Multilane Traffic Density Estimation with KDE and Nonlinear LS & Tracking with Scalar Kalman Filtering

A thesis submitted to the
Graduate School of Natural and Applied Sciences

by

Mikail Yılan

in partial fulfillment for the
degree of Master of Science

in

Electronics and Computer Engineering

İSTANBUL
ŞEHİR
ÜNIVERSITY

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Electronics and Computer Engineering.

APPROVED BY:

Assist. Prof. Mehmet Kemal Özdemir ....
(Thesis Advisor)

Prof. Ahmet Ademoğlu ....

Assist. Prof. Murat Küçükvar ....

This is to confirm that this thesis complies with all the standards set by the Graduate School of Natural and Applied Sciences of İstanbul Şehir University:

DATE OF APPROVAL: 17 August 2016

SEAL/SIGNATURE:

Declaration of Authorship

I, Mikail Yilan, declare that this thesis titled, 'Multilane Traffic Density Estimation with KDE and Nonlinear LS & Tracking with Scalar Kalman Filtering' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date: 17 August 2016

Multilane Traffic Density Estimation with KDE and Nonlinear LS & Tracking with Scalar Kalman Filtering

Mikail YILAN

Abstract

With increasing population, the determination of traffic density becomes very critical in managing the urban city roads for safer driving and low carbon emission. In this study, Kernel Density Estimation is utilized in order to estimate the traffic density more accurately when the speeds of the vehicles are available for a given region. For the proposed approach, as a first step, the probability density function of the speed data is modeled by Kernel Density Estimation. Then, the speed centers from the density function are modeled as clusters. The cumulative distribution function of the speed data is then determined by Kolmogorov-Smirnov Test, whose complexity is less when compared to the other techniques and whose robustness is high when outliers exist. Then, the mean values of clusters are estimated from the smoothed density function of the distribution function, followed by a peak detection algorithm. The estimation of variance values and kernel weights, on the other hand, are found by a nonlinear Least Square approach. As the estimation problem has linear and non-linear components, the nonlinear Least Square with separation of parameters approach is adopted, instead of dealing with a high complexity nonlinear equation. Finally, the tracking of former and latter estimations of a road is calculated by using Scalar Kalman Filtering with scalar state - scalar observation generality level. Simulations are carried out in order to assess the performance of the proposed approach. For all example data sets, the minimum mean square error of kernel weights is found to be less than 0.002 while error of mean values is found to be less than 0.261. The proposed approach was also applied to real data from sample road traffic, and the speed center and the variance was accurately estimated. By using the proposed approach, accurate traffic density estimation is realized, providing extra information to the municipalities for better planning of their cities.

Keywords: Traffic Density Estimation, Kernel Density Estimation, Kolmogorov-Smirnov Tests, Nonlinear Least Square, Newton-Raphson, Tracking, Scalar Kalman Filtering.

Çekirdek Yoğunluk Kestirimi ve Lineer olmayan En Küçük Kareler Yöntemi ile Çok Şeritli Trafik Yoğunluğu Kestirimi & Skalar Kalman Süzgeçleme ile İzleme ve Takip

Mikail Yılan

Öz

Kentsel bölgelerde populasyon artışıyla birlikte, yolların daha iyi yönetimi, güvenli sürüş ve düşük karbon emisyonu amacıyla yollardaki trafik yoğunluğunu bilmek çok büyük öneme sahip hale gelmiştir. Bu çalışmada, belirli bölgede araçların hızları veri olarak mevcutken, Çekirdek Yoğunluk Kestirimi trafik yoğunluğunu daha doğru kestirmek için kullanılmıştır. Önerilen yaklaşım için, ilk olarak, Çekirdek Yoğunluk Kestirimi tekniği hız verisinin olasılık yoğunluk fonksiyonunu elde etmek için uygulandıktan sonra hız gruplarının kümeleri ortalanmıştır. Öte yandan, hız verisinin yığmalı dağılım fonksiyonu Kolmogorov-Simirnov Sınaması ile bulunmuştur. Bu sınama diğer tekniklerle mukayese edildiğinde, daha az karmaşık ve aykırı değer olduğu durumlarda daha dayanıklıdır. Ardından, kümelerin beklenen değerleri zirve belirleme algoritmaları ile kestirilmiştir. Varyans değerlerinin ve çekirdek ağırlıklarının kestirimleri ise doğrusal olmayan En Küçük Kareler yaklaşımı ile bulunmuştur. Kestirim probleminin doğrusal olan ve doğrusal olmayan iki bileşeni olduğundan, çok karmaşık lineer olmayan denklemlerle uğraşmak yerine, doğrusal olmayan En Küçük Kareler parametrelerin ayrılması ile özelliği ile tatbik edilmiştir. Son olarak, bir yolun önceki ve sonraki kestirimlerinin izlemesi Skalar Kalman Filtresi'nin skalar durum skalar gözlem genelleştirme seviyesi kullanılarak hesaplanmıştır. Önerilen yaklaşımın performansını değerlendirmek amacıyla simülasyonlar ortaya konulmuştur. Bütün örnek veri setleri için, çekirdek yoğunluklarının minimum ortalama karesel hatası 0.002'den daha az bulunurken, ortalama değerlerin hatası 0.261'den daha az bulunmuştur. Önerilen yaklaşım, örneklem bir yoldan alınan gerçek verilerle de test edilmiştir ve hız ortalamaları ve varyanslar doğru kestirilmiştir. Bu çalışmada ortaya koyulan yaklaşım kullanılarak, belediyelere şehirlerini daha iyi planlamaları için ekstra bilgi sağlayan doğru trafik yoğunluk kestirimi gerçekleştirilmiştir.

