cluster, cluster threshold, will be set to the total number of points divided by the number of clusters:

$$\text{cluster threshold} = \frac{n}{k}$$

After determining the number of clusters, cluster thresholds, and all the partitioned data, the reduction process will begin. An additional number of points will be deleted randomly from those clusters with more points than the cluster threshold. After performing this task on all the clusters, the number of points in each cluster will be less than or equal to the cluster threshold. The task of clustering using K-means and reducing the data volume will be repeated until the total data set converges to its final value.

In Figures 3.1 and 3.2, two examples are provided to study the performance of the reduction algorithm. In these figures, red circles show the original data points, and blue dots show the reduced version of the original data set. Each point in the data sets has multiple dimensions, however, only velocity versus wheel angle is depicted.

In Figure 3.1, a data set used for lateral velocity estimation is shown. As it is seen in plot legends, the total number of original points is about $252k$ points. After reduction, the total number of points in the data set has been reduced to about $38k$ points. So, the original data set is reduced approximately by 85%. The overall view of the data set shows that almost none of the points in border regions, which are considered as sparse regions, has been removed, and the general form has been reserved.

One of the crowded regions of the reference data related to the lateral velocity estimation is shown in the right plot in Figure 3.1. Since most of the tests start and end with zero steering input, this is expected. The zero steering input is a default scenario for starting and ending status of most of the tests. So, the number of repeated points with a low value for steering signal is comparatively larger than other regions. As it can be seen, $131k$ points were originally located in a small region, defined by $|V_y| < 1(kph)$ and $|WA| < 1(deg)$. This is almost half of the total number of points within the original data set. After reduction, this number has been decreased to $4k$ points. So, the original data set in this region is reduced approximately by 97%. This is way more than the reduction percentage on the main scale.

It is observed that the reduction algorithm has reduced the data volume in overcrowded regions and reserved the overall structure of the main data set. Figure 3.2 shows similar results as observed in 3.1. The total number of points in the original data set is about $307k$ and it is reduced to $58k$ after reduction. Considering a smaller dense region, where $V_x < 80(kph)$ and $|WA| < 1(deg)$, the total number of points has been reduced to $6k$ from $130k$. These numbers show the exact results that the proposed reduction algorithm can properly maintain the overall structure of the data set and remove the points from dense and overpopulated regions.

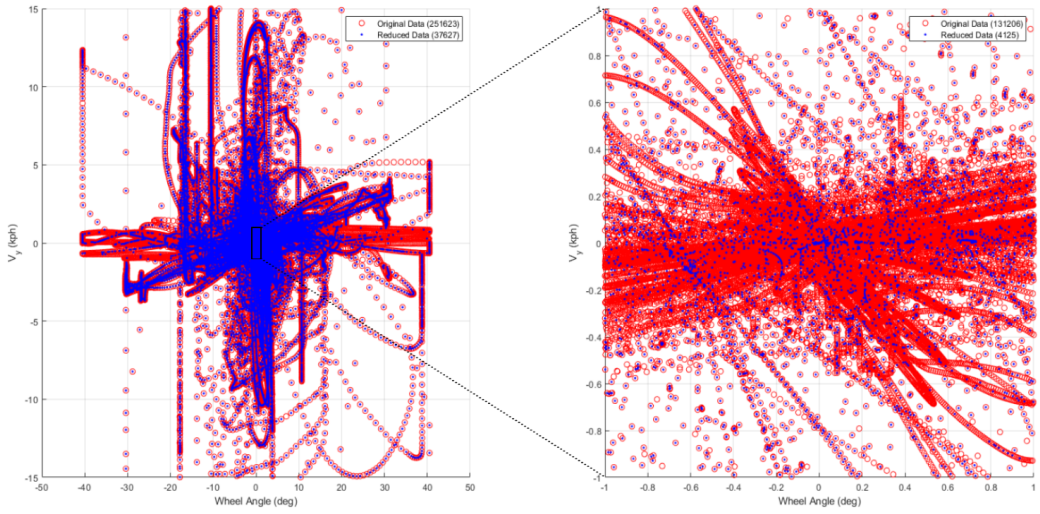

Figure 3.1: Example 1 for reduction algorithm. Right plot is the zoomed area of the left side portion.

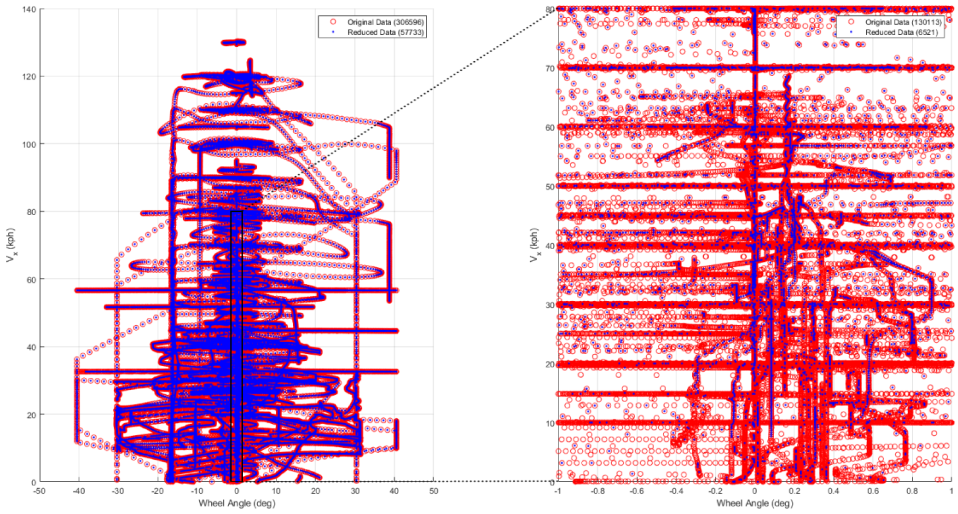

Figure 3.2: Example 2 for reduction algorithm. Right plot is the zoomed area of the left side portion.

3.6 Local Reference Data Selection

The methods used in this study are kernel-based methods that use reference data for estimation. Methods will be utilized with local reference data to be more efficient and accurate. These reference points are the nearest data to the point of interest. For data selection around a point, two criteria should be considered.

- The maximum distance of each sample point to the point of interest is considered and the distance measurement is based on Euclidean distance measurement.
- Feature importance is considered based on the correlation between input and output features.

To address these criteria first all the features need to be normalized in size. Hence, all the feature values will be projected to [0,1]. The maximum and minimum amount of each feature will be projected to 1 and 0. This will help to have a scale-free scheme in data selection. The size normalization process for the i^{th} feature of point X is:

$$X_i{}' = \frac{X_i - max(X_i)}{max(X_i) - min(X_i)} \tag{3.12}$$

3.6.1 Unit Ellipsoid

For data selection around a point, it is needed to consider the feature importance. To consider the feature importance, a modified unit ball (unit ellipsoid) method is suggested. In this method, reference points within a unit ellipsoid will be selected. The test point is located at the center and the radius along each feature axis will be determined based on that feature importance. Feature importance F_i will be calculated based on the correlation between that feature and the output feature (ρ_i):

$$F_i = \frac{|\rho_i|}{\sum_i |\rho_i|} \tag{3.13}$$

For calculating the distance, Euclidean distance between normalized test point x' and normalized reference points X' will be considered as:

$$d(x', X') = \sqrt{\sum_i (x_i{}' - X_i{}')^2} \tag{3.14}$$

Based on the feature importance, the radius of that ellipsoid along the axis i will be as:

$$r_i = r_0/(F_i^2) \tag{3.15}$$

where r_0 is an arbitrary radius picked to construct the ellipsoid. The amount of this r_0 will be shown in the results. Typically, it will be about 0.01 or 0.02. Naturally, the bigger r_0 gets, the more reference points will be picked for the estimation.

In this method, the unit ellipsoid will be stretched along with more correlated features, and it will be less stretched along with less correlated features. For example, Figure 3.3 shows a randomly generated data set. In this figure, based on the ellipsoid's orientation, it can be said that the feature i is more correlated with the output feature than the feature j. As it can be seen, the unit ellipsoid is depicted around two arbitrary test points. The enclosed reference points in each ellipsoid are used for the local estimation in each method.

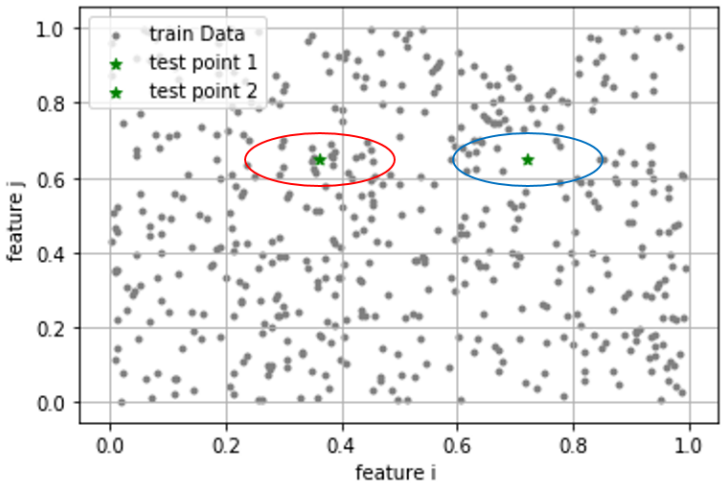

Figure 3.3: Ellipsoid Local Data Selection.

3.6.2 KNN for Sparse or Dense regions

Since the reference data are not uniformly distributed all-over the sample (reference) space, the number of reference points selected by the unit ellipsoid method can differ greatly for different test points. As it can be seen in Figure 3.3, compared to the blue ellipsoid on the right, the red ellipsoid on the left encompasses more reference points.

If the number of reference points in the unit ellipsoid is very high or very low, the K-nearest neighbours algorithm will be used to pick a certain number of reference points

among all the data around that test point. For upper and lower thresholds of this method, 400 and 20 are considered. So, if the number of reference points in the ellipsoid is lower than 20, 20 nearest neighbours will be picked for estimation. Also, if the number of reference points in the ellipsoid is greater than 400, only the nearest 400 reference points will be selected to be used as reference points in the estimation. These two thresholds are calculated based on trial-and-error on estimation tests. These values help the algorithm to minimize the run-time (maintain the run time less than 10ms), while maintaining the estimation accuracy.

Chapter 4

Feature Selection for Lateral and Longitudinal Velocity Estimation

In this chapter, the suggested system for vehicle-independent lateral and longitudinal velocity is discussed. As mentioned before, kernel-based machine learning methods are used for this data-driven estimation. For implementing the machine learning algorithms, features need to be selected and modified. However, two main objectives need to be considered for the feature selection task. First, the feature selection and data normalization must be done so that the estimation be universal. Secondly, only the most effective features are considered. A lower number of highly correlated and effective features will increase the accuracy of the estimation algorithm and save the computational cost.

In the following sections, the feature selection for lateral and longitudinal velocity estimations are discussed. Based on the previous works, and correlation tests the most effective set of features are selected for each estimation.

4.1 Feature Selection for Longitudinal Velocity Estimation

In this section, the features for longitudinal velocity estimation are selected. For the selection of these features, three main steps are considered:

1. The dynamic relations of the system should be considered to find the initial list of the features.
2. The features need to be dimensionally normalized to make the estimation universal and vehicle-independent.
3. The correlation test is performed on the features to distinguish the most correlated features and the final set of inputs features for estimation.

4.1.1 Dynamic Relations

Traditionally, the main source of longitudinal velocity estimation is wheel speeds. On one hand, it is an easy estimation that does not need any complicated and expensive equipment, and on the other hand, it can be inaccurate in many scenarios, e.g. slippery road, or in turns that wheels are showing various speeds. In many model-based velocity estimations, like [3, 2], the wheel speeds contribute to modifying observer gains or enhance the estimation accuracy in another way. In addition to wheel speeds, other measurements are used in model-based estimations. These measurements are yaw rate r, lateral acceleration a_y, longitudinal acceleration a_x, and steering wheel angle δ. Other sensor measurements are also used, e.g. roll rate and pitch rate, to estimate other states in the vehicle to enhance the velocity estimation accuracy. It is assumed that all of these measurements are available from IMU or other individual sensors.

Based on the dynamic governing equations shown in (4.1), the most correlated measurements are used for this estimation. Furthermore, since the kernel-based estimation methods are not able to capture the complex relations between the features, the final set of features need to be selected as small as possible. Failing to do so will cause computational complexity.

$$\begin{aligned} a_x &= \dot{u} - vr \\ a_y &= \dot{v} + ur \end{aligned} \tag{4.1}$$

4.1.2 Vehicle Normalized Features

Another aspect of feature selection for longitudinal velocity estimation is to make this estimation universal. In order to do so, the vehicle parameters related to this estimation need to be incorporated. Tire effective radius and steering ratio are the two main factors that need to be used to normalize the features. The tire effective radius $R_{e,i}$ is used to transform the wheel spins ω_i to the wheel speeds $R_{e,i}\omega_i$. Also, the steering ratio (SR) is used to calculate the wheel angles δ/SR. Incorporating vehicle and tire parameters will normalize the features and allow the estimation to be performed universally. It means that the reference data are not limited to an specific vehicle, and it can be used for all other vehicles at the same time.

4.1.3 Correlation Test

For the correlation test, the features nominated in previous sections are used. The correlation between every feature and the longitudinal velocity is calculated based on the Pearson correlation test, discussed in Section 3.4. The correlation test result that is presented in Figure 4.1 is performed on the data that is presented in Section 5.1. It is helpful to remember that the Pearson correlation test is only capable of showing the linear correlation between features. Although the correlations between the output and the wheel angle and longitudinal acceleration are small, the effect of these features is undeniable. Based on the preliminary set of tests, the effect of yaw rate r can be useful in the estimation of longitudinal velocity. Also, it was seen that the absence of lateral acceleration a_y can be compensated by other features.

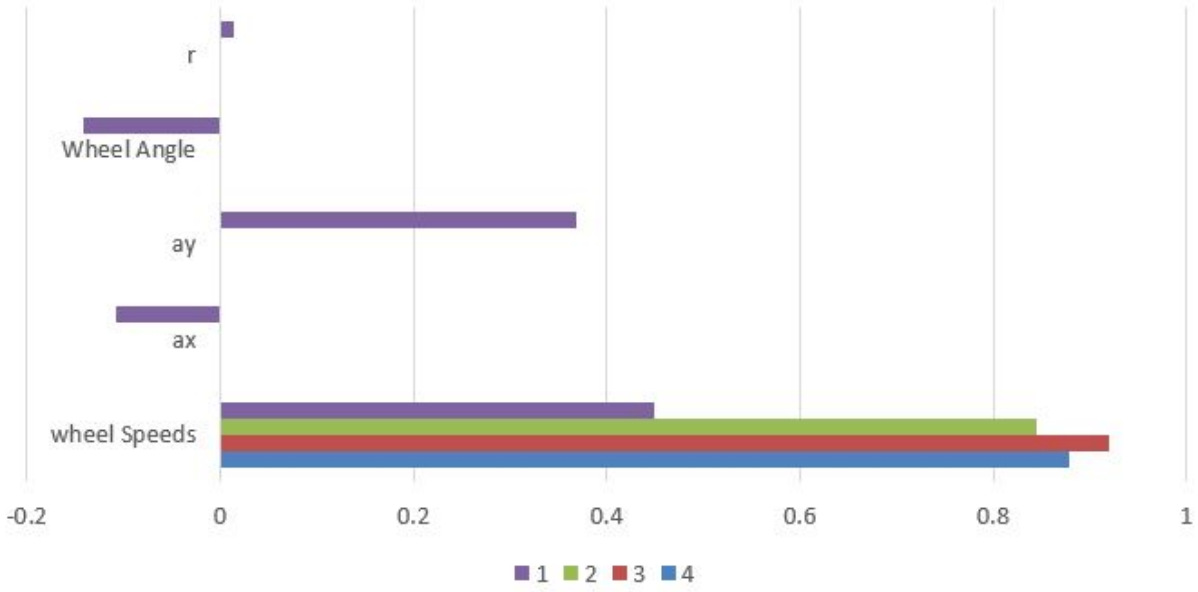

Figure 4.1: Pearson correlation coefficient for longitudinal velocity estimation features.

4.1.4 Final Features

Based on the discussions done, the final set of features are selected. These features consist of 4 wheel speeds $R_{e,i}\omega_i, i \in [1,2,3,4]$, yaw rate r, wheel angle δ/SR, and longitudinal acceleration a_x. The procedure scheme for longitudinal velocity is also depicted in Figure 4.2.

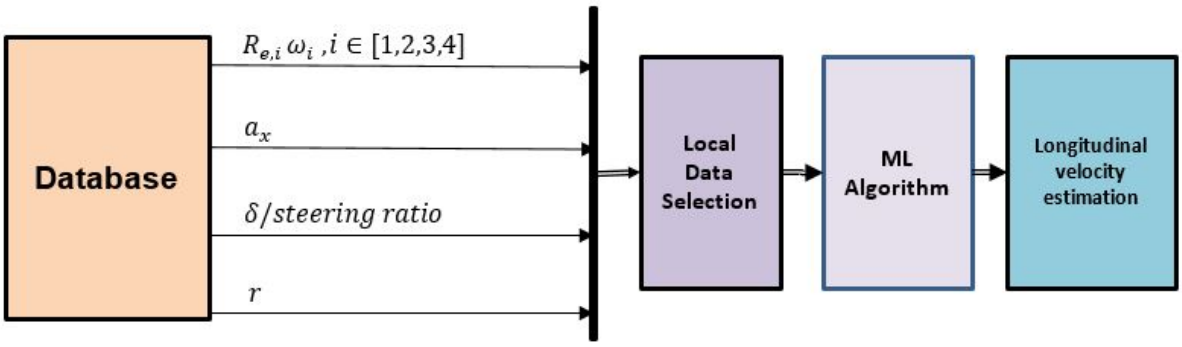

Figure 4.2: Overall estimation procedure for V_x.

Based on this chart, reference data with these features are picked in the neighborhood of the test point. The local data are used in the machine learning algorithm and the longitudinal velocity estimation is performed.

4.2 Feature Selection for Lateral Velocity Estimation

In this section, the effective features for lateral velocity estimation are selected. Similar to what is done for longitudinal velocity estimation, for the selection of these features, three main steps are considered:

1. A vehicle model is considered and the dynamic relations of the system are studied to find the initial list of the features.
2. The features are normalized to make the estimation universal and vehicle-independent.
3. The correlation test is performed on the features to distinguish the most correlated features and the final set of inputs for estimation.

4.2.1 Vehicle Model and Dynamic Relations

The vehicle model is mainly considered to study the effective parameters in the lateral velocity estimation. As shown in Figure 4.3, for this purpose the 2DOF bicycle model is considered. This is a simple lateral model that provides the lateral velocity and yaw rate based on the tire forces. The bicycle model's governing equations for lateral dynamic are [55, 56, 57]:

$$\begin{aligned} m(\dot{v} + ru) &= F_{xf}\sin\delta_f + F_{yf}\cos\delta_f + F_{yr} \\ I_z\dot{r} &= a(F_{xf}\sin\delta_f + F_{yf}\cos\delta_f) - bF_{yr} \end{aligned} \tag{4.2}$$

In these equations, Fs are tire forces and f, r subscriptions are for the front or rear axles. Also, subscriptions x, y symbolize longitudinal and lateral directions. Other parameters are yaw rate r, longitudinal velocity u, lateral velocity v, front wheel angle δ_f, front axle to CG distance a, rear axle to CG distance b, vehicle mass m, and yaw moment of inertia I_z.

To expand the equations and use them for further calculations, a tire model is needed to be used for calculations of forces. Since this expansion is not necessary for this study, only one method is shown for further suggestions.

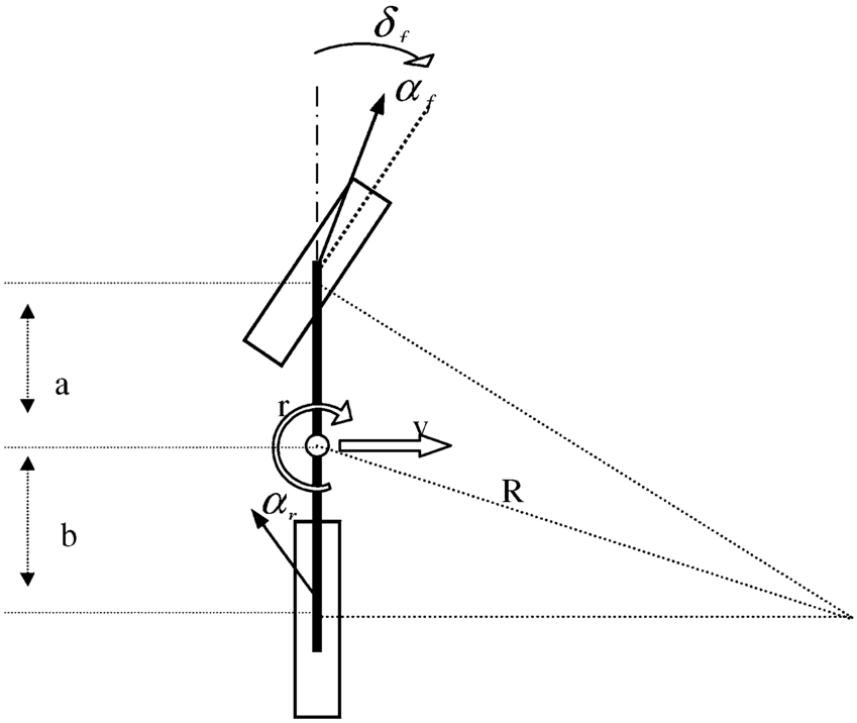

Figure 4.3: 2DOF Bicycle Model.

As shown in Figure 4.3, α_f, α_r are parameters for front and rear slip angles and can be calculated by:

$$\begin{aligned} \alpha_f &= \delta_f - \frac{v + ar}{u} \\ \alpha_r &= \frac{br - v}{u} \end{aligned} \tag{4.3}$$

A linear tire model suggests that the lateral forces of each axle can be taken as $F_{yf} = C_{\alpha_f}\alpha_f$, $F_{yr} = C_{\alpha_r}\alpha_r$. The coefficients of cornering stiffness are shown by $C_{\alpha_f}, C_{\alpha_r}$. For small wheel angles, the effect of longitudinal forces can be neglected and the lateral dynamic model with linear tire-vehicle models can be simplified as following:

$$\begin{bmatrix} \dot{v} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{C_{\alpha_f}+C_{\alpha_r}}{um} & -(u + \frac{aC_{\alpha_f}-bC_{\alpha_r}}{um}) \\ -\frac{aC_{\alpha_f}-bC_{\alpha_r}}{uI_z} & -\frac{a^2C_{\alpha_f}+b^2C_{\alpha r}}{uI_z} \end{bmatrix} \begin{bmatrix} v \\ r \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha_f}}{m} \\ \frac{aC_{\alpha_f}}{I_z} \end{bmatrix} \delta_f \tag{4.4}$$

As mentioned before, for feature selection two main factors that need to be considered are the dynamic governing equations and universal estimation. Based on the dynamic Equations (4.4), for lateral velocity estimation main features are δ_f,r,$\dot{r}$,u, and $\dot{v}$.

4.2.2 Vehicle Normalized Features

Due to the fact that this research aims to make the estimation universal, this section will focus on this aspect. The goal is to select normalized features that can be utilized for estimation, regardless of the vehicle type. It is seen that in Equations (4.2) $\dot{r}$ and $\dot{v}$ are linked to the coefficients I_z and m. So, instead of $\dot{r}$ and $\dot{v}$, normalized versions are used, which are $I_z\dot{r}$ and $m\dot{v}$. From Equations (4.1), the derivative of lateral velocity can be written as $\dot{v} = a_y - ru$.

Another point about the yaw acceleration is that it might not be available through measurements in the vehicle. However, the yaw rate is available from the inertia measurement unit (IMU). Therefore, the yaw acceleration is calculated by taking the derivative of the yaw rate numerically. However, this will intensify the noise effects. So, in this project, the derivative of the yaw rate is calculated based on the yaw rate change in the last three steps. The steps are supposed to be 25ms each. It means that three consecutive measurements with $40Hz$ frequency are used to measure the yaw acceleration. Hence, the term $I_z\dot{r}$ is equal to $I_z\frac{\Delta r}{\Delta t}$ from now on.

Four wheel speeds are considered as $R_{e,i}\omega_i$. This is the normalized form of wheel spins. Regardless of the vehicle type, these features help to have a speed measurement for each corner.

One of the main inputs to the vehicle system is the steering wheel angle (SWA). To make it more independent from the vehicle, the front wheel angle is taken into account. So, δ that is showing the SWA is divided by the steering ratio, which is the ratio between driver steering input and the front wheel angles.

Another factor that needs to be taken into account for lateral velocity estimation is the lateral acceleration itself. To make it more compatible with vehicle independent estimation, a product of mass and lateral acceleration ma_y is used as another feature for this estimation.

As it can be seen in Equation (4.4), the geometry of the vehicle has a great impact on the lateral velocity model. It is necessary to choose some other features to address this matter. Lateral corner accelerations are considered to address this issue. These accelerations can reflect the geometry of the vehicle because of the vehicle parameters that are included in

them:

$$
\begin{aligned}
a_{yFL} &= a_{yCM} + \frac{1}{2} T_w r^2 + a\dot{r} \\
a_{yFR} &= a_{yCM} - \frac{1}{2} T_w r^2 + a\dot{r} \\
a_{yRL} &= a_{yCM} + \frac{1}{2} T_w r^2 - b\dot{r} \\
a_{yRR} &= a_{yCM} - \frac{1}{2} T_w r^2 - b\dot{r}
\end{aligned}
\tag{4.5}
$$

In the set of Equations (4.5), the geometry is captured within the features. In these equations, a_{yCM} is the lateral acceleration at the center of gravity of sprung mass, T_w is showing the wheel track, a is the front axle distance to CG, and b is the rear axle to CG distance. Also, the FL subscription shows the front left corner. Lateral acceleration at other corners is shown with similar subscriptions.

4.2.3 Correlation Test

In this section, the correlation coefficient between the input features and the output feature, which is the lateral velocity, is studied. The Pearson correlation test explained in Section 3.4 is used to extract the linear dependency of the features in this section. All the aforementioned features for the lateral velocity estimation are examined. The reference data used for the correlation coefficient test is explained in Section 6.1.

After preliminary tests with the above mentioned set of features, an issue emerged. The issue was because of the fact that the noise to sensor measurement ratio for lateral velocity sensor, was considerable. One of the suggested solutions was using the history of the input features as separate features. These historical features are related to the last few consecutive estimations. As mentioned before, the estimation is supposed to be performed on $40Hz$ frequency. For history features, the last four steps of yaw rate $r_{-1}, r_{-2}, r_{-3}, r_{-4}$ are considered. Another set of history features that were used for this purpose were the last four steps of lateral acceleration $a_{y_{-1}}, a_{y_{-2}}, a_{y_{-3}}, a_{y_{-4}}$.

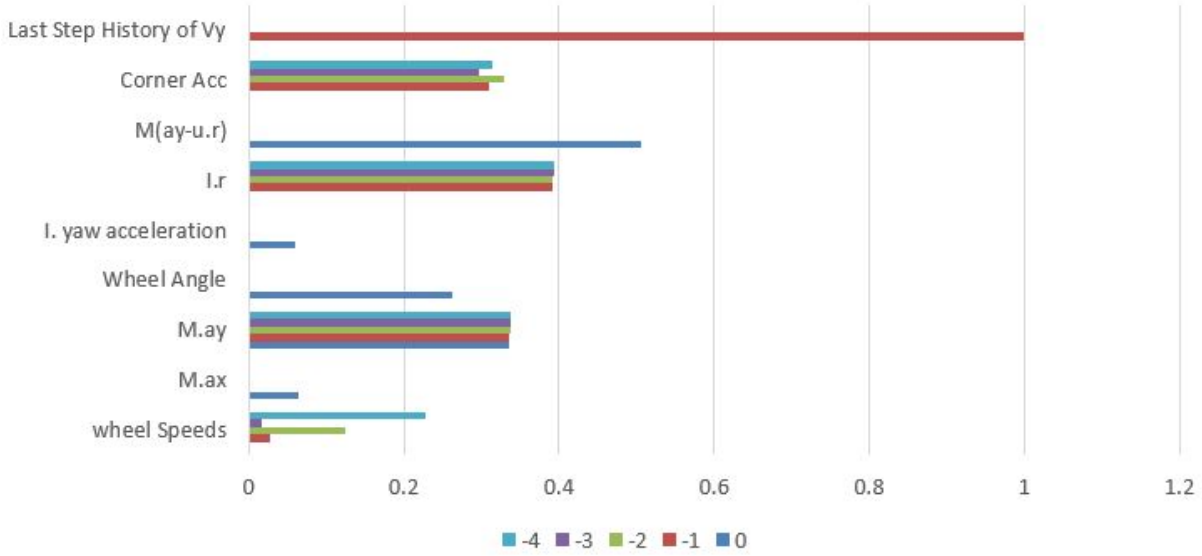

Figure 4.4: Pearson correlation coefficients for lateral velocity estimation features.

Another feature that can be considered along with other features to mitigate the error spikes happening in estimation is the last step of lateral velocity estimation. Since, this feature is calculated based on a simple data-based ML method, it cannot cause severe divergence. It will only help the estimator to maintain the range of output and prevents sudden changes in results, hence, it addresses the error spikes.

In Figure 4.4 the correlation between the possible input features and the output feature, lateral velocity, can be seen. Four vertical bars labeled as Ma_y and Ir, as well as the last step history of v_y are the historical features. Other bars are showing the correlations of four corner accelerations, $M\dot{v}$, $I_z\ddot{r}$, wheel angle δ_f, wheel speeds ω_i, and two other terms related to longitudinal and lateral accelerations Ma_x and Ma_y.

4.2.4 Final Features

The final set of features for lateral velocity estimation are shown in Figure 4.5. The main features from the dynamic equations are selected. Also, for the history input feature, the last step of lateral velocity is selected. All the features are normalized to make the estimation vehicle independent. For example, instead of using the lateral acceleration a_y, $m \times a_y$ is considered. Incorporating the vehicle parameters in features makes the data

universal and applicable for estimation of other vehicles as well. So, the reference data can be made by combining data from different vehicles, and it is applicable for velocity estimation of any vehicle. It is assumed the nominal values for all the vehicle parameters, e.g. $R_{e,i}, a, b, T_w, SR, m$, are available.

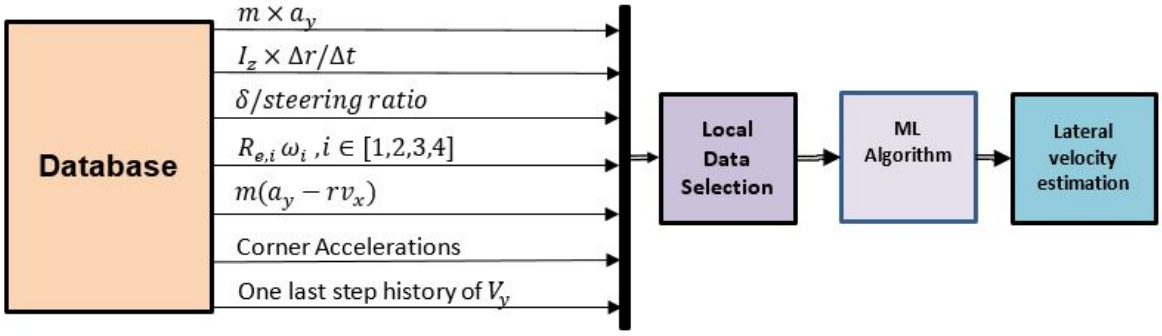

Figure 4.5: Overall estimation procedure for V_y.

4.3 Summary

In this chapter, estimation features were selected. For each of longitudinal and lateral estimations, a simple model was considered. The model only nominates the most effective factors in mode-based estimation. Using the model equations, the features were selected in a normalized way to be applicable to any vehicle. Feature normalization allows the estimation method to be applied to all the vehicles without needing to be modified or changed. After normalization, a correlation test was performed to determine the linear dependency between each input feature and the output feature. Pearson correlation coefficient was utilized for this purpose. At the end, the most effective features, considering both the physical interpretation and the correlation results, were selected for each estimation.

Chapter 5

Longitudinal Velocity Estimation

In this chapter, the longitudinal velocity estimation using kernel-based methods are studied. Two kernel-based methods considered for this estimation are Nadaraya-Watson and Gaussian Process Regression methods. These two methods are explained in Sections 3.1 and 3.2.

Data used for executing tests in this chapter are extracted from a database gathered in the MVS lab. The data in this database is related to other former researches. It is consists of data from experimental vehicle tests in different conditions, as well as simulation tests in various road conditions and driving scenarios. A total number of over 500 tests are included in this database related to six different vehicles. Different maneuvers are included in this database, such as longitudinal maneuvers, double lane change, single lane change, acceleration in turn, slalom tests, sinusoidal steering inputs, and step steers. For different road conditions, dry road, wet road, and slippery road conditions can be named.

In this chapter, the reference data selection procedure is explained, and after that, some test results are presented.

5.1 Reference Data Selection

For vehicle longitudinal velocity estimation, reference data need to be selected. Since the estimation methods used in this study rely on the reference data in the neighborhood, it is important to have reference data available in the vicinity of the test points. Hence, the maneuvers included in the reference data are similar to those of the test maneuvers. However, none of the test maneuvers are included in the reference data.

All the test and reference data considered for this study are from maneuvers on roads with zero grade angle and zero bank angles. As mentioned before, the features need to be dimensionally normalized to comply with the universal estimation scheme. The vehicle parameters used in this estimation are tire effective radius R_e and $SteerRatio$. In Table 5.1 the vehicle names and parameters are listed. The experimental tests included in the data set, are performed by the Chevrolet Equinox vehicle. All other vehicles in the database are related to the simulation tests performed in CarSim.

Table 5.1: Vehicle parameters used for longitudinal velocity estimation.

		Parameters	
		R_e	$SteerRatio$
Vehicles	E-class	0.4	20.05
	D-class	0.33	17.41
	Chevrolet Equinox	0.34	17.40
	C-class	0.33	17.61
	Cadillac Escalade	0.4	17.77
	C-class Hatchback	0.33	17.97

In some tests, which are included in the test set or reference data set, the wheel speed may exceed the reasonable amounts for this variable. This is because of the fact that the wheel is spinning freely. To prevent this incident to generate outlier data, the wheel speeds over 150 kph are changed to 150 kph. This also will help the normalization algorithm, stated in Equation (3.12), to find a more reasonable interval for wheel speeds.

The reference data are reduced and the local data are selected using the algorithms explained in Sections 3.5, 3.6. The reduction algorithm will help to get rid of duplicated data. For example, since almost every maneuver starts and ends with the zero steering input, the data can be very dense and duplicated in regions near the zero steering wheel angle inputs. This algorithm will help the estimator to perform faster and avoid biased results.

For estimation of every test point, only a limited number of reference points around that is used. Two reasons can be named for this choice. First, the total number of all the reference points is very large. So, if all the reference points are considered for each test point estimation, the calculations can be time-consuming. A second reason for this is that the system is interpretable linearly in small intervals and since the kernel-based method is almost like an averaging method, the results are more accurate and reliable. Local reference data need to be selected using the methods mentioned in Section 3.6.

Based on the feature importance factor, a unit ellipsoid is used to select the reference data in the vicinity of each test point. For those test points located in overcrowded or sparse regions of the reference data set, a KNN algorithm is used to select the maximum or minimum number of reference points needed for the estimation.

5.2 Test Results

In this section, a few test results are presented based on the selected features for longitudinal velocity estimation. Every test is evaluated with two kernel-based methods NW and GPR. Also, none of the test maneuvers are included in the reference set.

The longitudinal velocity estimation results are compared with the ground truth for this state in the following tests. If the test is a simulation test, the ground truth is directly read from the simulation software. And if the test is an experimental test, the ground truth can be determined by either a GPS sensor or model-based observer. Based on the test situation, the reliable measurement is used to be compared with the estimated value.

5.2.1 Longitudinal Velocity Test 1

In this test, a simulation maneuver performed by a D-class vehicle model is used for longitudinal velocity estimation. This is a continuous test performed on a dry road, which consists of 800 test points. In this maneuver, wide range of longitudinal velocity is reached and multiple accelerations and decelerations states are observed.

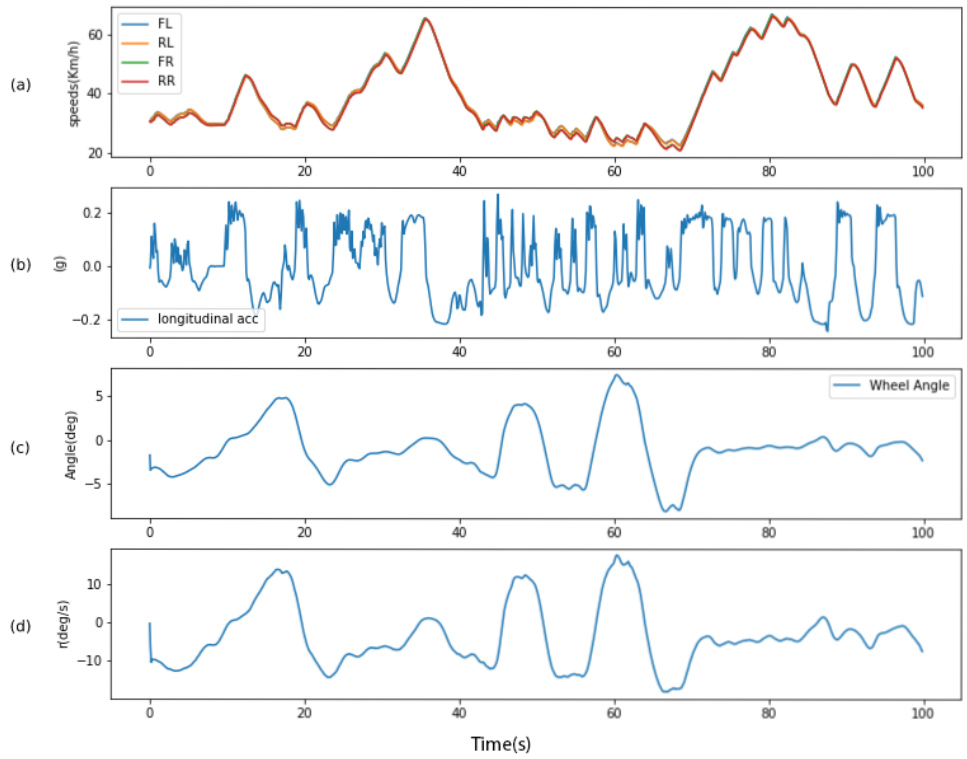

Figure 5.1: Sensor measurements for longitudinal velocity test 1.
a) Four wheel speeds. b) Longitudinal acceleration. c) Front wheel angle. d) Yaw rate.

In Figure 5.1 sensor measurements for this test are shown. All of these sensor measurements are used for estimation. Based on every feature importance, the local reference points are selected and used for this estimation.

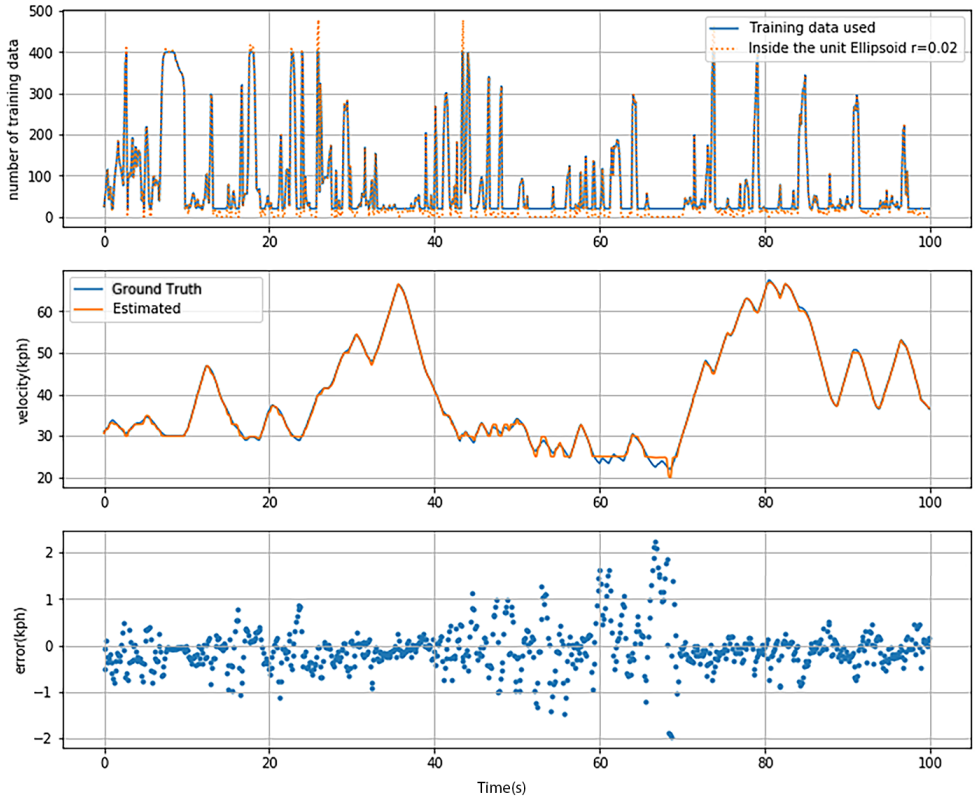

Figure 5.2: Longitudinal velocity estimation using NW method for test 1.
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors.

The result of longitudinal velocity estimation with the NW method is shown in Figure 5.2. As shown, Figure 5.2 is consist of 3 plots. The plot at the top is showing the number of available reference data around each test point which was inside the unit ellipsoid (shown with orange dashed line). This number can be less than 500 for each of the test points. On the one hand, the number of reference points encompassed by the ellipsoid is zero for some test points. This indicates that no reference data was selected within the search radius. On the other hand, some of the test points are located in crowded areas and there are plenty of reference data encircled by the unit ellipsoid. As a result, the KNN method is used to select the 20 or 400 nearest reference points to be utilized in the estimation algorithm (shown as the blue line).

The middle and bottom plots in Figure 5.2 show the test result and the errors. The middle plot shows a comparison between the ground truth, which was taken from the CarSim, and the estimation result. As the plots show, the estimation is following the ground truth closely.

This estimation was performed on 800 test data and it took 4.13 seconds. The total root mean square error is 0.4915 kph and the r-squared value is 0.9984.

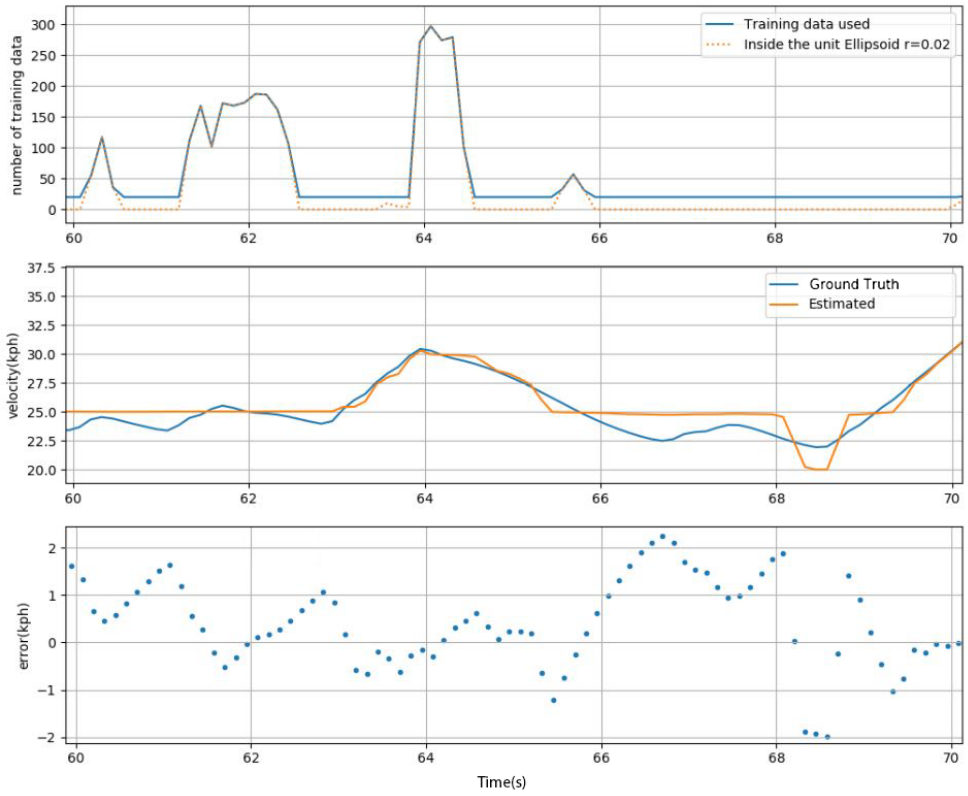

Figure 5.3: The 60-70 seconds interval of the estimation for longitudinal velocity test 1.

The largest errors of this estimation are happening in 60-70 second interval. In Figure 5.3, this interval is shown in larger scale. As it can be seen, the number of reference points around the test points with the greatest errors is very low. Particularly, test points that happened between seconds 66 and 70, are located in sparse regions of the reference data set. Lack of sufficient reference points in the vicinity of these test points can be the main reason for errors happening in the estimation.

Another kernel-based method considered for this estimation is the GPR method. For the GPR method, the estimation is performed for each test point using the nearest 30 reference points.

The result of this estimation is presented in Figure 5.4. As it is seen in this figure, two plots are shown regarding this estimation. The top plot is showing the result of GPR in comparison with the ground truth. The plot at the bottom is showing the error of this estimation.

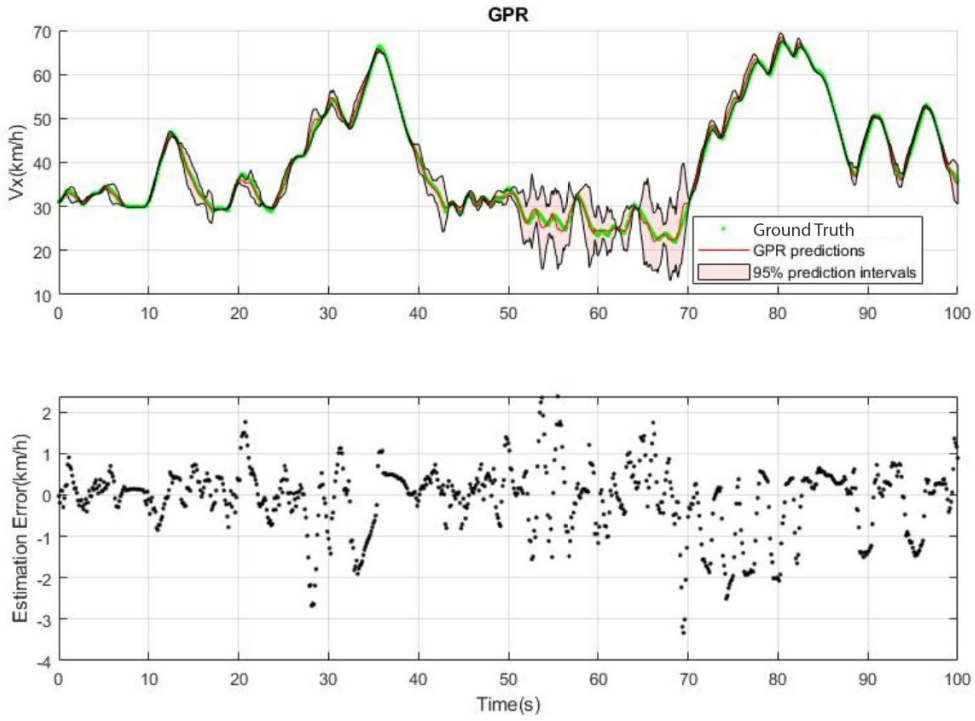

Figure 5.4: Longitudinal velocity estimation for test 1 using the GPR method.

In addition to the estimation values for the longitudinal velocity, the GPR method provides a confidence interval for each test point estimation. This confidence interval expands or shrinks based on the variance of the estimation, which is directly related to the reference data provided for the GPR model.

Hence, the 95% confidence interval is wider between 50 and 70 seconds due to the insufficient reference points around the test points. This result was also observed in the estimation by the NW method. The overall run-time for the 800 test points estimation

is 11.77 seconds. The root mean squared error is 0.8242 kph, and the r-squared value is equal to 0.9955, for this estimation.

Table 5.2: Estimation statistics for longitudinal velocity Test 1.

	Test Points	Total Run-Time(s)	Individual Average Time(ms)	RMSE(kph)	r^2
NW method	800	4.13	5.2	0.491	0.9984
GPR method	800	11.77	14.7	0.824	0.9955

The information related to test 1 estimation is presented in Table 5.2. As it can be seen, the accuracy of the NW method is better because of lower RMSE and higher r^2 values. Also, the total run-time is shorter for the NW method. Despite the fact that for each estimation by the GPR method only 30 nearest reference points are considered, this method is more time-consuming. The error patterns in both methods show that these two methods are highly dependent on the availability of reference points in the neighborhood.

5.2.2 Longitudinal Velocity Test 2

The second test considered for longitudinal velocity estimation is a double lane change test performed with a constant longitudinal velocity. This test is a simulation test with the Cadillac Escalade vehicle model in CarSim. The friction coefficient is set to $\mu = 0.6$ to simulate a wet road condition. The test is consists of 505 test points. The sensory data needed for this test is presented in Figure 5.5.

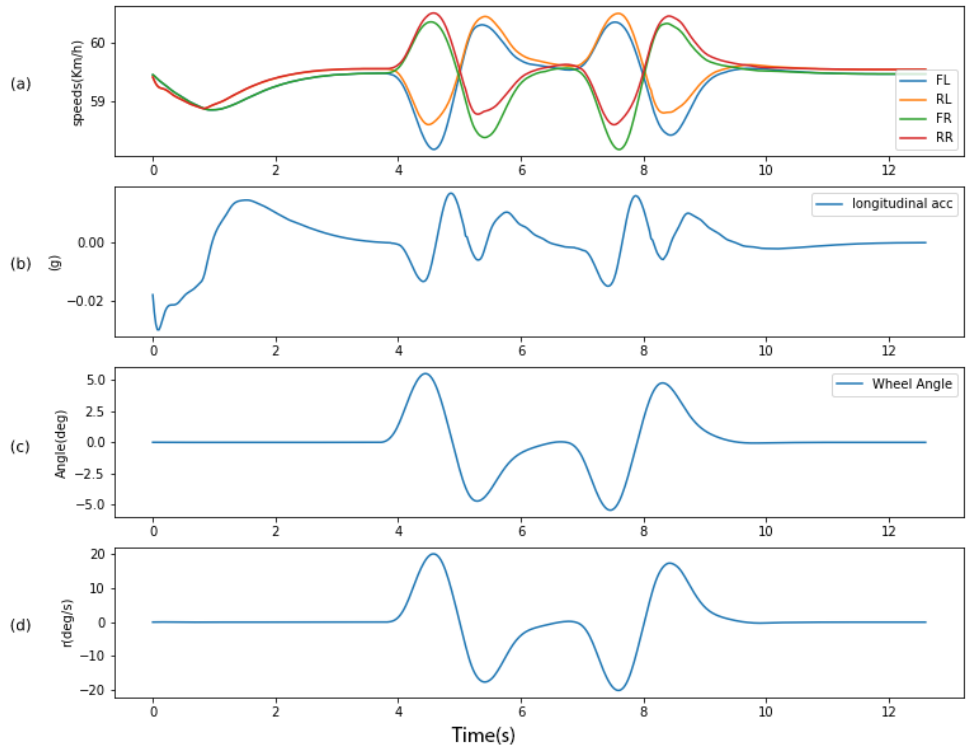

Figure 5.5: Sensor measurements for the longitudinal velocity test 2.
a) Four wheel speeds. b) Longitudinal acceleration. c) Front wheel angle. d) Yaw rate.

The estimation results using the local data with NW and GPR methods are presented in Figures 5.6 and 5.7, respectively. As the number of reference points around each test point is showing, the test is happening in a well-observed area. Hence, there are no problems related to finding the nearest reference points around the test points.

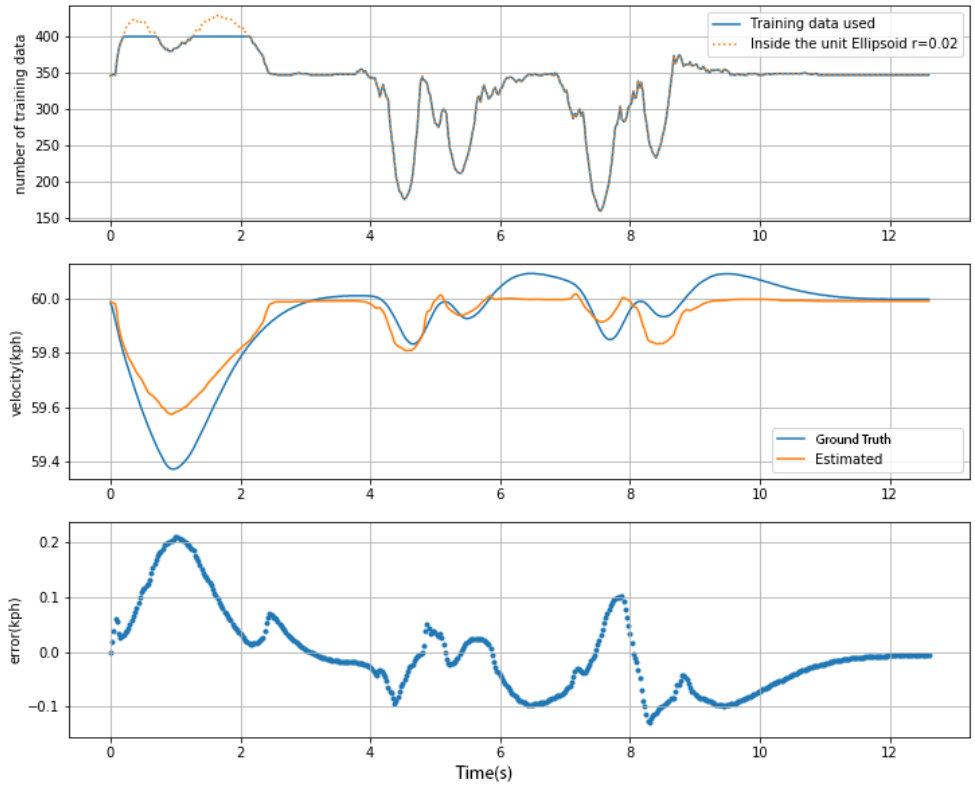

Figure 5.6: Longitudinal velocity estimation using NW method for test 2.
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors.

As the results are showing, the algorithms are able to estimate the velocity with very small errors. Although the only noticeable error has occurred in GPR estimation around the fifth second of the maneuver, the error is still less than 1% of the signal value. Therefore, the accuracy of the estimations is very good.

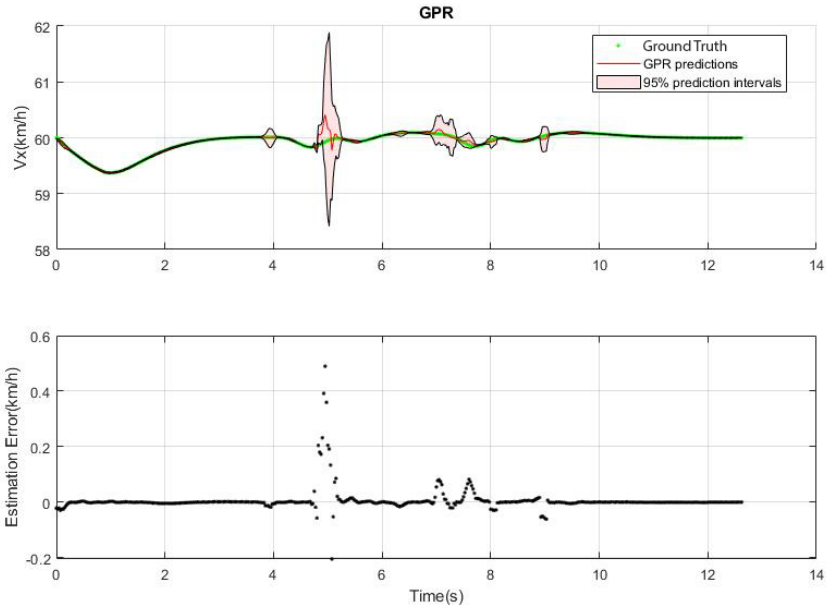

Figure 5.7: Longitudinal velocity estimation for test 2 using the GPR method.

Table 5.3: Estimation statistics for longitudinal velocity Test 2.

	Test Points	Total Run-Time(s)	Individual Average Time(ms)	RMSE(kph)	r^2
NW method	505	7.11	14.1	0.074	0.9109
GPR method	505	9.37	18.5	0.043	0.9435

As it is shown in Table 5.3, both of the estimations have very small amounts of root mean square errors. In terms of accuracy and errors, the GPR method is showing a better result. However, in therm of the estimation time, the NW method has performed a faster estimation.

It is worth mentioning that the reference set used for this estimation is the same reference set used for longitudinal velocity test 1. Hence, did not need to be changed for these two tests that were performed with different vehicles.

5.2.3 Longitudinal Velocity Test 3

In this maneuver, the Cadillac Escalade vehicle CarSim model is used. This maneuver includes acceleration and brake with steering from 12 to 17 seconds. Like the last test, this test is also done with a coefficient of $\mu = 0.6$ to simulate a wet road condition. In contrast with the previous test, the speed is not constant in this test and longitudinal acceleration is happening. The sensory measurements for this test are depicted in Figure 5.8.

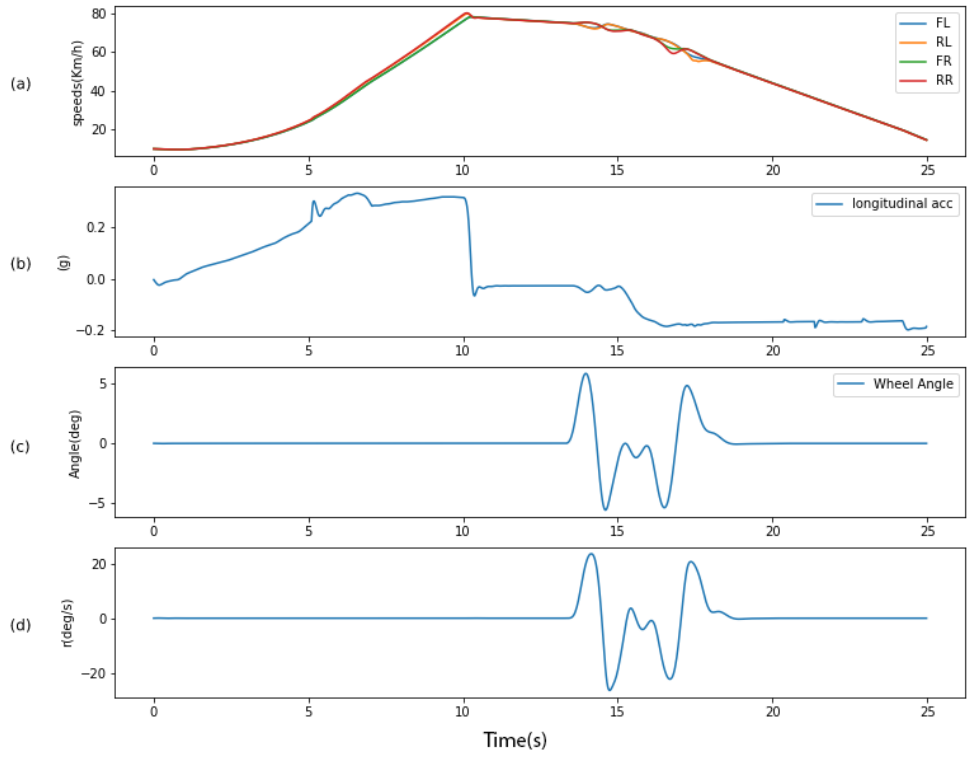

Figure 5.8: Sensor measurements for longitudinal velocity test 3.
a) Four wheel speeds. b) Longitudinal acceleration. c) Front wheel angle. d) Yaw rate.

The unit ellipsoid radius for the NW estimator is set to $r_0 = 0.01$, to obtain the data in close vicinity. This test is comprised of 1000 test points and the algorithm took 5.85 seconds to estimate the outputs for the NW method. It can be said that the algorithm has taken 5.85ms on average to estimate each test point. The root mean square error for this estimation is 1.56 kph and the r-squared value is 0.9963.

As it is shown in Figure 5.9, the estimation error is very small except for the error spikes around $t = 15$s. As it can be seen, there is a sharp drop in the number of reference points enclosed within the unit ellipsoid just before the first major errors appear. In other words, the test data enters a sparse region of the reference data set, right after getting out of a crowded one. Therefore, the lack of properly distributed reference data in that region results in some error spikes.

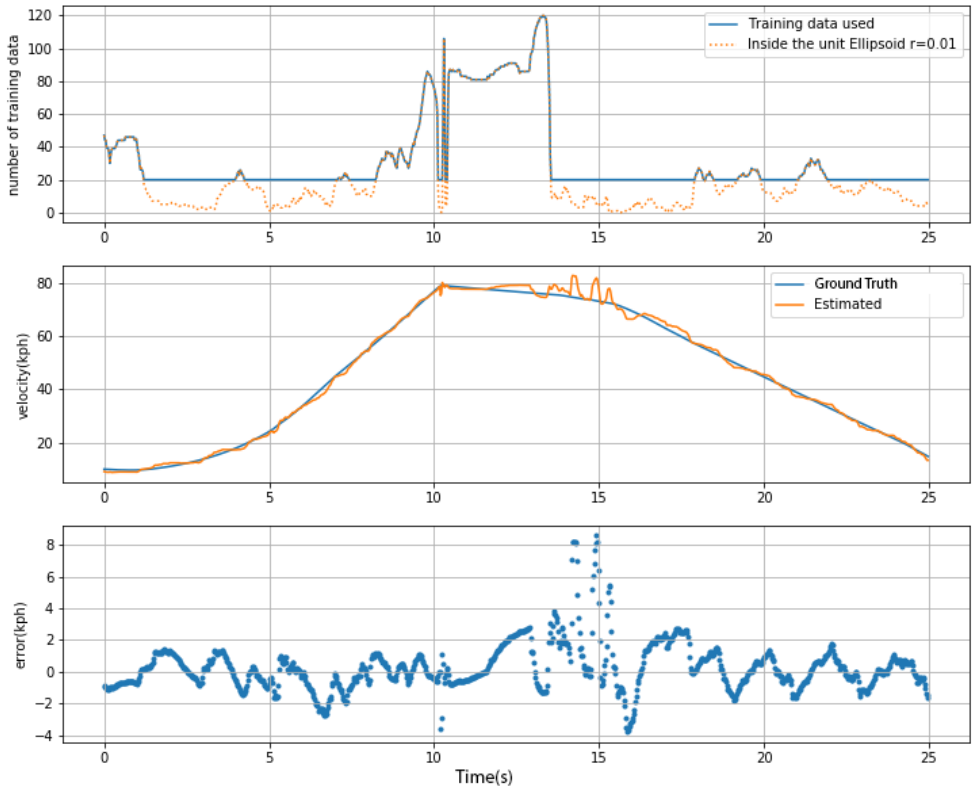

Figure 5.9: Longitudinal velocity estimation using NW method for test 3.
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors.

The estimation results using the local data with NW and GPR methods are presented in Figures 5.9 and 5.10, respectively. In a comparison of these two plots, the difference between the two estimators is noticeable. The GPR method trains a Gaussian Process model for each estimation. This model is trained based on the thirty nearest reference

points. Hence, the GPR model can perform more precisely in situations where NW may not provide perfect results. Considering that it is a weighted average algorithm, the NW may not be as accurate in situations where there is not enough relevant reference data available.

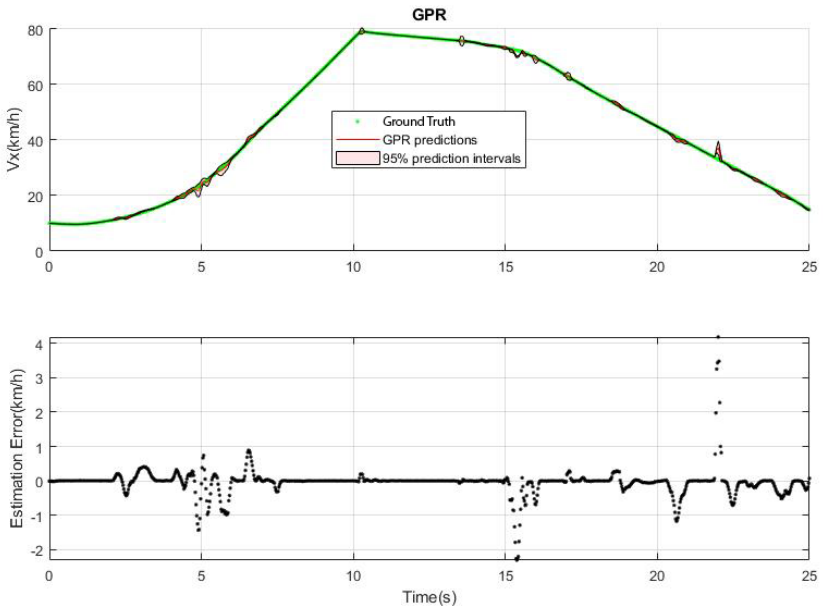

Figure 5.10: Longitudinal velocity estimation for test 3 using the GPR method.

Table 5.4: Estimation statistics for longitudinal velocity Test 3.

	Test Points	Total Run-Time(s)	Individual Average Time(ms)	RMSE(kph)	r^2
NW method	1000	5.85	5.8	1.562	0.9963
GPR method	1000	17.12	17.1	0.390	0.9997

In Table 5.4, the estimation numbers of both estimations are provided. The GPR method provides better RMSE and r^2 values. However, the NW algorithm is computationally more efficient.

5.3 Summary

In this chapter, longitudinal velocity estimation using local data is performed. Utilizing the results shown in Chapter 4, the proper features are built and used for the longitudinal velocity estimation. Based on the selected features, reference data is gathered. Local reference points are selected for each test point based on the distance and feature importance from the reduced reference data. Using the reference and test data provided, estimations are performed using NW and GPR algorithms.

Three longitudinal velocity estimations are presented in this chapter. These tests are simulation tests performed with two different vehicle models on dry or wet roads. They are double lane change with constant velocity, acceleration and deceleration in presence of steering input, and a general case with a large range of longitudinal velocity.

According to the provided results, both GPR and NW methods can estimate the longitudinal velocity with an error of les than 2kph for most of the estimation interval. To be more specific, both have root mean square errors less than 2 kph for all three tests presented in this chapter. Although in some cases, the GPR method is showing better performances, the NW method is more time-efficient. The largest error of the longitudinal velocity estimations is 8 kph, which is for the third estimation using the NW method. The largest error occurred in a region with insufficient reference data around the estimation region.

Both of the methods used in this study are data-driven methods. Since the local data is used for the estimations in this chapter, it is very important to have the proper reference data available in the region. Proper data will have an even distribution, and it won't be sparse or crowded. A proper set of reference data will lead to a more accurate and more precise estimation. Therefore, having healthy, rich reference data is very vital for this form of estimation.

Chapter 6

Lateral Velocity Estimation

In this chapter, lateral velocity estimation using kernel-based methods is studied. Similar to the method used in the previous chapter, kernel-based methods considered for this estimation are Nadaraya-Watson and Gaussian Process Regression methods. These two methods were explained in Sections 3.1 and 3.2.

Total raw data used for performing tests in this chapter are the same as the data used for Chapter 5. It is explained that the data set consists of test data related to various vehicles and maneuvers. Also, different road conditions are met in the data set. Tests included in the data are performed either with a high fidelity vehicle model or an actual vehicle. The raw data has the vehicle sensor measurements, and they are normalized to be used for making the reference data set for lateral velocity estimation.

This chapter explains the reference data selection procedure, and some test results are presented. Later, two healing systems are defined and implemented to richen the reference data set. Each healing procedure generates data with a high fidelity vehicle model for adding to the reference data and enriching the sparse regions. The performance analysis is studied at the end.

6.1 Reference Data Selection

For vehicle lateral velocity estimation, reference data need to be selected. The above-mentioned reference data are used as reference points for the weighting average process in each method. Since the estimation methods used in this study rely on the reference data in the neighborhood, it is important to have reference data available in the vicinity of the

test points. Hence, the maneuvers included in the reference data are similar to those of the test maneuvers. However, none of the test maneuvers are included in the reference data set. All the test and reference data considered for this study are from the maneuvers on the roads with zero grade and zero bank angles.

As mentioned before, the features need to be normalized for making the estimations applicable for any vehicle. The vehicle parameters used in this estimation are $Mass$, yaw moment of inertia I_z, tire effective radius R_e, front axle to CG distance a, rear axle to CG distance b, wheel track T_w, and $SteerRatio$. The vehicle parameters are presented in Table 6.1. All the vehicles in the database are related to the simulation tests performed in CarSim, except for the tests done by the Equinox vehicle.

Table 6.1: Vehicle parameters used for lateral velocity estimation.

		Parameters						
		$Mass(Kg)$	$I_z(Kg.m^2)$	R_e	$a(m)$	$b(m)$	$T_w(m)$	$SteerRatio$
Vehicles	E-class	1860	2687	0.4	1.18	1.77	1.57	20.05
	D-class	1530	2355	0.33	1.11	1.67	1.55	17.41
	Chevrolet Equinox	2270	4605	0.34	1.42	1.44	1.62	17.40
	C-class	1830	3234	0.33	1.40	1.65	1.60	17.61
	Cadillac Escalade	2444	4183	0.4	1.45	1.49	1.75	17.77
	C-class Hatchback	1413	1537	0.33	1.01	1.89	1.67	17.97

The procedure of the lateral velocity estimation in this chapter is similar to the explained steps in Section 5.1. The wheel speed features are corrected, and the reduction algorithm is utilized. Also, the local reference points for each estimation are selected using the introduced ellipsoid method.

6.2 Test Results

In this section, six maneuvers are used for lateral velocity estimation. Simulation and experimental tests performed by different vehicle models with different road conditions are used to test the estimation performance, using both NW and GPR methods.

The methods used for lateral velocity estimation depend on the availability of reference points in the estimation space. Hence, lack of proper reference points in the estimation region may decrease the estimation accuracy. For generating data and adding to the reference set, two healing methods are implemented and the results are presented for last four tests.

6.2.1 Lateral Velocity Test 1

A maneuver performed by a Cadillac Escalade CarSim model is considered. Sensor measurements used for this estimation are presented in Figure 6.1. This maneuver is a double lane change performed on a dry road. The longitudinal velocity is almost constant about $85k/h$. It should be noted that the reference data set used for this estimation is not changed nor modified for this vehicle.

Using the selected features, the estimation of lateral velocity with the NW method is performed. The result of this estimation is presented in Figure 6.2. The figure shows that the estimation is done within an acceptable error range.

Additionally, considerable errors are happening in the regions associated with a low number of reference points in the unit ellipsoid around the test point. For instance, the largest error, according to Figure 6.2, happens at 0.25 m/s and $t = 3.5$ s.

For further investigation, box plots and histograms related to different features are presented in Figures 6.3 and 6.4. These plots are related to the reference set selected for the aforementioned test point, which has the maximum error.

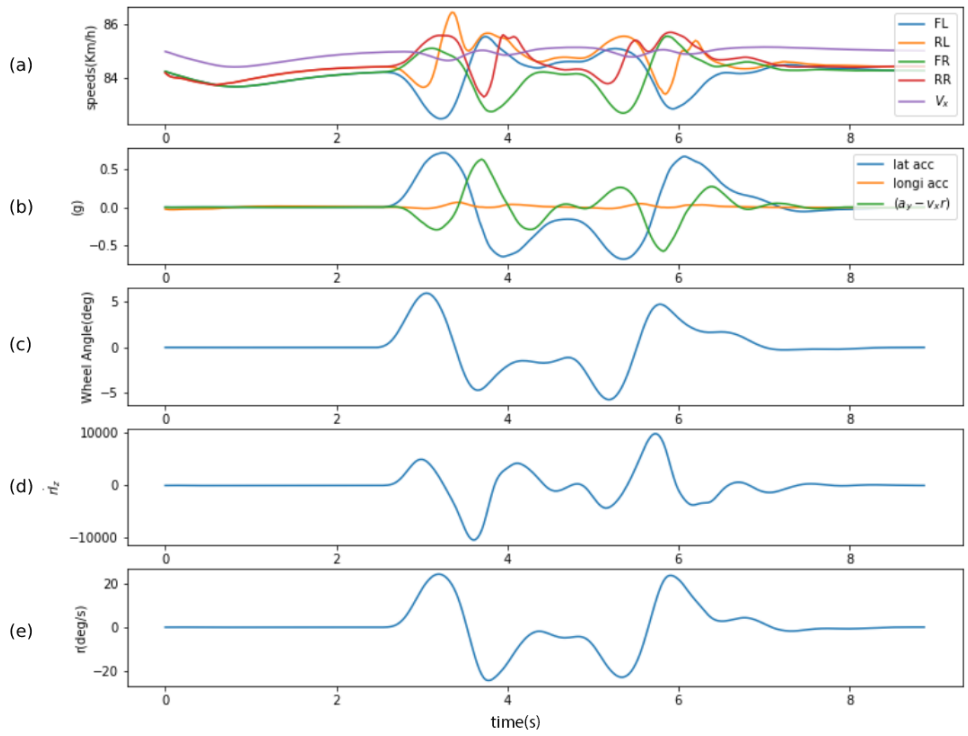

Figure 6.1: Sensor measurements for the lateral velocity test 1.
a) Four wheel speeds and longitudinal velocity. b) Longitudinal and lateral acceleration, lateral velocity derivative. c) Front wheel angle. d) Yaw moment of inertia times yaw acceleration rate. e) yaw rate.

Based on the box plots shown in Figure 6.3, the estimated lateral velocity is based on the selected reference points. The lateral velocities of all the reference points, which are the nearest 20 points to the test point, are between $-0.6m/s$ and $-1m/s$. Therefore, the estimated velocity for this point is located within the mentioned interval.

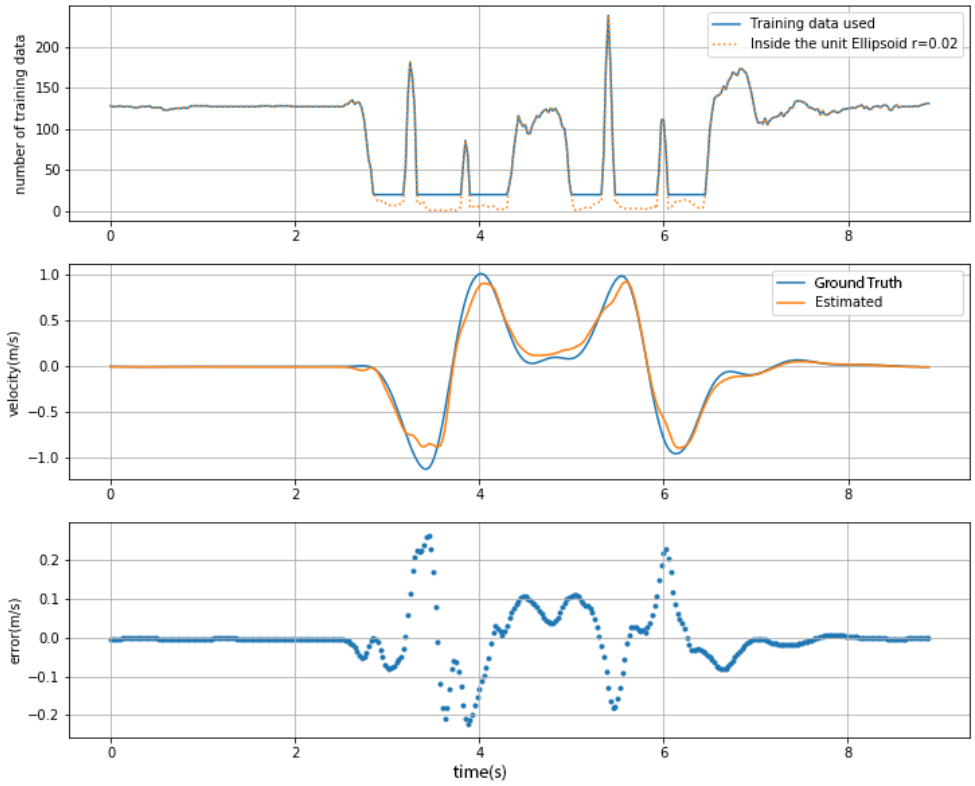

Figure 6.2: Lateral velocity estimation using NW method for test 1.
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors.

As it can be seen in the box plots related to the wheel angle, $\dot{r}I_z$, and all the corner accelerations in Figure 6.3, the point of interest is placed outside of the interquartile ranges. Hence, it is not possible for the estimator to estimate the test point with the selected reference points.

Another point that needs to be mentioned is that the wheel speeds of the reference points are located only at the ends of the selected range. This non-uniform distributions of reference points are depicted in Figure 6.4.

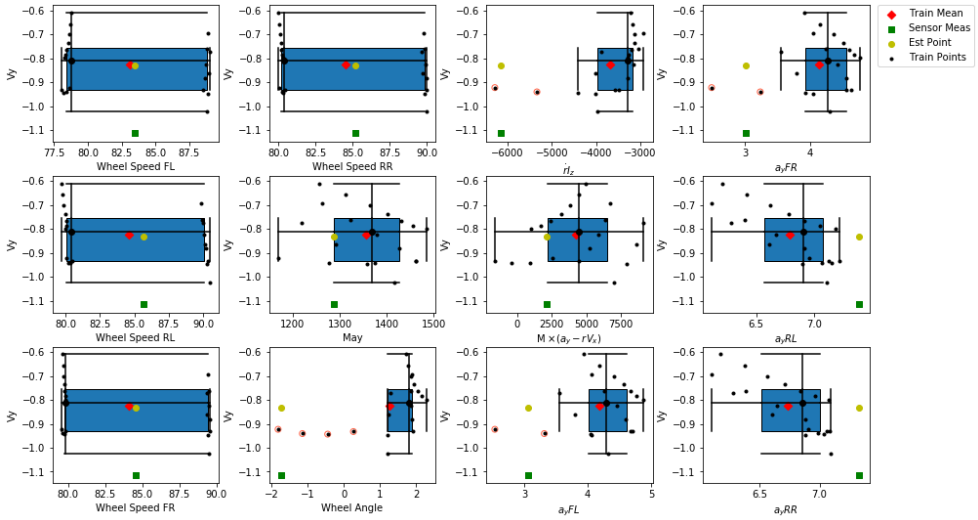

Figure 6.3: Box plots for different features of the selected reference points for a test point estimation with the largest error.

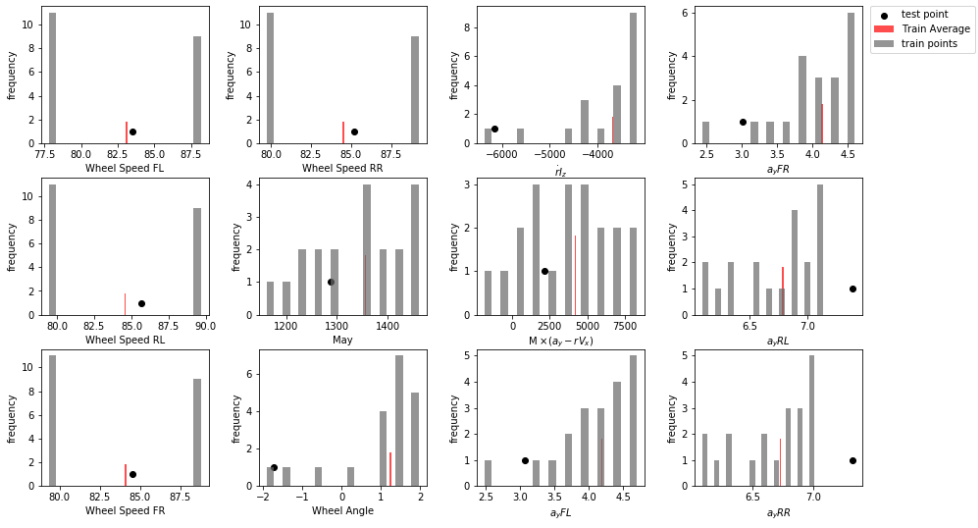

Figure 6.4: Histogram plots for different features of the selected reference points, for an estimation point with the largest error. Each interval is divided into 10 equal sub-intervals. Each bar is showing the number of reference points with the mentioned feature within the sub-interval.

The lateral velocity was also estimated using the GPR method. The result are shown in Figure 6.5. Nearly all the sensor measurements are located within the 95% confidence interval. This shows the great accuracy of the estimation using the GPR method. Also, it is noticeable that the confidence intervals get wider around $t = 3.5s$ and $t = 6s$. Also, the NW method showed the largest error in these points.

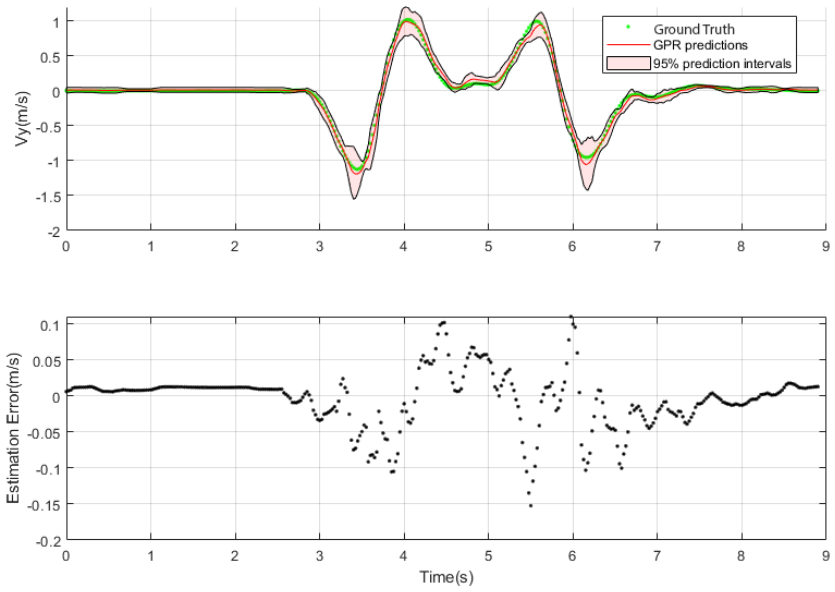

Figure 6.5: Lateral velocity estimation using GPR method for test 1.
Top: Ground truth, estimated values, and the 95% intervals. Bottom: Estimation errors.

As shown in Figure 6.5, the estimation errors are smaller, compared to those of NW estimation. The statistical properties of this test are shown in the 6.2. Although the estimation using the NW is faster, GPR shows a better accuracy in terms of RMSE and r^2.

Table 6.2: Estimation statistics for lateral velocity Test 1.

	Test Points	Total Run-Time(s)	Individual Average Time(ms)	RMSE (m/s)	r^2
NW method	356	3.1	8.7	0.072	0.9730
GPR method	356	4.9	13.7	0.038	0.9915

Although the maneuver is performed with a considerable longitudinal velocity and lateral acceleration, the estimators shows a great performance. The only drawback is related to the lack of properly distributed local reference points.

6.2.2 Lateral Velocity Test 2

The second test for estimating lateral velocity is a step steer maneuver with a constant velocity of $70kph$ performed by an E-class vehicle in CarSim on a dry road. The vehicle parameters used for feature normalization are stated in Table 6.1. The sensor measurements used for this estimation are presented in Figure 6.6.

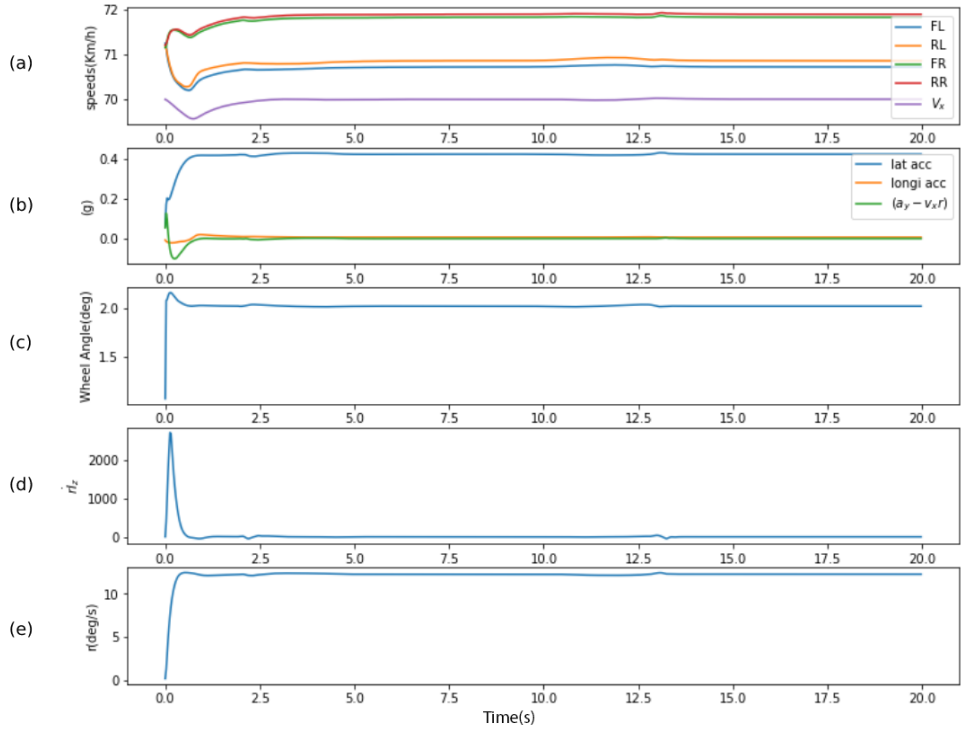

Figure 6.6: Sensor measurements for lateral velocity test 2.
a) Four wheel speeds and longitudinal velocity. b) Longitudinal and lateral accelerations, lateral velocity derivative. c) Front wheel angle. d) Yaw moment of inertia times yaw acceleration rate. e) yaw rate.

The reason behind having various values for tire speeds and longitudinal speed mea-

surements is the small error in tire effective radius as well as tire speed differences in the turn.

NW method estimates the lateral velocity of the maneuver based on the nearest reference data within the unit ellipsoid around each test point. As shown in Figure 6.7, almost all of the test points have more than 40 reference points around them for estimation. Hence, the test points are located in a well-trained area.

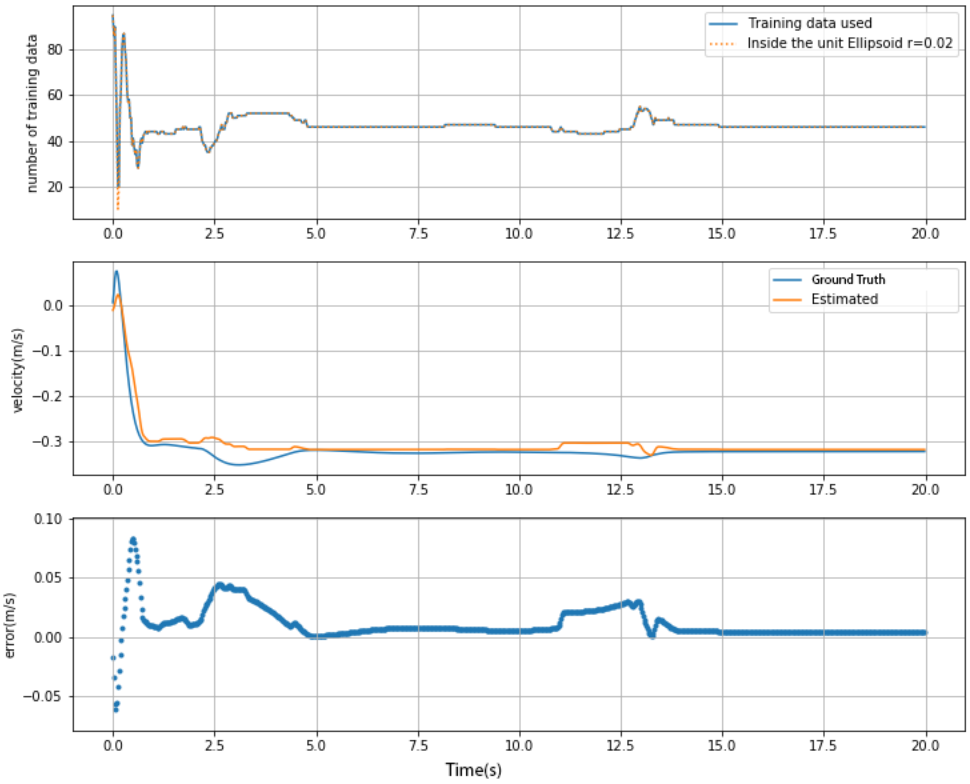

Figure 6.7: Lateral velocity estimation using NW method for test 2.
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors

The error values are less than $0.05m/s$ for most of the estimation. In some areas, e.g. $t = 2.5s$ or $t = 13s$, the error increases slightly as the number of reference points in vicinity increases. This mainly happens because the reference points are slightly skewed to other

directions. Therefore, the estimation output will see a slight increase in error. However, the overall estimation accuracy for this test is high.

The result of lateral velocity estimation using GPR method is presented in Figure 6.8. The estimated values are very close to the ground truth, which is shown as sensor measurements.

Some sudden changes are observed in this estimation around $t = 3s$ and $t = 13s$, which are the same regions mentioned for NW estimation.

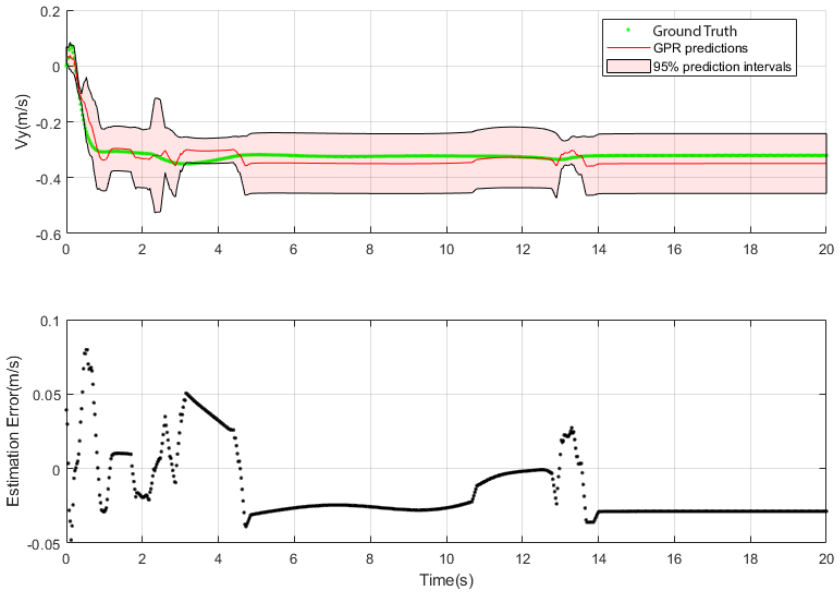

Figure 6.8: Lateral velocity estimation using GPR method for test 2.
Top: Ground truth, estimated values, and the 95% intervals. Bottom: Estimation errors.

The estimations statistical values are shown in Table 6.3. This test consists 800 test points, and the overall estimation time of the lateral velocity for this test using the NW method is 4.2s and this time for GPR estimator equals to 11.1s.

Although the estimations are performed with relatively good accuracy, the r-squared values show low values. This is because of the fact that most of the test points have the same level of lateral velocity. however, the RMSE for two estimations show their good accuracy. The lateral velocity error is less than 0.05 m/s for the most of the maneuver.

Table 6.3: Estimation statistics for lateral velocity Test 2.

	Test Points	Total Run-Time(s)	Individual Average Time(ms)	RMSE (m/s)	r^2
NW method	800	4.2	5.3	0.018	0.9146
GPR method	800	11.1	13.9	0.027	0.8161

6.3 Lateral Velocity Estimation Using Healed Data

The methods used for velocity estimation are data-driven methods, and they rely on the proper reference data in the locality of test points. Therefore, the proposed estimation methods are not able to provide accurate results in absence of proper local reference data.

In addition, some of the tests in the test set are from experimental tests. This means that data features need to be constructed using sensor measurements mounted on the vehicle. The sensor measurements can be noisy. To mitigate the noise effects on test input features, local regions of reference data need to be well-observed.

In this section, a healing method is introduced. The effect of this healing method on the estimation methods are studied.

6.3.1 Healing Using Simulation

The healing process considered for this section depends on the test data that are going to be estimated. The general idea of this healing method is to regenerate the test data using a high fidelity model and add the generated data to the reference data. This action will make the reference data richer around the test points of interest.

The high fidelity model of the vehicle that is considered to regenerate the test is a CarSim vehicle model. It can be selected and customized to be similar to the test vehicle. The parameters that need to be modified in the test vehicle are presented in Table 6.1.

Using the recorded sensor measurements in the test set, the inputs to the high fidelity model can be generated. The longitudinal velocity is used as a reference for controlling wheel torques. Also, the steering wheel angle can be utilized as an input to the model, directly.

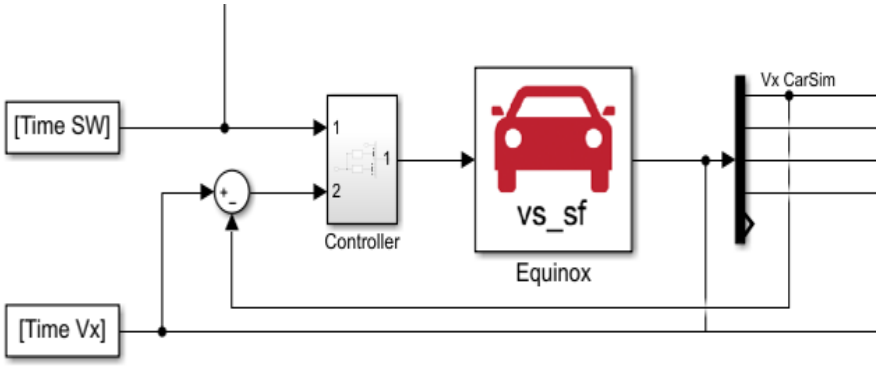

Figure 6.9: Data generation set-up using the test data.

Figure 6.9 shows the data generation scheme used in Simulink. Since the tests were originally performed with a Chevy Equinox vehicle, the high fidelity model is also set to an Equinox model from CarSim. The steering wheel angle and longitudinal velocity are read from the test file. All the vehicle parameters should be the same as the real vehicle parameters. The steering wheel angle and longitudinal velocity are read from the test file. Using PID controllers, tire torques are generated and fed to the high fidelity simulation model.

In Figure 6.10, two tests are presented. Both tests are performed by Chevrolet Equinox vehicle originally and regenerated by mentioned approach.

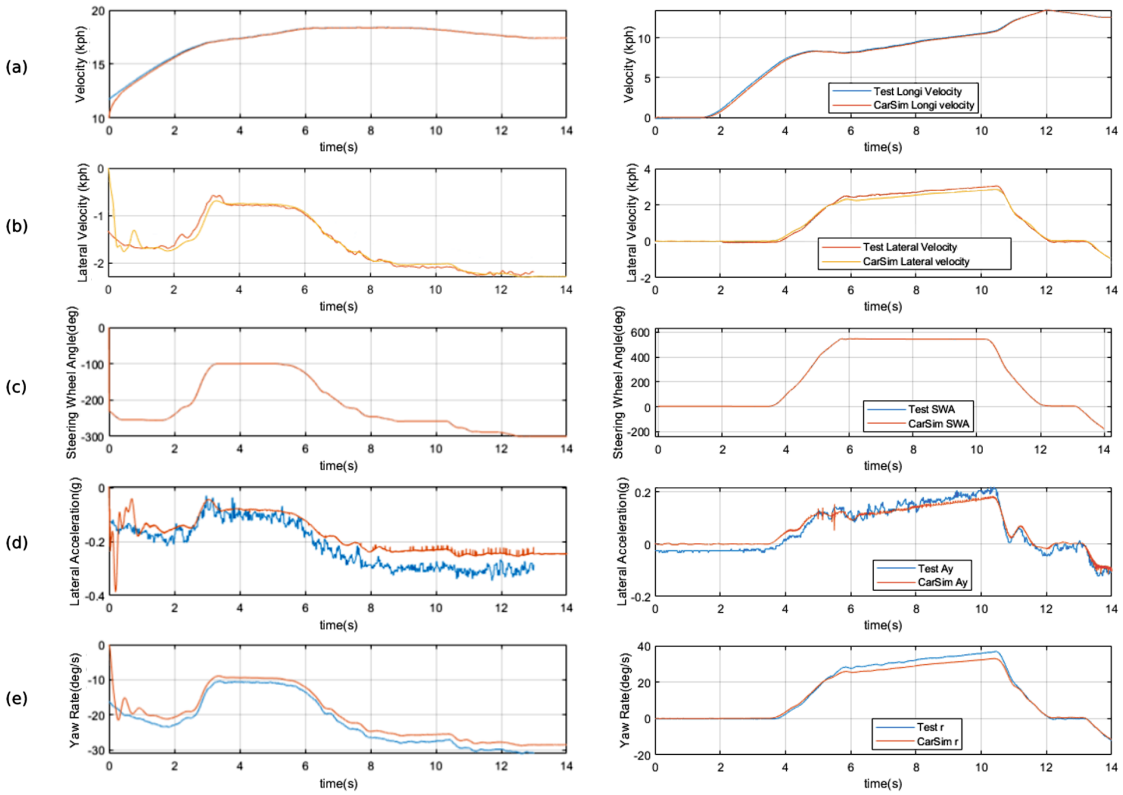

Figure 6.10: Two Regenerated Tests
a) The longitudinal velocity of original and regenerated tests. b) Lateral velocity of original and regenerated tests. c) Steering wheel angle. d) Lateral accelerations of original and regenerated tests. e) Yaw rate of original and regenerated tests

As it is also shown in Figure 6.10, the regenerated feature are not exactly the same as the experimental sensor measurements. For example, the lateral velocities, shown in second-row plots from the top, do not match the ground truth completely. The same issue arises for the lateral acceleration and yaw rate measurements, shown in 4^{th} and 5^{th} rows.

This difference will not cause any harm to the healing process of the estimation methods. Since the estimator is using the reference data from the vicinity of test points, the goal of this healing procedure is to make the test regions richer in reference data. Hence, the region will get richer, even if the regenerated data is not completely the same as the test

data.

Another effective factor in these simulations is the road condition, which may not be exactly known for each test. Since the road condition is not recorded within the test data, very little information can be extracted from the test data about this matter. To cover all the possible scenarios, all the tests are regenerated for various road friction coefficients, i.e. $\mu = [0.3, 0.5, 0.7, 0.9]$. This can represent dry, wet, or snow conditions.

All the regenerated test data are added to the reference data, after being reduced using the data reduction algorithm. This will prevent adding duplicated data to the reference data. It also helps to improve the reference data density in sparse regions.

In the following subsections, two tests are performed and the accuracy of estimation methods are studied after healing the reference data set.

6.3.2 Lateral Velocity Test 3 With Healing

The third test used for lateral velocity estimation is studied in this section. A Chevrolet Equinox vehicle is used for this test. All the test data used for this test are measured using the sensors on the vehicle. The lateral and longitudinal velocities are measured using a GPS sensor installed on the vehicle. Additionally, the IMU sensor measurements provide the rest of the features. Also, the effect of the healing method presented in Section 6.3.1 is studied for this test.

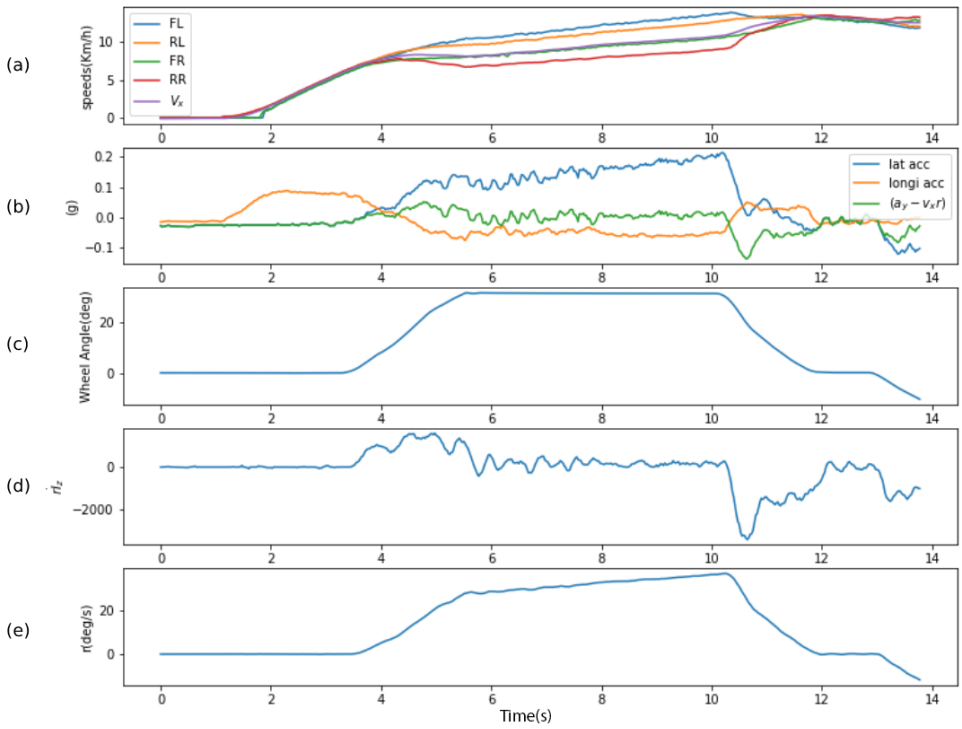

Figure 6.11: Sensor measurements for the lateral velocity test 3.
a) Four wheel speeds and longitudinal velocity. b) Longitudinal and lateral accelerations, lateral velocity derivative. c) Front wheel angle. d) Yaw moment of inertia times yaw acceleration rate. e) yaw rate.

As shown in Figure 6.11, this test is performed with low longitudinal velocity. In addition, an intense step steering input is applied. Although the longitudinal speed is very low, because of intense steering, the lateral velocity is considerable. Hence, because of the

large side slip angle, the vehicle can be near edge of instability. This test is performed on a wet road condition.

The lateral velocity is estimated using the NW and GPR algorithms with two reference data sets. First, the lateral velocity is estimated using the original reference data. This is the same data set used for other tests of this study. Then, the reference data set is healed using the healing method presented in Section 6.3.1.

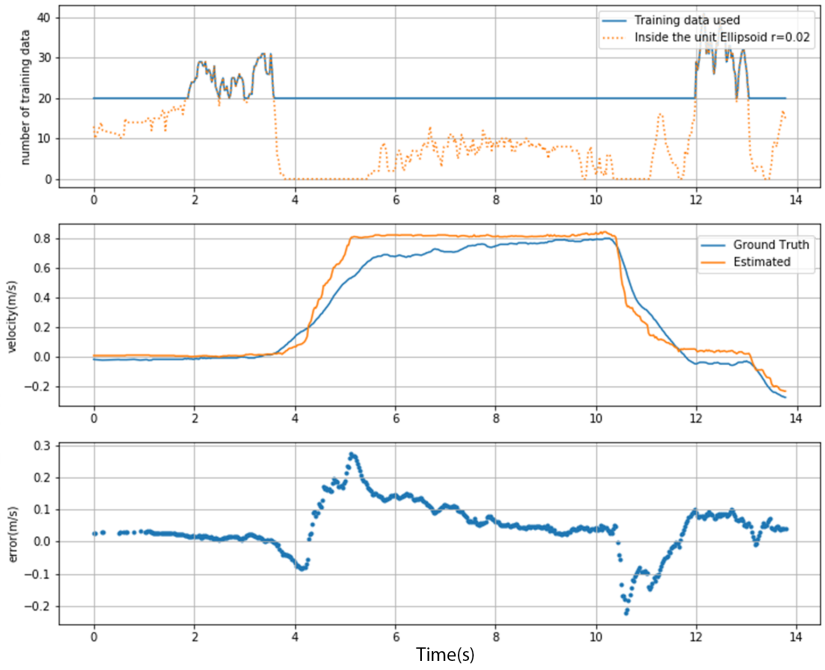

Figure 6.12: Lateral velocity estimation using NW method for test 3, using the original reference data.
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors.

As shown in Figure 6.12, the estimation encountered lack of reference points for most of the test points. All the test points located within $t = 4s$ to $t = 12s$ have less than 20 reference points available around them, inside the unit ellipsoid. In these regions that 20 nearest reference points are used to estimate the lateral velocity, the estimation show considerable errors. In this case, if the point of interest is located near another relatively dense region, the estimation will seem to have bias error, like what is happening between

$t = 5s$ to $t = 10s$.

The result of the estimation using the NW method with healed reference data set is presented in Figure 6.13. The number of reference points within unit ellipsoids related to test points located between $t = 5s$ and $t = 10s$ is increased. However, not enough reference points have been generated in some regions. Harsh conditions of the maneuver and differences between simulation and experimental tests can be two reasons for this issue.

Although the number of reference points are still less than 20 in some regions, adding the generated reference points has made a significant improvement in the estimation results.

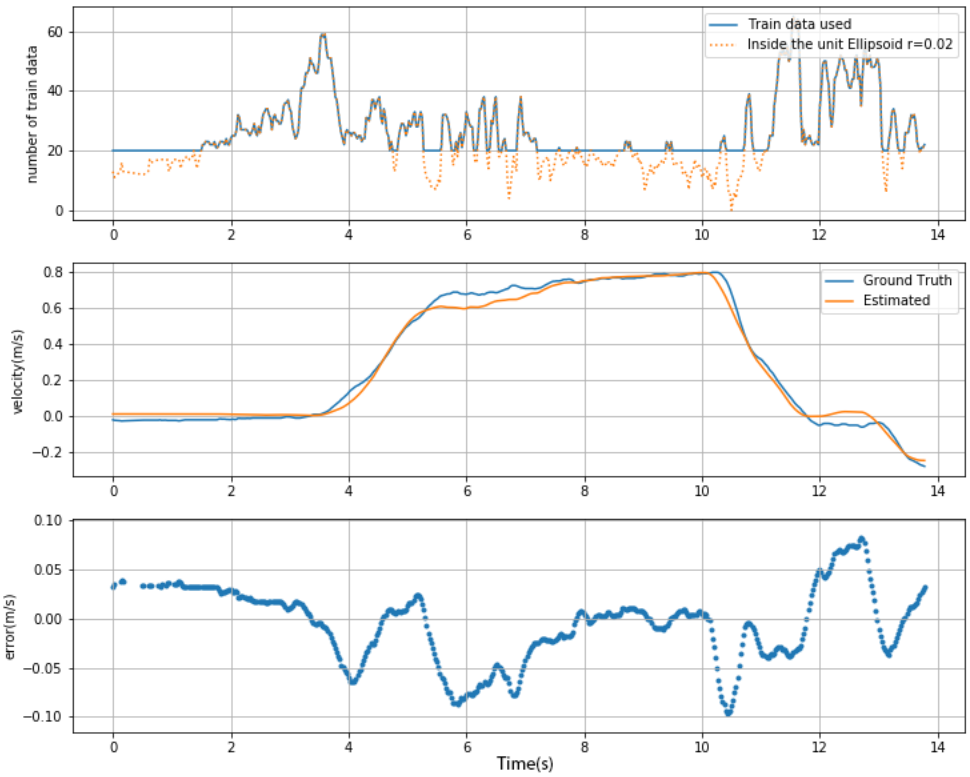

Figure 6.13: Lateral velocity estimation using NW method for test 3, using the healed reference data.
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors.

The estimations of lateral velocity using GPR method are presented in Figures 6.15

and 6.14. The results resemble those of the NW method. With considering the additional reference data for healing the reference data set, performance is improved and bias error is decreased. Due to restrictions applied for the GPR method, the maximum error happening in this method is greater than the maximum error in the NW method. Considering the fact that most of the test points were located in sparse regions, the nearest reference points may not reflect the test point situation accurately. Since the GPR method is more complicated than the NW method, only a limited number of nearest reference points are considered. The reason behind this is that the estimation can be very time-consuming by itself and such constraints make the estimation more suitable for real-time applications.

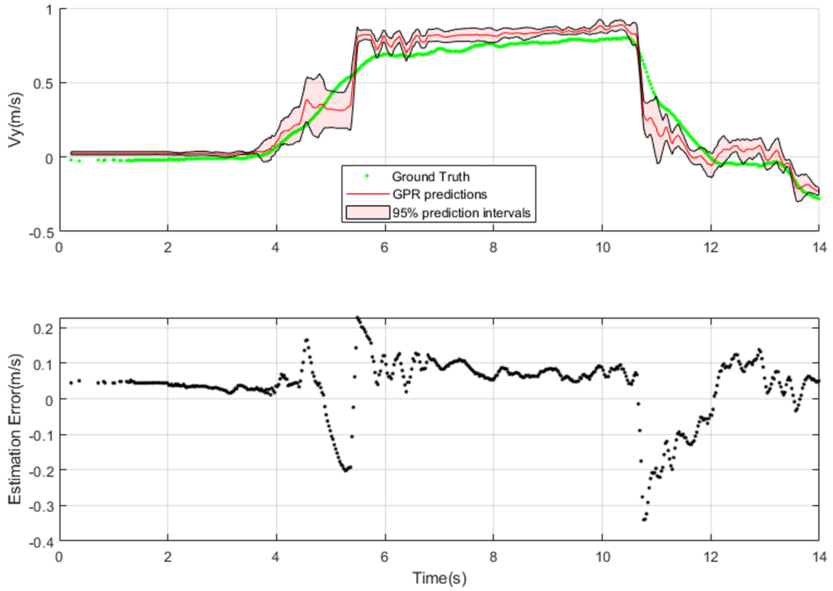

Figure 6.14: Lateral velocity estimation using GPR method for test 3, using the original reference data.
Top: Ground truth, estimated values, and the 95% intervals. Bottom: Estimation errors..

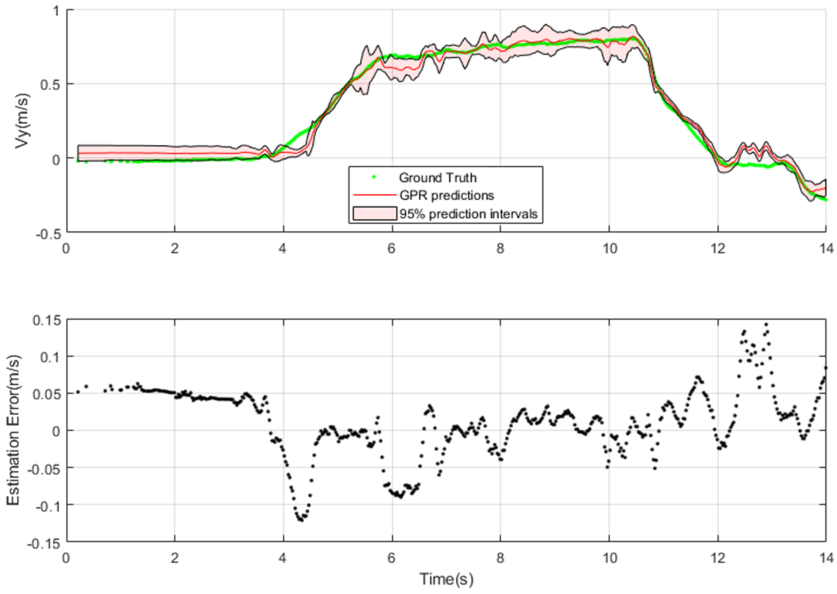

Figure 6.15: Lateral velocity estimation using GPR method for test 3, using the healed reference data.
Top: Ground truth, estimated values, and the 95% intervals. Bottom: Estimation errors.

The estimation statistics for these four estimations are shown in Table 6.4. Based on the given information in this section, the run-time for NW method is less than run-time of the GPR method. The need for updating the covariance matrix in GPR method can be the main reason of this time difference. The individual average run-time is increased with a small percentage after healing.

The results show significant improvements in estimation performance after healing the reference data set for both NW and GPR methods. The root mean square error has decreased for both estimation methods and the r^2 values are increased. Hence, it is concluded that healing the reference data set has improved the estimation accuracy.

Table 6.4: Estimation statistics for lateral velocity Test 3.

	Test Points	Total Run-Time (s)	Individual Average Time (ms)	RMSE (m/s)	r^2
NW method, Healed	520	3.14	6.0	0.033	0.9939
NW method, Original	520	2.96	5.9	0.090	0.9642
GPR method, Healed	520	8.44	16.2	0.046	0.9831
GPR method, Original	520	7.69	14.8	0.093	0.9489

6.3.3 Lateral Velocity Test 4 With Healing

The fourth lateral velocity estimation test is studied in this section. This is another experimental test performed by the Chevrolet Equinox vehicle. All the test data used for this estimation are measured using the sensors on the vehicle. The lateral and longitudinal velocities are measured using a GPS sensor installed on the vehicle.

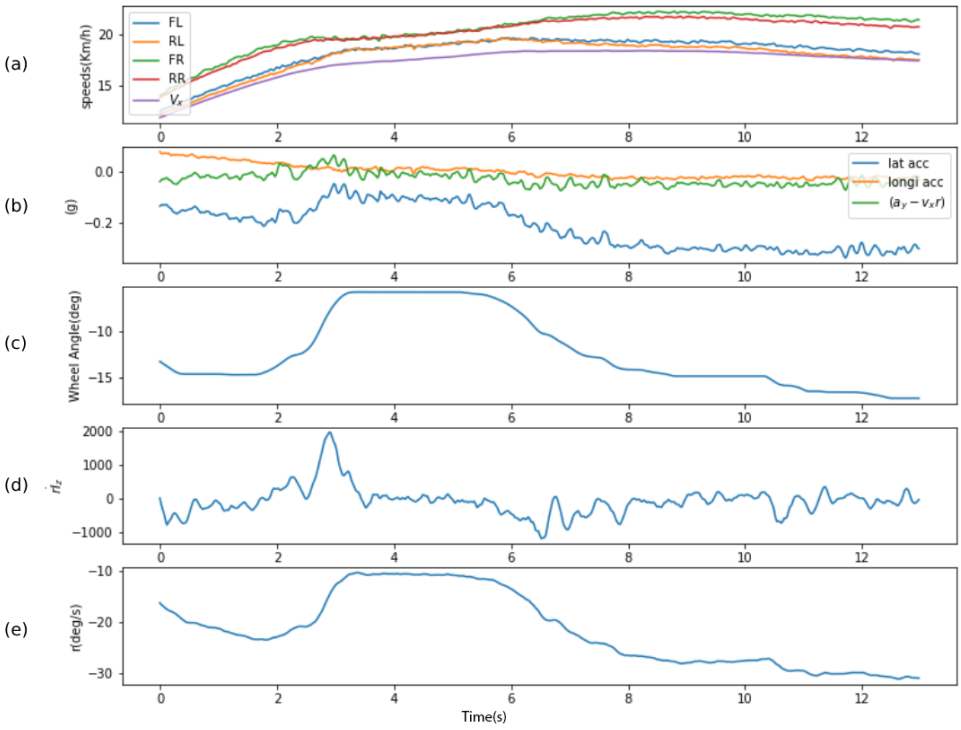

Figure 6.16: Sensor measurements for lateral velocity test 4.
a) Four wheel speeds and longitudinal velocity. b) Longitudinal and lateral accelerations, lateral velocity derivative. c) Front wheel angle. d) Yaw moment of inertia times yaw acceleration rate. e) yaw rate.

As shown in Figure 6.16, this test is performed with longitudinal velocity within a low range(less than $20kph$). The steering input for this test consists of harsh steering inputs. Considering the harsh steering input, the vehicle can be near the edge of stability. The fact that this test is performed on an icy road condition makes the estimation more difficult.

The lateral velocity for this test is estimated using the NW and GPR algorithms with

two reference data sets for comparison. First, the test is estimated using the original reference data set. This is the same data set used for other tests of this study. Then, the reference data set is healed using the healing method presented in Section 6.3.1. By comparing the estimation results for these reference data sets, the effect of healing on the estimations is explained.

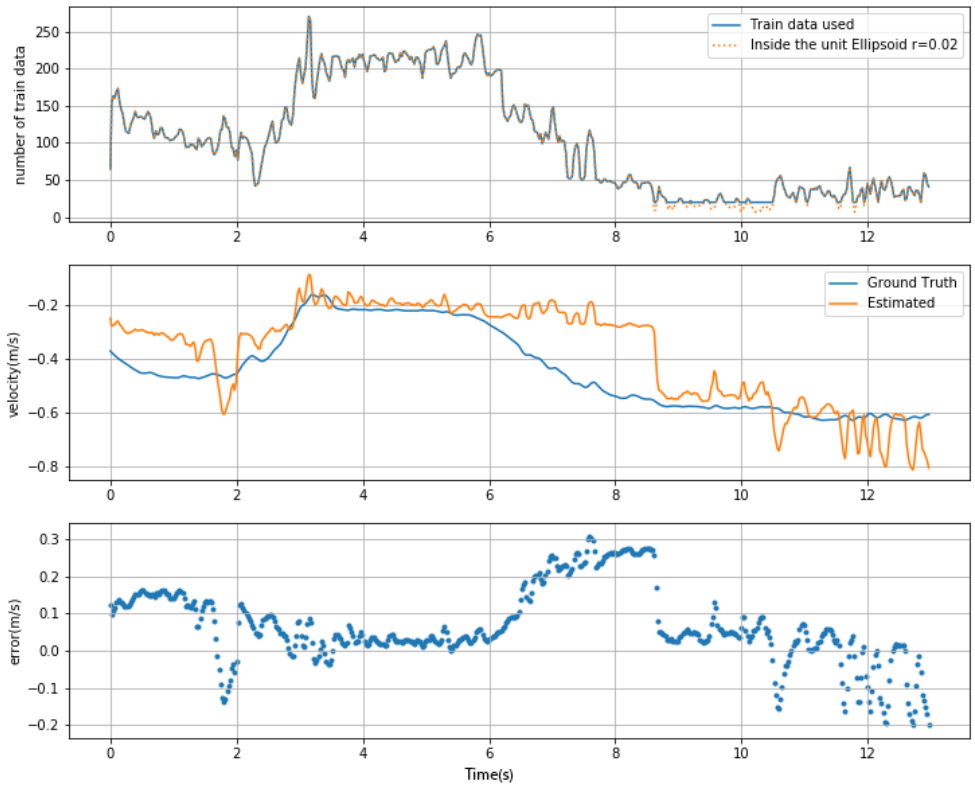

Figure 6.17: Lateral velocity estimation using NW method for test 4, using the original reference data.
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors.

As shown in Figure 6.17, the estimation is performed on the test data using the local reference points, with the NW method. After point $t = 6s$, the number of reference points in the vicinity of the test points drops sharply. Hence, a bias issue happens. As explained earlier, since the test points are located near a dense region, the result of data-driven

estimation is biased in that area. After $t = 8s$ the number of reference points around the test points is very low, so the reference points for estimation are selected using the KNN method. Lack of proper reference points in that area causes the enlargement of the unit ellipsoid around the test point, consequently, increase in estimation variance.

The result of the estimation using NW method with healed reference data is presented in Figure 6.18. The number of reference points within the unit ellipsoids is increased, significantly. In addition to better accuracy, presence of more reference points in vicinity improves the smoothness.

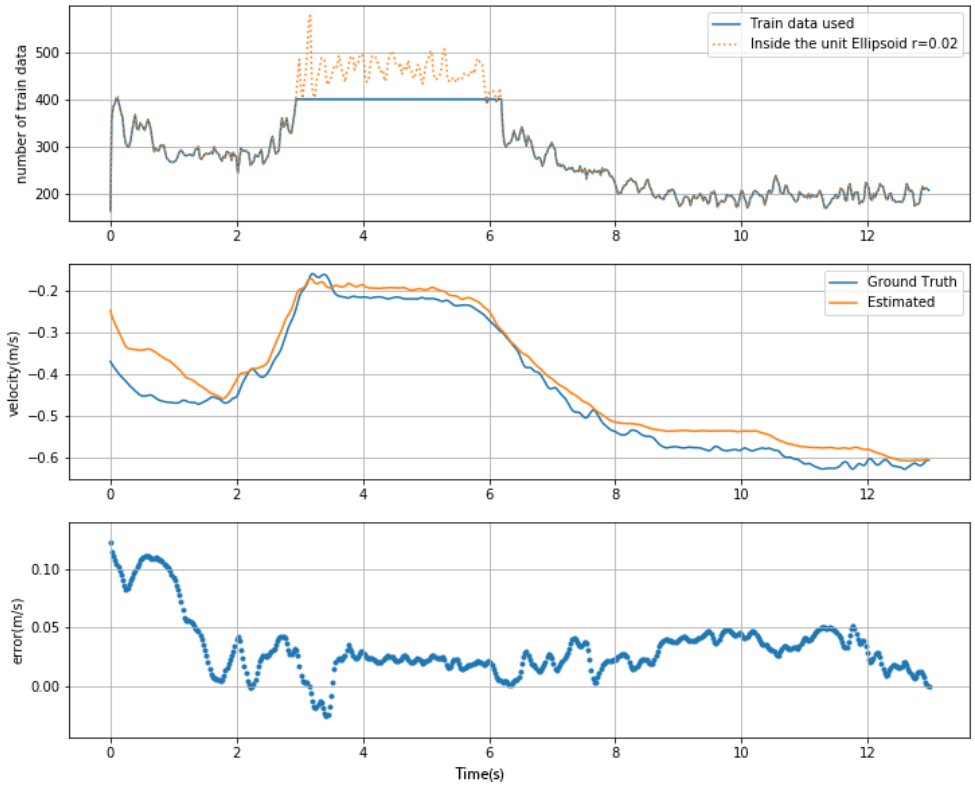

Figure 6.18: Lateral velocity estimation using NW method for test 4, using the healed reference data.
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors.

The estimations of lateral velocity using GPR method are presented in Figures 6.19

and 6.20. The results resemble those of the NW method. With considering the additional reference data for healing the reference data set, performance is improved and bias error is decreased.

As shown in Figure 6.19, the prediction bonds get larger in the areas with insufficient and less confident reference data. In contrast, as shown in Figure 6.20, after healing the reference data set, the estimation confidence intervals as well as the estimation variances, got smaller. This is showing the great effect of healing on the estimation accuracy.

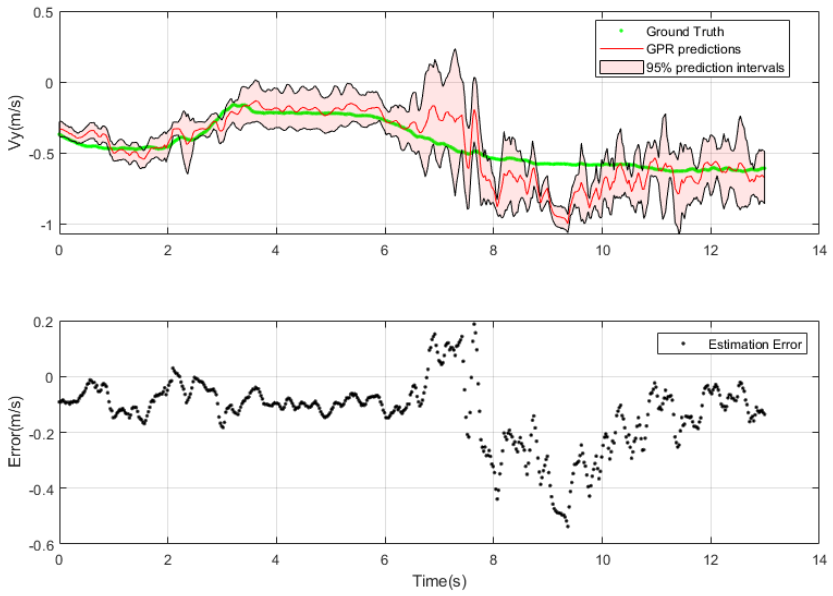

Figure 6.19: Lateral velocity estimation using GPR method for test 4, using the original reference data.
Top: Ground truth, estimated values, and the 95% intervals. Bottom: Estimation errors.

The estimation statistics for these four estimations are shown in Table 6.5. Based on this information, the run-time for the NW method is less than the run-time for the GPR method, which is mainly because of the need for updating covariance matrices within the GPR method. The individual average run-time is increased by a small percentage after healing.

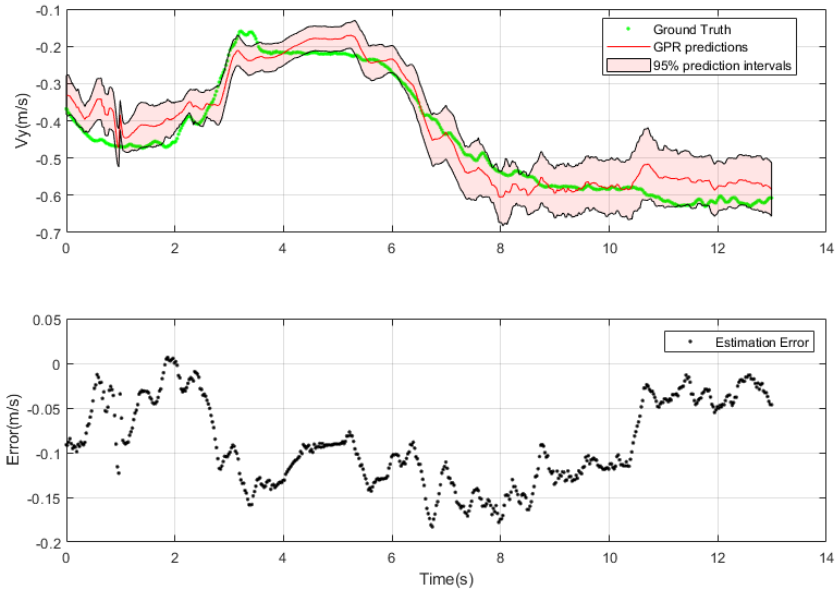

Figure 6.20: Lateral velocity estimation using GPR method for test 4, using the healed reference data.
Top: Ground truth, estimated values, and the 95% intervals. Bottom: Estimation errors.

The results show that the amount of r^2 values that shows the linear correlation has increased significantly after healing. Also, RMS errors are decreased after healing.

In general, it can be concluded that the healing has increased the estimation accuracy comparatively.

Table 6.5: Estimation statistics for lateral velocity Test 4.

	Test Points	Total Run-Time (s)	Individual Average Time (ms)	RMSE (m/s)	r^2
NW method, Healed	520	4.21	8.1	0.041	0.9712
NW method, Original	520	3.87	7.4	0.123	0.6841
GPR method, Healed	520	7.44	14.2	0.085	0.9214
GPR method, Original	520	6.83	13.1	0.121	0.7210

6.4 Lateral Velocity Estimation Using Healed data for Harsh Maneuvers

In this section, another healing method is used for healing the reference data set and making the reference data richer.

Some of the regions that are not well observed by the reference data are the edge scenarios. These are the situations that the vehicle is near the edge of stability. A harsh maneuver with high velocity, sudden steering while braking or a maneuver with very low road friction coefficient can be named as examples of edge scenarios. In all of the edge scenarios, vehicle tires are close to saturation. These situations can happen when the road friction is very low, or the tires are facing combined slip scenarios.

Most of the maneuvers in the reference data are from stable regions. This will lead to the fact that the number of reference points to cover the boundary regions is comparatively less than other regions. Hence, the estimation in boundary regions have errors.

In this section, it is tried to make the data set richer in the above mentioned regions. First, it is explained that how the healing process is done by using a high fidelity vehicle model, which is a CarSim vehicle model. Afterwards, two harsh tests are estimated in order to demonstrate the effect of this healing method.

6.4.1 Healing Using Manifolds of Harsh Maneuvers

The goal of the proposed healing process in this section is to improve the estimation of edge scenarios. In these scenarios, the vehicle is near the edge of instability, and close to losing control. For this purpose, some harsh maneuvers need to be performed using a high-fidelity vehicle model. Similar to the previous one, a Chevrolet Equinox simulation model is used in CarSim to generate data for healing the reference data set. The simulation scheme is shown in Figure 6.21.

The healing process is similar to the last healing process, presented in Section 6.3.1, with a difference. In the first healing process, the target population of healing was a specific test and the test inputs were used to regenerate the maneuver. However, in this healing process, the goal is to make the reference data richer around the border regions.

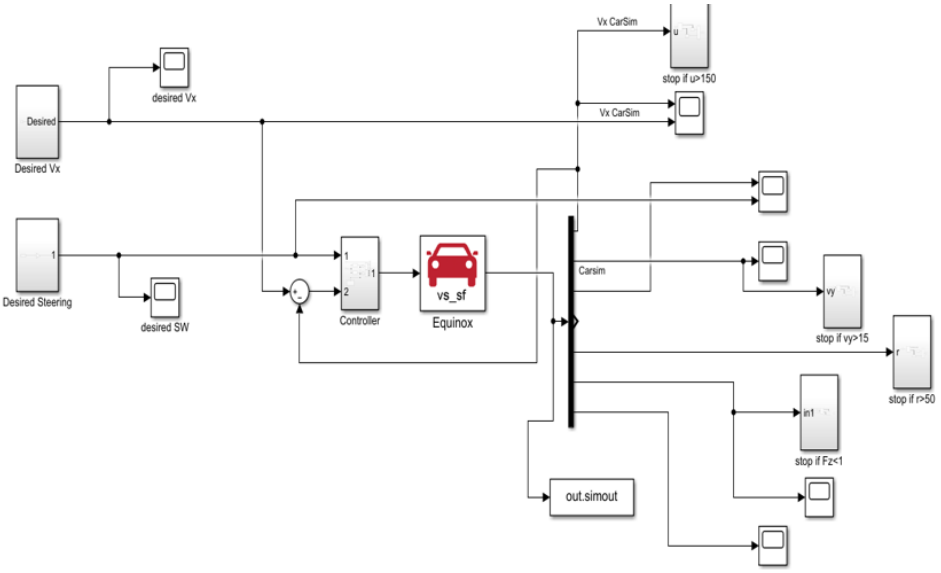

Figure 6.21: Simulation scheme for healing data generation.

To make this healing process happen, two questions need to be answered:

- First, where is the border of stability and what are its properties?
- Second, how should the vehicle be pushed to reach the target regions?

The explanation for the given questions are presented in the following.

Edge of Stability Regions

Finding the edge of stability has great importance in control systems implemented in vehicle dynamics. C.G.Bobier and J.C.Gerdes [1] have used an envelope for controlling and keeping the vehicle within the envelope. They have used the single-track (bicycle) model for their vehicle model. For calculating the tire forces, they have used a variant of the Fiala nonlinear brush tire model, with one coefficient of friction and a parabolic force distribution.

As stated in the article, the envelope is calculated based on three criteria.

1. Maximum and minimum steady state yaw rate:

$$r_{max/min} = \pm \frac{\mu g}{V_x} \tag{6.1}$$

2. Maximum and minimum rear slip angles:

$$\beta_{\alpha r\_max/min} = \frac{br}{V_x} \pm tan(\alpha_{s1,f}) \tag{6.2}$$

3. Maximum and minimum front slip angles:

$$\beta_{\alpha f\_max/min} = -\frac{ar}{V_x} + tan(\pm\alpha_{s1,f} + \delta) \tag{6.3}$$

In these equations, a and b are the front and rear axle to CG distances, V_x is the longitudinal velocity, r is the yaw rate, δ is the front wheel angles, g is the gravity acceleration, and μ is the road friction coefficient. Also, the slip angle corresponding to the peak tire forces is shown as $\alpha_{s1,f}$, which is calculated as $\arctan(\frac{3\mu F_z}{C_\alpha})$.

To show the stability envelope the β-r phase plane is used. The borders stated in Equations (6.1), (6.2), and (6.2) are shown for an arbitrary maneuver in Figure 6.22. The borders are shown with solid black lines. Two horizontal lines are corresponding to the max and min yaw rates. Two lines with positive-slopes are showing the max and min rear slip angles. Negative-slope inclined lines are also showing the min and max front slip angles.

To avoid unnecessary complexity, in this study the introduced intervals related to the yaw rate and rear slip angles are considered. This envelope is shown with red lines in Figure 6.22.

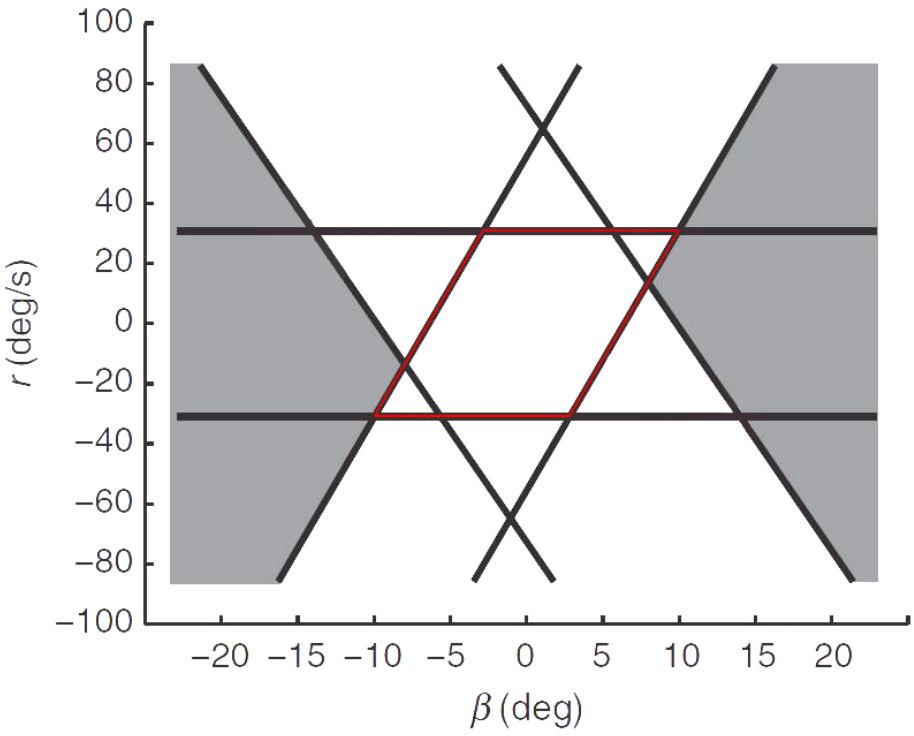

Figure 6.22: Stability envelopes for arbitrary maneuver parameters. The red envelope is used in this study [1].

Maneuvers

To push the vehicle toward the envelope edges, maneuver inputs are needed. Two maneuvers are considered for this purpose. The first one is a flick maneuver that is performed using constant longitudinal velocity. The steering input is a step steer followed by a greater counter steering input. The load transfer between tires will make the tires slip and cross the stability boundaries. This test is repeated for various steering input levels, various longitudinal velocity ($V_x \in \{40, 60, 80, 100\}$(kph)), and various friction coefficients ($\mu \in \{0.4, 0.6, 0.8, 0.9\}$).

As an example, the steering input of a flick maneuver is shown in Figure 6.23. With repeating the test using different steering wheel angle inputs, a set of curves are generated in

the phase-plane that can be used for healing. A set of maneuvers are shown in Figure 6.24, performed with constant velocity of $V_x = 80$kph and road friction coefficient of $\mu = 0.6$, and different steering angles.

As it is apparent from the equations stated for the stability envelope, the stability region will get smaller as the longitudinal velocity increases. In addition, the stability region will get smaller as the friction coefficient gets smaller.

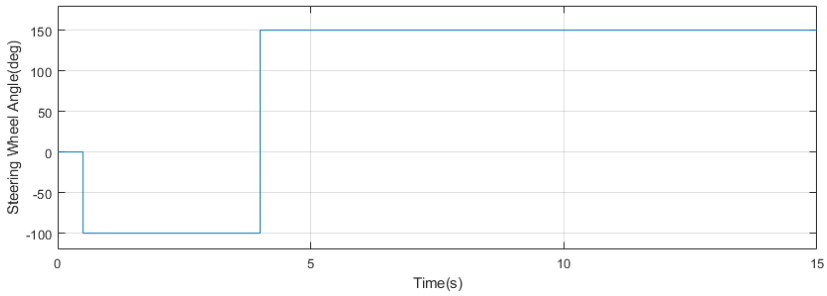

Figure 6.23: Steering wheel angle input of a flick maneuver.

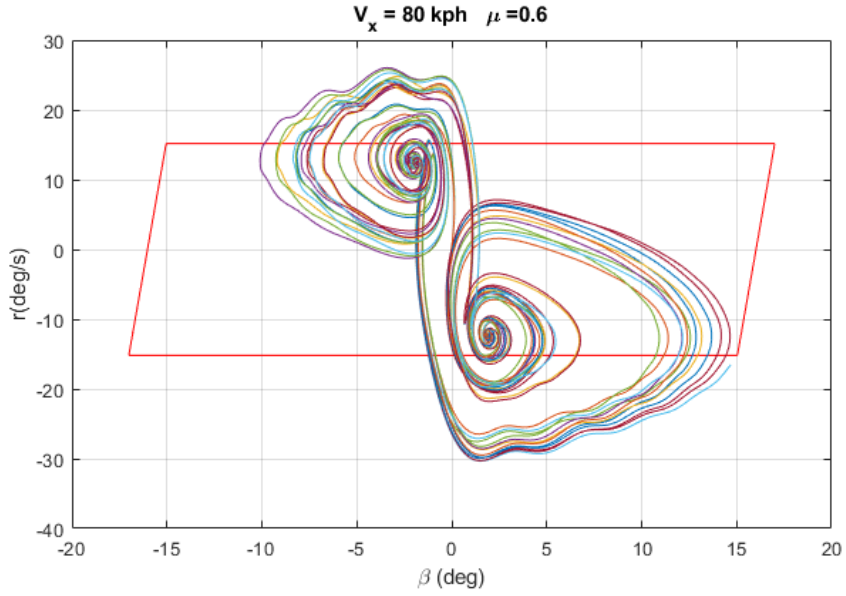

Figure 6.24: Flick the maneuvers using different steering wheel angle inputs for $V_x = 80$ kph and $\mu = 0.6$

The second series of tests are performed using acceleration in turn maneuvers. In these maneuvers constant steering wheel angles are applied. The longitudinal velocity is increased from 10kph to 150kph with different slopes. The input velocity patterns are shown in Figure 6.25. For different friction coefficients, the tests are repeated until the manifolds are made. The test results for two different friction coefficients are provided in Figure 6.26.

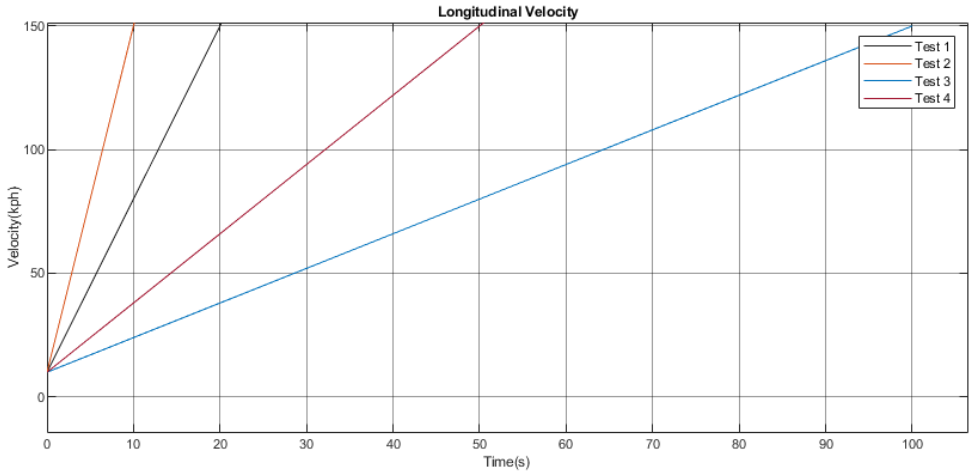

Figure 6.25: Different velocity inputs for acceleration in turn maneuvers.

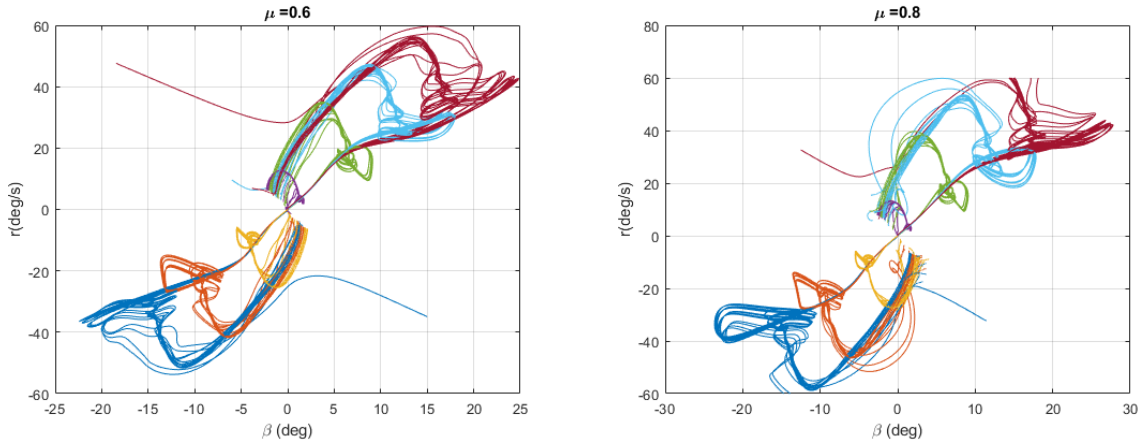

Figure 6.26: Test results of acceleration in turn maneuvers for $\mu = 0.6$ and $\mu = 0.8$. Each color is corresponding to a steering input.

Because of the fact that the stability envelopes are driven assuming constant longitudinal velocity, the envelopes will not be applicable for acceleration in turn maneuvers. instead of that, some stop conditions are used to stop the simulations in case the vehicle is getting out of control.

Stop Conditions

In the maneuvers performed for healing the reference data set, those points near the edge of the stability region are of interest. So, the points outside the envelope, yet not completely out of control, are the target points. For stopping the simulation in order not to collect data corresponding to the completely out of control regions some stopping criteria are considered.

The stop conditions, also shown in Figure 6.21 as subsystems, are as follows:

- If the longitudinal velocity gets greater than $150kph$.
- If the absolute value of lateral velocity gets greater than $15kph$.
- If the absolute value of yaw rate becomes greater than $50deg/s$.
- If any of the tire normal forces becomes zero.

At any time that one of the above conditions is true, the simulation is ceased. A total number of 320 tests are generated using acceleration in turn and flick maneuvers. These test results are used to form a data set for the healing process. The newly generated data are added to the original reference data to study the performance of estimation in presence of reference data healing.

Although the healing process is performed using simulation tests with the Equinox vehicle model, the impact of healing on other test vehicles is studied in the following sections.

6.4.2 Lateral Velocity Test 5 With Healing

The fifth lateral velocity estimation test is studied in this section. This is another experimental test performed by a Chevrolet Equinox vehicle. All the test data used for this test are measured using the sensors on the vehicle. The lateral and longitudinal velocities are measured using a GPS sensor installed on the vehicle. For other needed measurements, IMU sensors are used.

As shown in Figure 6.27, this test is performed with low longitudinal velocity values and harsh steering inputs. Considering the harsh steering input, and the wet road condition the vehicle is near the edge of stability.

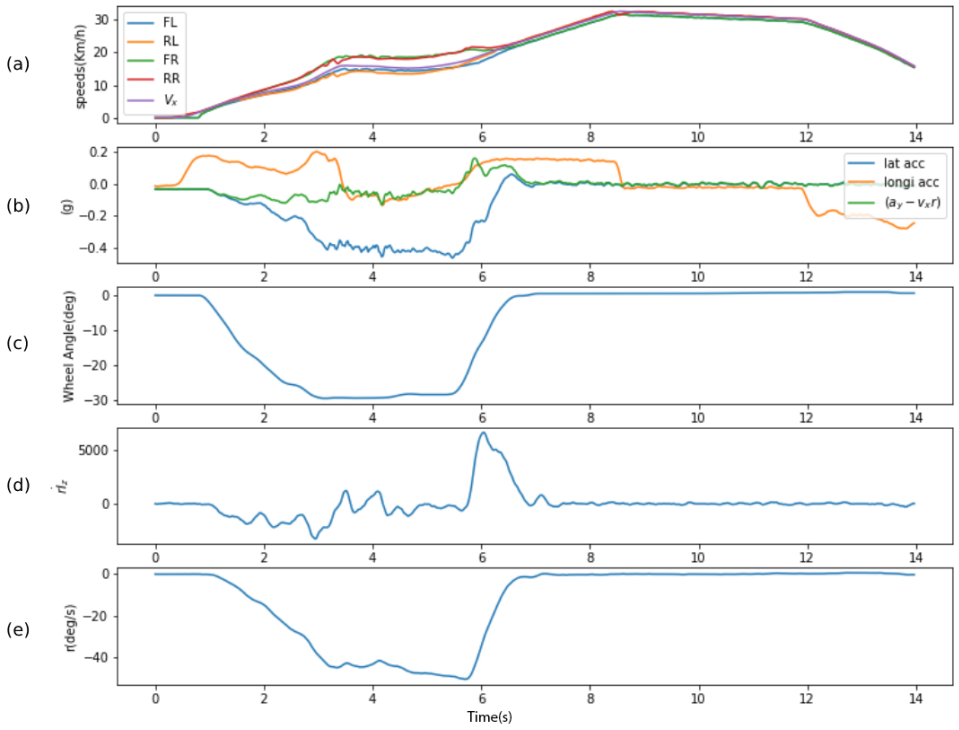

Figure 6.27: Sensor measurements for the lateral velocity test 5.
a) Four wheel speeds and longitudinal velocity. b) Longitudinal and lateral acceleration, lateral velocity derivative. c) Front wheel angle. d) Yaw moment of inertia times yaw acceleration rate. e) yaw rate.

The lateral velocity in this test is estimated using the NW and GPR algorithms with

two reference sets. First, the lateral velocity is estimated using the original reference data. This is the same data set used for other tests of this study. Then, the reference data set is healed and modified by healing method presented in Section 6.4.1.

In addition to the mentioned reference data sets, another reference data set is used for the NW test. The third reference data set consists of both original and the healing data after an additional round of data reduction. This third reference set has the lowest number of reference points within itself.

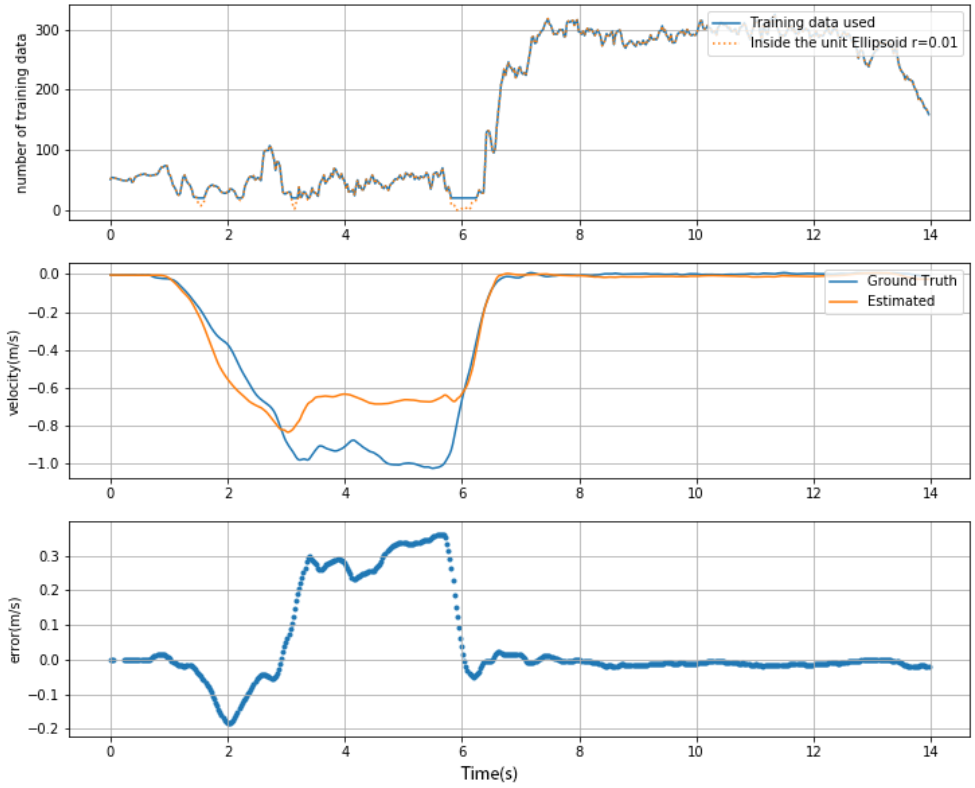

Figure 6.28: Lateral velocity estimation using NW method for test 5, using the original reference data set.
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors.

The result of lateral velocity estimation using the NW algorithm is presented in Figure 6.28. As it is shown in the plots, from $t = 3$ to $t = 6$ the estimator is facing an error.

Although the error is not that large, it is noticeable. Due to the local reference points that have relatively smaller lateral velocities, this error occurs.

The number of reference points used for most of the test points before time $t = 6.5$ seconds is smaller than 100 points. This is because of using the original reference set.

The test results for the healed reference data are shown in Figure 6.29. After adding the healed reference set, the number of reference points around each test point has increased significantly. The maximum error happens at $t = 6s$ which has the lowest number of reference points.

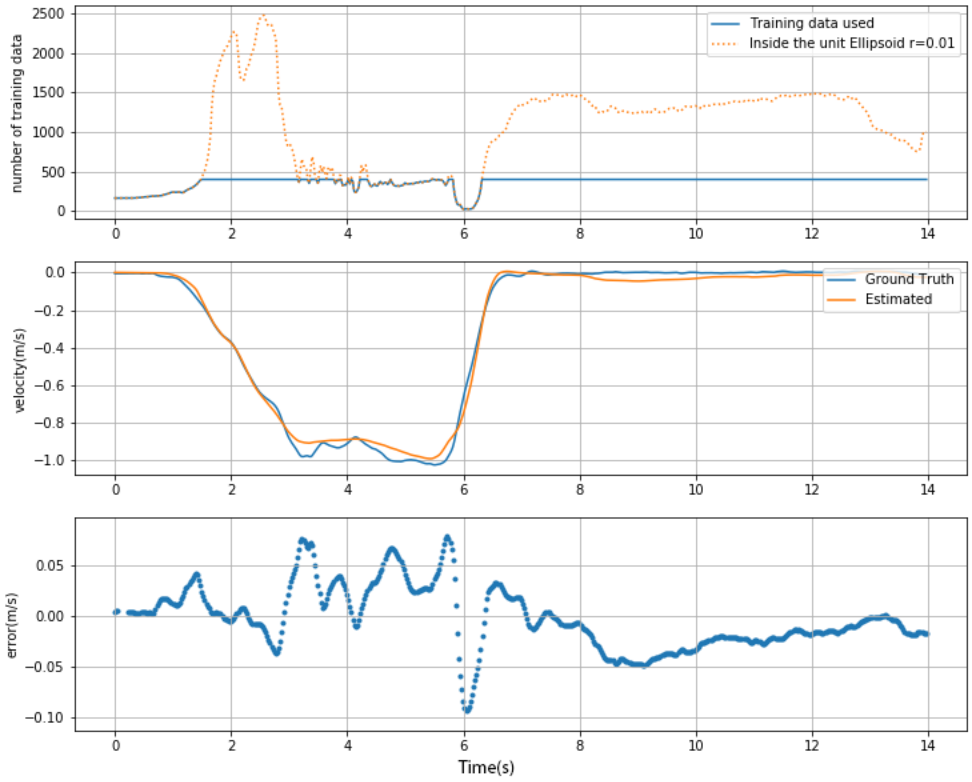

Figure 6.29: Lateral velocity estimation using NW method for test 5, using the healed reference data. Three plots are showing from top to bottom:
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors

Since the regenerated points related to the healing process has increased the reference

points in the neighborhood, the reduction algorithm is used to reduce the reference data one step further. So, this reference set is basically the same as the healed reference set, with a lower data density. The results of lateral velocity estimation on the fifth test is presented in Figure 6.30. As shown, the number of reference points around test points has a significant decrease in comparison with the non-reduced version, Figure 6.29. This version shows a very rapid estimation with a small reduction in accuracy in exchange. This shows the great capacity of this method for real time applications.

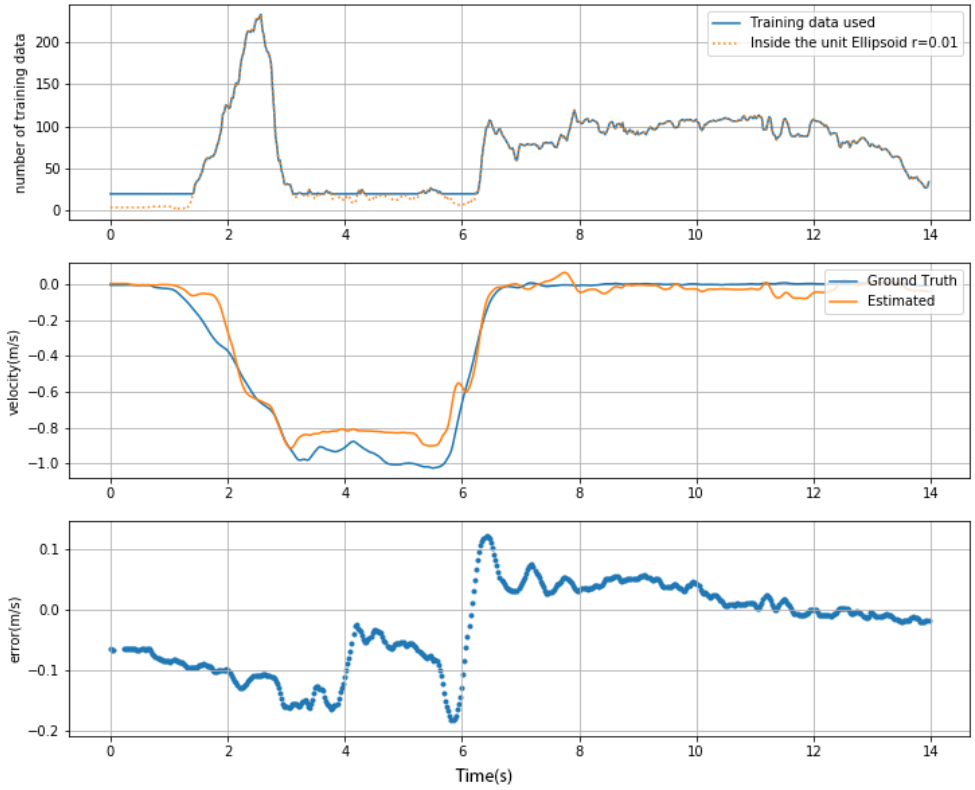

Figure 6.30: Lateral velocity estimation using NW method for test 5, using the healed and double reduced reference data. Three plots are showing from top to bottom:
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors

To compare both estimations, run-time is decreased and the accuracy is a little compromised. In therms of run-time, the test estimation using healed reference set has taken

27.7 seconds for 554 points. And the test estimation using the healed and double reduced reference data has taken only 4.7 seconds.

Other estimation measurements are presented in Table 6.6.

The estimation of lateral velocity using GPR method are presented in Figures 6.31 and 6.32. The results resemble those of the NW method. However, the results in NW had slightly better accuracy. This is because in the GPR estimation, only 30 nearest reference points are used for the estimation. This issue can be addressed by considering all the reference points inside the unit ellipsoid around each test point, however, the solution is much more time-consuming.

Adding the generated data to the reference set and reducing using the reduction algorithm increase the estimation confidence and addressed the large errors between 4 and 6 seconds.

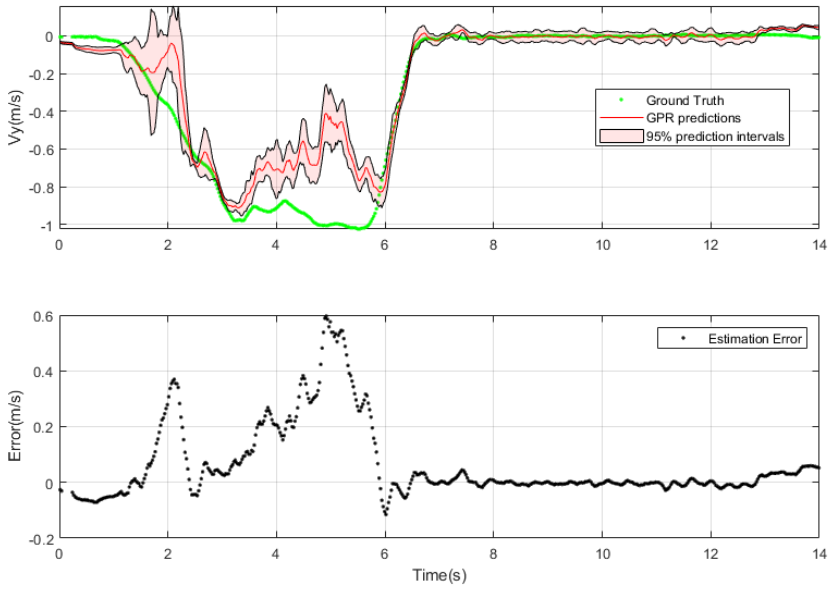

Figure 6.31: Lateral velocity estimation using GPR method for test 5, using the original reference data. Two plots are showing from top to bottom:
Top: Ground truth, estimated values, and the 95% intervals. Bottom: Estimation errors.

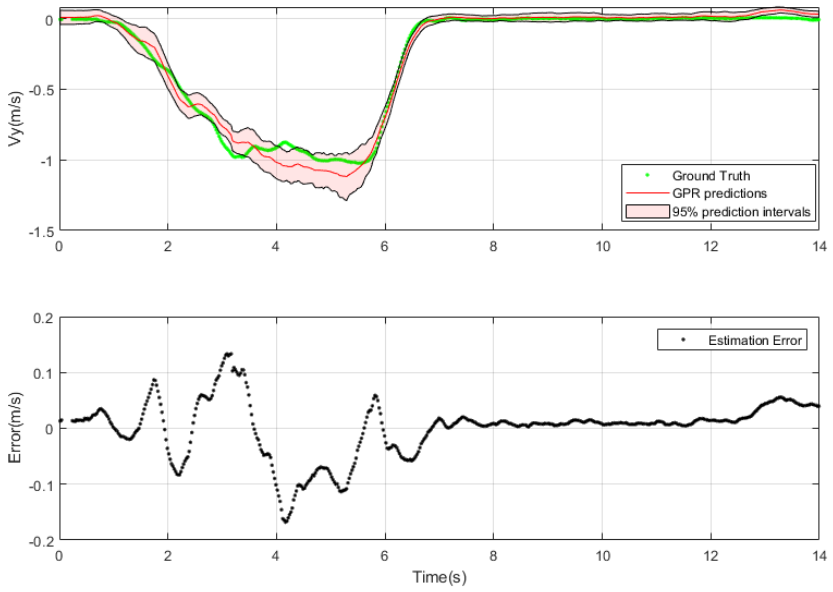

Figure 6.32: Lateral velocity estimation using GPR method for test 5, using the healed reference data. Two plots are showing from top to bottom:
Top: Ground truth, estimated values, and the 95% intervals. Bottom: Estimation errors.

Table 6.6: Estimation statistics for lateral velocity Test 5.

	Test Points	Total Run-Time (s)	Individual Average Time (ms)	RMSE (m/s)	r^2
NW method, Healed	554	29.73	53.6	0.074	0.9837
NW method, Healed and double reduced	554	4.71	8.5	0.089	0.9632
NW method, Original	554	7.07	12.8	0.127	0.9118
GPR method, Healed	554	11.52	20.8	0.097	0.9322
GPR method, Original	554	9.07	16.4	0.146	0.8360

6.4.3 Lateral Velocity Test 6 With Healing

The sixth lateral velocity estimation test is studied in this section. This test is a simulation test performed by the Cadillac Escalade vehicle model in CarSim. The main difference that distinguishes this test from the other mentioned tests is that this vehicle has not been used in the healing process. However, the effect of the vehicle-independent lateral velocity estimation can be assessed by this test.

As depicted in Figure 6.33, this test is a harsh flick maneuver performed with an almost constant longitudinal velocity of 80 kph. Since the steering input has a large counter action at $t = 3s$, with considerable load transfer effect, the tires nearly reach the saturation condition. This can be also concluded from the large lateral acceleration occurring after $t = 3s$.

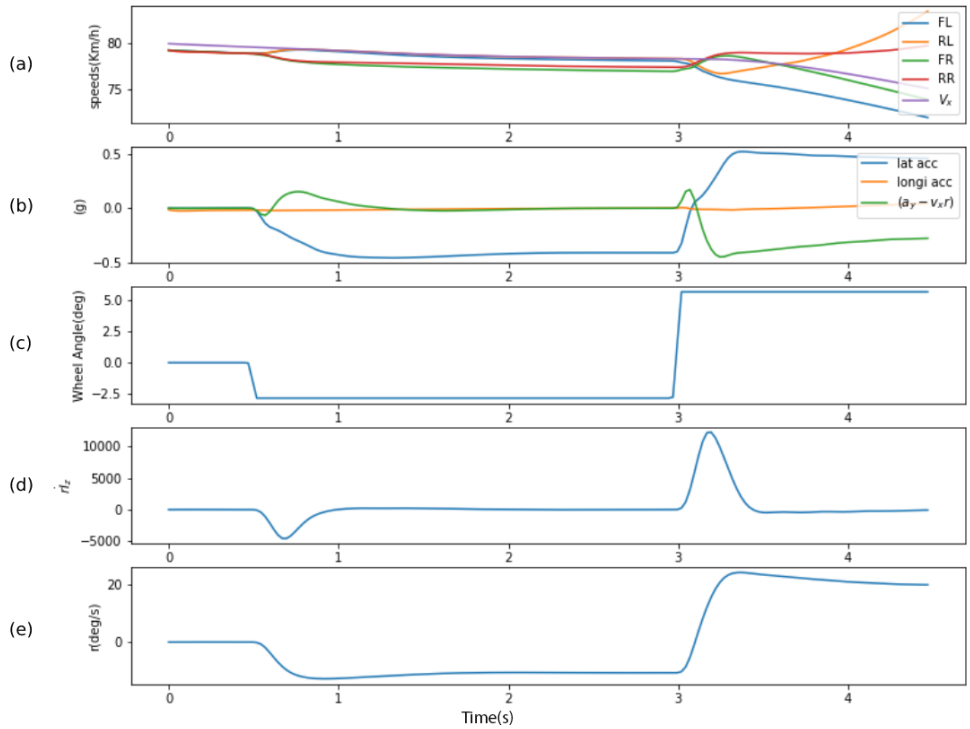

Figure 6.33: Sensor measurements for the lateral velocity test 6.
a) Four wheel speeds and longitudinal velocity. b) Longitudinal and lateral accelerations, lateral velocity derivative. c) Front wheel angle. d) Yaw moment of inertia times yaw acceleration rate. e) yaw rate.

This test is relatively short. Because of the harsh maneuver inputs, the system goes toward instability. With the increase of lateral velocity after $t = 3$s, the vehicle gets out of control. Hence, only the part that the vehicle is still not out of control is used for this estimation.

This test is estimated using the NW and GPR algorithms with two reference data sets. First, the test is estimated using the original reference data. This is the same data set used for other tests of this study, as well. Then, the reference data set is healed using the healing method presented in Section 6.4.1. In this healing method, only the harsh simulation maneuvers performed by the Chevrolet Equinox vehicle are added to the reference data.

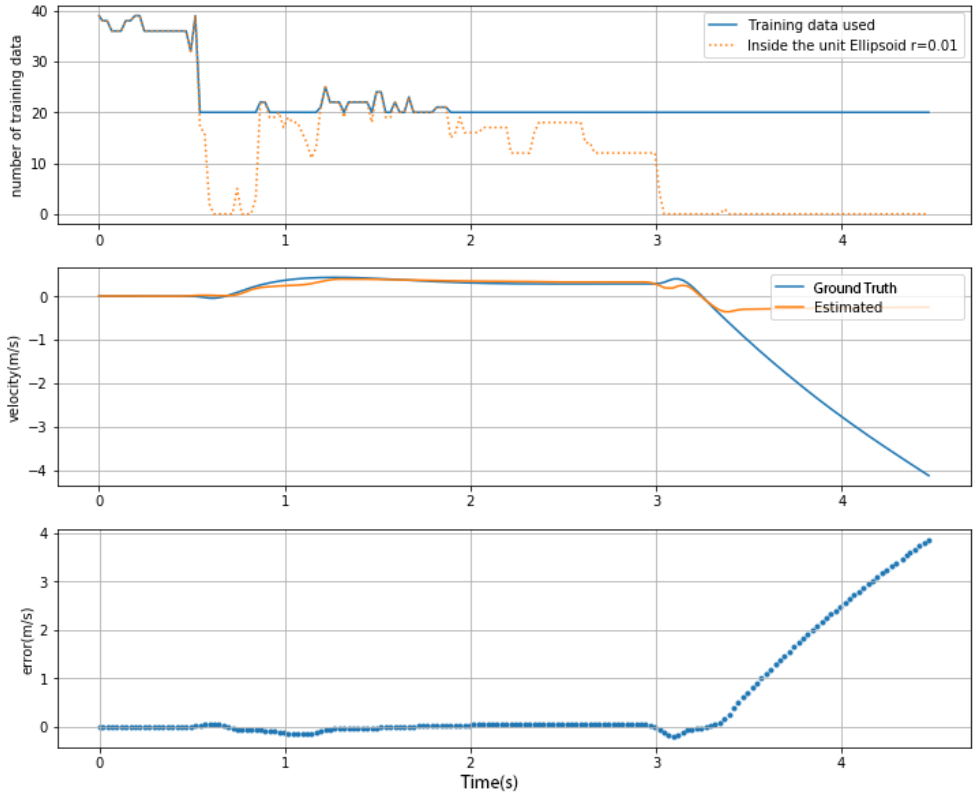

Figure 6.34: Lateral velocity estimation using NW method for test 6, using the original reference data. Three plots are showing from top to bottom:
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors.

The result of lateral velocity estimation using the NW algorithm is presented in Figure 6.34. The first three seconds of the test show a smooth and relatively accurate estimation. However, after the intense counter steering, at $t = 3s$, the estimation does not follow the lateral velocity ground truth. Also, it is seen that the number of the reference points decreases to zero. For the estimation of these test points, only twenty nearest reference points are utilized as observations for the estimations. The twenty nearest points may not be in the close vicinity of the test points, hence, the estimations have errors.

As shown in Figure 6.34, the lateral velocity is increases with a steep slope after the counter steer at $t = 3s$. This shows that the vehicle reaches the border regions, which have sparse data in the reference data set.

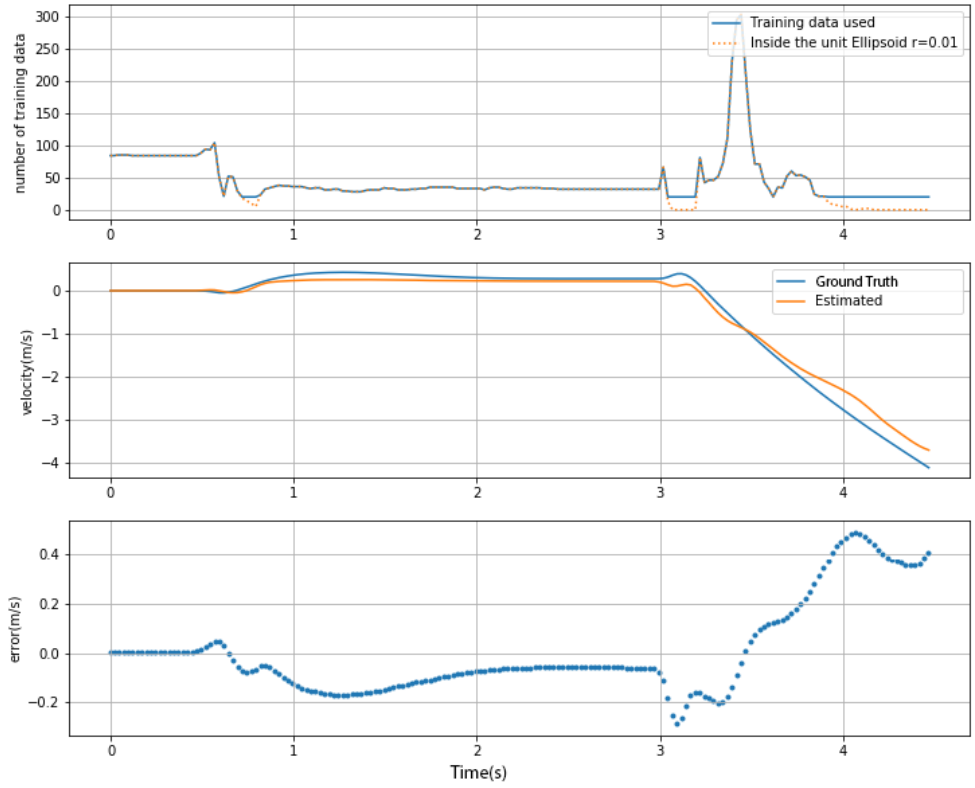

Figure 6.35: Lateral velocity estimation using NW method for test 6, using the healed reference data. Three plots are showing from top to bottom:
Top: Number of reference points in the unit ellipsoid and the number of reference points used for estimation of each test point. Middle: Ground truth and estimated values. Bottom: Estimation errors.

The test results for the healed reference data set are shown in Figure 6.35. The reference data set is healed using the harsh maneuvers performed by the Chevrolet Equinox simulation model. Hence, no more reference data are added to the reference data, from the Cadillac Escalade vehicle model.

It can be seen that the number of the reference points around the test points increase for all of the test points, except for those after $t = 4$s. Although the number of reference points in the unit ellipsoid, after $t = 4s$, are limited, the accuracy of the velocity estimation is increased. This is because of additional reference points compared to the original data set.

The maximum error of estimation has happened around $t = 4s$. The number of reference points inside the unit ellipsoid reaches zero and the points selected by KNN method can be biased.

The estimations of lateral velocity using GPR method are presented in Figures 6.36 and 6.37. The behavior observed in Figure 6.36 is the same as that in Figure 6.34, due to lack of proper data availability.

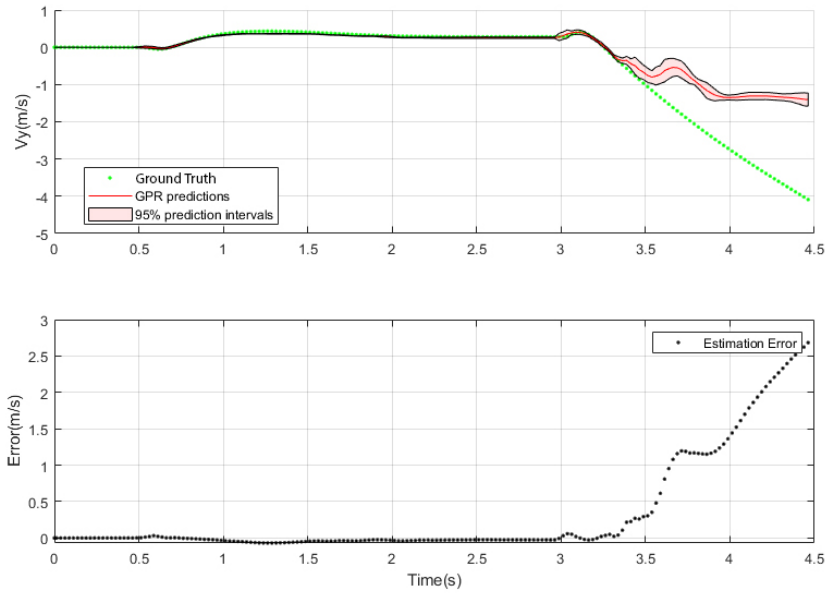

Figure 6.36: Lateral velocity estimation using GPR method for test 6, using the original reference data. Two plots are showing from top to bottom:
Top: Ground truth, estimated values, and the 95% intervals. Bottom: Estimation errors.

The results after healing the reference data set resemble those of the NW method. The comparison of two Figures 6.36 and 6.37 reveals that the healing process has greatly improved the estimation accuracy. The impact of healing on the estimation results is undeniable, even though only the 30 nearest reference points are used for each estimation with the GPR method.

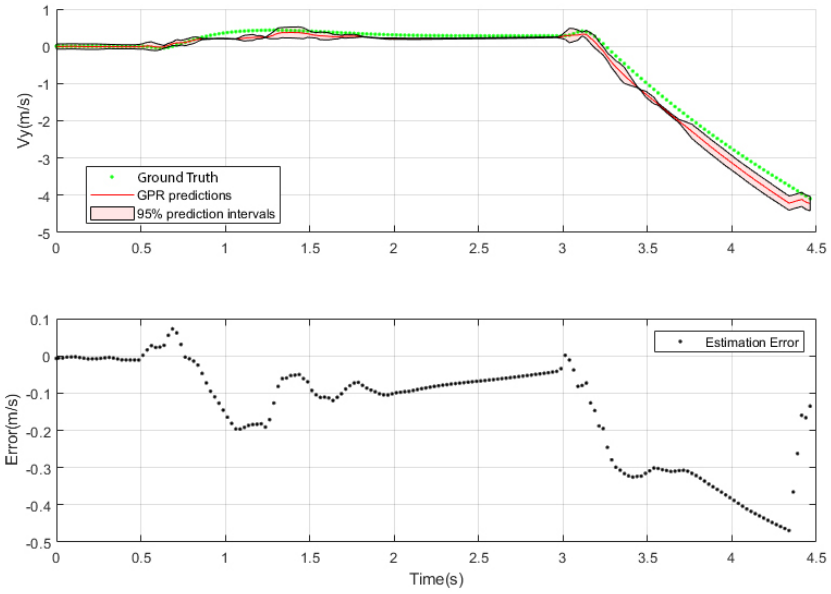

Figure 6.37: Lateral velocity estimation using GPR method for test 6, using the healed reference data. Two plots are showing from top to bottom:
Top: Ground truth, estimated values, and the 95% intervals. Bottom: Estimation errors.

The statistical properties of the estimations for lateral velocity test 6 are presented in Table 6.7. The total run-time increases for both methods after healing and adding the generated data. Considering the overall increase in the total number of reference points in the unit ellipsoid, the longer run-time is expected. The NW method is faster than the GPR. Although the GPR uses only 30 reference points for each estimation, the need for updating the covariance matrix has made it slower.

After healing the reference data set, root mean square error of estimations decreases significantly. It is because of adding more relevant points to the test point regions.

Furthermore, the r^2 values, which show the linear correlation between the estimation

results and the ground truth, also increase after healing.

Table 6.7: Estimation statistics for lateral velocity Test 6.

	Test Points	Total Run-Time (s)	Individual Average Time (ms)	RMSE (m/s)	r^2
NW method, Healed	180	5.29	29.4	0.186	0.9963
NW method, Original	180	2.10	11.6	1.211	0.6394
GPR method, Healed	180	9.61	53.4	0.197	0.9900
GPR method, Original	180	6.64	36.9	0.783	0.4970

This test confirms two main claims of this study. First, the data-driven methods rely on the availability of reference data in the vicinity, and they can perform good estimations in the presence of proper, rich reference data. Second, the reference data set can be healed using a vehicle to improve the estimation of another vehicle. In other words, the estimation methods can work independently from the vehicle type, which was used in reference data generation. With feature normalization, the generated data are universal and can be added to the reference data to be used for estimation of any other vehicles.

6.5 Summary

In this chapter, six lateral velocity tests, as well as two healing methods, were presented. Three different vehicles, including Cadillac Escalade, Chevrolet Equinox, and E-class simulation results were used. Also, both forms of simulation and experimental tests were studied among these six tests.

In the first two tests, two simulation tests from Escalade and E-class vehicle were used for estimation. A similar reference data set was used for the estimation of both tests and was shown that both NW and GPR methods had acceptable results. It was shown that the availability of proper rich data in the vicinity of test points improves the estimator accuracy. In case of not having enough reference points available in the test point neighborhood, the estimation cannot guarantee accuracy.

Test 3 and test 4 are two experimental tests performed by Chevrolet Equinox vehicle. For improving the estimation results of these two experimental tests, a healing method was offered. In this healing method, several simulation tests were performed. The steering wheel angle and longitudinal velocity signals were used from experimental test data to regenerate some similar simulation tests. These tests were performed on various road conditions to regenerate more points for healing.

The regenerated test points were added to the previous reference data. The final healed reference data set was a reduced version of this aggregated set. The data reduction algorithm makes sure the duplicated data are eliminated and the overpopulated regions are balanced. This algorithm increases the accuracy of estimation and mitigates bias issues.

Test 3 and test 4 both had shown results with errors less than 0.1 m/s after healing the reference data set. The healing process increased the number of reference points in the vicinity of test points.

Another region that usually has a lack of reference data, is the border region. Since most of the maneuvers included in the reference data set are from stable tests, the border regions encounter a lack of diverse and rich reference data. This fact causes bias issues when the vehicle is at the edge of instability or it is getting out of control. To heal the reference data at the border regions, a healing algorithm was proposed.

Tests used for healing of border regions were harsh maneuvers performed by the Chevrolet Equinox simulation model in CarSim. It was tried to push the vehicle out of the stable regions to collect data from border situations. Two main maneuvers considered for this task were flick maneuvers with constant velocity and acceleration in turn. Both maneuvers were repeated for different road conditions. The generated data were added to the original reference data set and then reduced using a data reduction algorithm.

Test 5 and test 6 are two harsh tests considered to study the effect of healing using harsh maneuvers. In test 5 an experimental test was considered. This test was performed by the Chevrolet Equinox vehicle on a wet road. The lateral velocity was estimated using the original reference data, healed reference data, and a double reduced version of the healed reference data. It was shown that the healed versions had better performances than the non-healed version. In the healed version, test regions were richer in reference points.

Lateral velocity test 6 was a simulation test performed by Cadillac Escalade vehicle in CarSim. In this harsh maneuver, the vehicle was facing great amounts of lateral velocity. It was shown that after healing the reference data using Equinox-performed tests, the number of reference points in the vicinity of test points was increased, consequently, the estimation accuracy improved significantly.

Test 6 showed that the healing method performed by one vehicle can improve the results of estimation for other vehicles too. This is because of the vehicle-independent structure of the estimation.

To conclude, the proposed algorithms were capable of estimating the lateral velocity in presence of local reference points. Furthermore, by using normalized selected features, the vehicle-independent lateral velocity was achieved.

Chapter 7

Conclusions And Future Works

This study presented and validated a universal data-driven state estimation algorithm through simulation and experimental tests. The algorithm uses kernel-based machine learning methods for the estimation of vehicle lateral and longitudinal velocities. Also, by using normalized features it was shown that the estimation algorithm does not depend on the vehicle type. So, the reference data sets consist of generic data and do not need to be changed or reassembled if the vehicle changes.

In this chapter, the conclusions made from this study are presented and discussed. Furthermore, the suggested future works are explained.

7.1 Conclusions

The proposed data-driven algorithm was used for longitudinal and lateral velocity estimation, using the Nadaraya-Watson and GPR methods. Both of these methods are kernel-based algorithms, which allow the local reference points to be used for estimation. Both algorithms show effective results for lateral and longitudinal velocity estimations.

A bicycle model was considered to find the most effective features for the estimation. By performing the linear correlation tests, the most correlated features were selected for each estimation. A method was used to normalize the features to make the estimation general for any vehicle.

The features were selected by incorporating nominal vehicle parameters. This is mainly done to make the estimation method universal and generic. Different vehicle data were

included in the same reference data set that was used for the velocity estimation of different vehicles. The vehicle-independent estimations were shown to be effective on four different vehicles in this study. The reference data generated with this method are generic and ready to be used for any vehicle without modification. The methods could estimate the lateral and longitudinal velocities with errors less than 0.1m/s and 2kph in the presence of proper local reference data.

Another contribution of this study is using the local data for velocity estimation. Using the local data allows to run the estimation faster and evaluate the reference data quality in the neighborhood. In this approach, the reference points selected for each estimation are the nearest points to the point of interest. A feature importance comparison was implemented for the selection of input data. Since the kernel-based algorithms cannot capture complex relations between features, using the data within small intervals (local data) enables the kernel-based algorithms to perform smoother and more accurately in complicated scenarios.

A data reduction algorithm was used to reduce the volume of the reference data set. This made the estimation not only faster but also more accurate. The reduction algorithm eliminates the duplicated or very close reference points. The reduction algorithm also decreases the reference point numbers in very compact and dense regions of the reference data set. It decreases the chance of bias errors and increases the estimation algorithm effectiveness.

For longitudinal velocity estimation using NW and GPR methods, three tests with different maneuvers on various road conditions were presented. It was seen that the estimation algorithm provides accurate results in regions with enough reference data, and considerable errors happen mostly in sparse regions of the reference data set. The estimations performed by NW methods show shorter run times compared to the GPR method. Both the estimations showed fast estimations(more than 50 Hz), and they were suitable for real-time applications.

Six tests were estimated and analyzed with various simulation and experimental tests and different road conditions for lateral velocity estimation. The suggested universal lateral velocity estimation method worked on different test vehicles without a need to change or modify the reference data set. The estimation algorithm was able to estimate the lateral velocity with an error of less than 0.1m/s in the presence of available local reference points.

Since the local estimation algorithms presented in this study rely on the availability of reference points in the neighborhood, the estimation may have low accuracy if the test point is located in a sparse region. To address this issue, two data generation schemes were proposed, as healing algorithms. These two healing algorithms were implemented and

tested to make the reference data set richer and cover more regions. The algorithms showed improvements in estimation results of unseen experimental tests and harsh maneuvers. Also, it was shown that the healing with one vehicle could improve the estimation of other vehicles, as well. This confirms and aligns with the vehicle-independent estimation goal.

The longitudinal and lateral velocity estimation methods are fast and can be implemented on any vehicle for real-time application, with greater than 50 Hz frequency. The methods are accurate for all the trained regions, even in harsh situations.

7.2 Future Works

In this section, suggestions are made for potential future works. The suggestions are meant to make the algorithms more applicable and more efficient.

In this study, an estimation algorithm was implemented to estimate the longitudinal and lateral velocities using the data-driven methods, considering only the local data. All the test and reference data considered for this study are from the road with no grade or bank angle. The estimation can be further expanded and validated by incorporating road bank and grade angles.

The study can be extended to include test data for roads with different friction coefficients and bank angles. Also, Including tests performed on roads with different surface roughness can help toward the extension.

Furthermore, kernel-based algorithms can be used for vehicle parameter estimation. The main estimation performance can be studied in case of having a better knowledge of vehicle parameters.

Due to the time constraints of this project, the algorithm was not implemented on a real vehicle. Studying the algorithm performance on a real vehicle processing unit can be another possible expansion of this work.

References

[1] C. G. Bobier and J. C. Gerdes, "Staying within the nullcline boundary for vehicle envelope control using a sliding surface," *Vehicle System Dynamics*, vol. 51, no. 2, pp. 199–217, 2013.

[2] A. Rezaeian, A. Khajepour, W. Melek, S.-K. Chen, and N. Moshchuk, "Simultaneous vehicle real-time longitudinal and lateral velocity estimation," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 1950–1962, 2016.

[3] L. Imsland, T. A. Johansen, T. I. Fossen, H. F. Grip, J. C. Kalkkuhl, and A. Suissa, "Vehicle velocity estimation using nonlinear observers," *Automatica*, vol. 42, no. 12, pp. 2091–2103, 2006.

[4] J. Farrelly and P. Wellstead, "Estimation of vehicle lateral velocity," in *Proceeding of the 1996 IEEE International Conference on Control Applications IEEE International Conference on Control Applications held together with IEEE International Symposium on Intelligent Contro.* IEEE, 1996, pp. 552–557.

[5] L. Chu, Y. Shi, Y. Zhang, H. Liu, and M. Xu, "Vehicle lateral and longitudinal velocity estimation based on adaptive kalman filter," in *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol. 3. IEEE, 2010, pp. V3–325.

[6] J. Villagra, B. d'Andrea Novel, M. Fliess, and H. Mounier, "Estimation of longitudinal and lateral vehicle velocities: an algebraic approach," in *2008 American control conference.* IEEE, 2008, pp. 3941–3946.

[7] L.-H. Zhao, Z.-Y. Liu, and H. Chen, "Design of a nonlinear observer for vehicle velocity estimation and experiments," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 664–672, 2010.

[8] H. F. Grip, L. Imsland, T. A. Johansen, J. C. Kalkkuhl, and A. Suissa, "Vehicle sideslip estimation," *IEEE control systems magazine*, vol. 29, no. 5, pp. 36–52, 2009.

[9] L. Imsland, H. F. Grip, T. A. Johansen, T. I. Fossen, J. C. Kalkkuhl, and A. Suissa, "Nonlinear observer for vehicle velocity with friction and road bank angle adaptation-validation and comparison with an extended kalman filter," SAE Technical Paper, Tech. Rep., 2007.

[10] A. H. Korayem, A. Khajepour, and B. Fidan, "A review on vehicle-trailer state and parameter estimation," *IEEE Transactions on intelligent transportation systems*, 2021.

[11] G. You, S. Park, and D. Oh, "Real-time state-of-health estimation for electric vehicle batteries: A data-driven approach," *Applied energy*, vol. 176, pp. 92–103, 2016.

[12] Y. Wei, X. Zhang, Y. Shi, L. Xia, S. Pan, J. Wu, M. Han, and X. Zhao, "A review of data-driven approaches for prediction and classification of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 1027–1047, 2018.

[13] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[14] X. Jin, G. Yin, and N. Chen, "Advanced estimation techniques for vehicle system dynamic state: A survey," *Sensors*, vol. 19, no. 19, p. 4289, 2019.

[15] Z. Zhang, C. Sun, R. Bridgelall, and M. Sun, "Application of a machine learning method to evaluate road roughness from connected vehicles," *Journal of Transportation Engineering, Part B: Pavements*, vol. 144, no. 4, p. 04018043, 2018.

[16] H. M. Ngwangwa, P. S. Heyns, F. Labuschagne, and G. K. Kululanga, "Reconstruction of road defects and road roughness classification using vehicle responses with artificial neural networks simulation," *Journal of Terramechanics*, vol. 47, no. 2, pp. 97–111, 2010.

[17] P. Nitsche, R. Stütz, M. Kammer, and P. Maurer, "Comparison of machine learning methods for evaluating pavement roughness based on vehicle response," *Journal of Computing in Civil Engineering*, vol. 28, no. 4, p. 04014015, 2014.

[18] Y. Qin, R. Langari, Z. Wang, C. Xiang, and M. Dong, "Road excitation classification for semi-active suspension system with deep neural networks," *Journal of Intelligent & Fuzzy Systems*, vol. 33, no. 3, pp. 1907–1918, 2017.

[19] T. Zahid, K. Xu, W. Li, C. Li, and H. Li, "State of charge estimation for electric vehicle power battery using advanced machine learning algorithm under diversified drive cycles," *Energy*, vol. 162, pp. 871–882, 2018.

[20] M. A. Patil, P. Tagade, K. S. Hariharan, S. M. Kolake, T. Song, T. Yeo, and S. Doo, "A novel multistage support vector machine based approach for li ion battery remaining useful life estimation," *Applied energy*, vol. 159, pp. 285–297, 2015.

[21] T. Leidinger, "Hybrid powertrain online control using reinforcement learning/submitted by thomas leidinger, bsc," 2021.

[22] Y. L. Murphey, J. Park, Z. Chen, M. L. Kuang, M. A. Masrur, and A. M. Phillips, "Intelligent hybrid vehicle power control—part i: Machine learning of optimal vehicle power," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, pp. 3519–3530, 2012.

[23] Y. L. Murphey, J. Park, L. Kiliaris, M. L. Kuang, M. A. Masrur, A. M. Phillips, and Q. Wang, "Intelligent hybrid vehicle power control—part ii: Online intelligent energy management," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 69–79, 2012.

[24] T. Novi, R. Capitani, and C. Annicchiarico, "An integrated artificial neural network–unscented kalman filter vehicle sideslip angle estimation based on inertial measurement unit measurements," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 233, no. 7, pp. 1864–1878, 2019.

[25] M. Milanese, C. Novara, and I. Gerlero, "Robust estimation of vehicle sideslip angle from variables measured by esc system," in *15. Internationales Stuttgarter Symposium.* Springer, 2015, pp. 1063–1076.

[26] W. Wei, B. Shaoyi, Z. Lanchun, Z. Kai, W. Yongzhi, and H. Weixing, "Vehicle sideslip angle estimation based on general regression neural network," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[27] T. Gräber, S. Lupberger, M. Unterreiner, and D. Schramm, "A hybrid approach to side-slip angle estimation with recurrent neural networks and kinematic vehicle models," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 1, pp. 39–47, 2018.

[28] M. Mangeas, S. Glaser, and V. Dolcemascolo, "Neural networks estimation of truck static weights by fusing weight-in-motion data," in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.(IEEE Cat. No. 02EX5997)*, vol. 1. IEEE, 2002, pp. 456–462.

[29] S. Torabi, M. Wahde, and P. Hartono, "Road grade and vehicle mass estimation for heavy-duty vehicles using feedforward neural networks," in *2019 4th International Conference on Intelligent Transportation Engineering (ICITE)*. IEEE, 2019, pp. 316–321.

[30] E. Paulsson and L. Åsman, "Vehicle mass and road grade estimation using recursive least squares," 2016.

[31] A. Vahidi, A. Stefanopoulou, and H. Peng, "Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments," *Vehicle System Dynamics*, vol. 43, no. 1, pp. 31–55, 2005.

[32] H. Taghavifar, "Neural network autoregressive with exogenous input assisted multi-constraint nonlinear predictive control of autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6293–6304, 2019.

[33] Z. Chu, D. Zhu, and S. X. Yang, "Observer-based adaptive neural network trajectory tracking control for remotely operated vehicle," *IEEE Transactions on Neural networks and learning systems*, vol. 28, no. 7, pp. 1633–1645, 2016.

[34] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 22–35, 2017.

[35] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[36] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 37, no. 7, pp. 984–995, 1989.

[37] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE signal processing magazine*, vol. 13, no. 4, pp. 67–94, 1996.

[38] X. Dong, Y. Jiang, Z. Zhong, W. Zeng, and W. Liu, "An improved rollover index based on bp neural network for hydropneumatic suspension vehicles," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[39] D. R. Woerner, R. Ranganathan, and A. C. Butler, "Developing an artificial neural network for modeling heavy vehicle rollover," *SAE transactions*, pp. 551–561, 2000.

[40] C. Lv, Y. Xing, C. Lu, Y. Liu, H. Guo, H. Gao, and D. Cao, "Hybrid-learning-based classification and quantitative inference of driver braking intensity of an electrified vehicle," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5718–5729, 2018.

[41] S. Blume, P. M. Sieberg, N. Maas, and D. Schramm, "Neural roll angle estimation in a model predictive control system," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1625–1630.

[42] J. Garcia Guzman, L. Prieto Gonzalez, J. Pajares Redondo, M. M. Montalvo Martinez, and M. J. L Boada, "Real-time vehicle roll angle estimation based on neural networks in iot low-cost devices," *Sensors*, vol. 18, no. 7, p. 2188, 2018.

[43] A. Rakotomamonjy, R. Le Riche, D. Gualandris, and Z. Harchaoui, "A comparison of statistical learning approaches for engine torque estimation," *Control Engineering Practice*, vol. 16, no. 1, pp. 43–55, 2008.

[44] M. Nguyen-H and C. Zhou, "Improving gps/ins integration through neural networks," *arXiv preprint arXiv:1005.5115*, 2010.

[45] G. S. Watson, "Smooth regression analysis," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 359–372, 1964.

[46] E. A. Nadaraya, "On estimating regression," *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[47] M. I. Shapiai, Z. Ibrahim, M. Khalid, L. W. Jau, and V. Pavlovich, "A non-linear function approximation from small samples based on nadaraya-watson kernel regression," in *2010 2nd International Conference on Computational Intelligence, Communication Systems and Networks*. IEEE, 2010, pp. 28–32.

[48] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[49] Wikipedia contributors, "Kriging — Wikipedia, the free encyclopedia," 2021, [Online; accessed 8-September-2021]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Kriging&oldid=1043002369

[50] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[51] MATLAB, *9.7.0.1190202 (R2019b)*. Natick, Massachusetts: The MathWorks Inc., 2018.

[52] T. Gasser and H.-G. Müller, "Estimating regression functions and their derivatives by the kernel method," *Scandinavian journal of statistics*, pp. 171–185, 1984.

[53] E. S. Pearson, "The test of significance for the correlation coefficient," *Journal of the American Statistical Association*, vol. 26, no. 174, pp. 128–134, 1931.

[54] F. Galton, "Regression towards mediocrity in hereditary stature." *The Journal of the Anthropological Institute of Great Britain and Ireland*, vol. 15, pp. 246–263, 1886.

[55] T. D. Gillespie, "fundamental of vehicle dynamics, warrendale, pa 15096-0001: Society of automotive engineers," 1992.

[56] E. Hashemi, M. Pirani, A. Khajepour, and A. Kasaiezadeh, "A comprehensive study on the stability analysis of vehicle dynamics with pure/combined-slip tyre models," *Vehicle system dynamics*, vol. 54, no. 12, pp. 1736–1761, 2016.

[57] A. Y. Ungoren, H. Peng, and H. Tseng, "A study on lateral speed estimation methods," *International Journal of Vehicle Autonomous Systems*, vol. 2, no. 1-2, pp. 126–144, 2004.