Anahtar Sözcükler: Trafik Yoğunluk Kestirimi, Çekirdek Yoğunluk Kestirimi, Kolmogorov -Smirnov Sınaması, Doğrusal Olmayan En Küçük Kareler, Newton-Raphson, İzleme, Skalar Kalman Filtresi.

This thesis is dedicated to my mother Perihan and my father Esat for their love, encouragement, and endless support throughout my life.

Acknowledgments

I would like to express my most valuable appreciation to my advisor Asst. Prof. Mehmet Kemal Özdemir for his support, contribution, patience, and guidance. Without his patience and sacrificing hours for me, this thesis will never be written.

Special thanks to Prof. Ahmet Ademoğlu and Assist. Prof. Murat Küçükvar for being committee members of my thesis.

I am grateful to Kevser Bayraktar for introducing İstanbul Şehir University to me and for her advices and encouragement at decision making process to get a master's degree.

Finally, I wish to thank my parents, my sisters, and my friends for their support throughout my study.

Contents

List of Figures

List of Tables

Abbreviations

CDF	**C**umulative **D**istribution **F**unction
CRLB	**C**ramer-**R**ao **L**ower **B**ound
CvM	**C**ramer-**v**on **M**ises
DS-CDMA	**D**irect **S**equence - **C**ode **D**ivision **M**ultiple **A**ccess
eCDF	**e**mpirical **C**umulative **D**istribution **F**unction
EM	**E**xpectation **M**aximization
KDE	**K**ernel **D**ensity **E**stimation
KF	**K**alman **F**ilter
KS	**K**olmogorov-**S**mirnov
LS	**L**east **S**quares
MAI	**M**ulti**a**ccess **I**nterference
MSCEE	Normalized **M**ean **S**quare **C**hannel **E**stimation **E**rror
ML	**M**aximum **L**ikelihood
MMSE	**M**inimum **M**ean **S**quare **E**rror
NMSE	**N**ormalized **M**ean **S**quare Channel Estimation **E**rror
N-R	**N**ewton-**R**aphson
NLS	**N**onlinear **L**east **S**quares
PDA	**P**eak **D**etection **A**lgorithm
PDF	**P**robability **D**ensity **F**unction
WGN	**W**hite **G**aussian **N**oise

Chapter 1

Introduction

1.1 Motivation

Estimating the traffic density and predicting it for a specified time interval, especially in mega cities, has become an urgent issue that needs to be tackled down. Traffic issue leads the social, economical and environmental damages. If we consider traffic problem in İstanbul, it has the second most congested traffic [5] and the most sudden-stopping traffic in the world [6]. Mega cities like İstanbul need to deal with the traffic problem. Because sudden-stopping and congested traffic make people more stressful, driving more costly and air pollution worse.

Estimation and prediction of the traffic density is necessary to prevent citizens from congested traffic. An efficient tool providing the traffic density estimation and its prediction will help the municipalities to better manage the traffic and plan their infrastructure. If the decision makers have the knowledge of the current and future traffic reports, municipalities would come up with better solutions against the traffic problem. Moreover, in their daily lives, drivers would make relatively better decisions, thereby directly having impacts on gas and time savings, as well as low carbon emission.

Although there are lots of studies about traffic density estimation, multilane traffic density estimation has been becoming crucial as well. Knowing densities of different lanes in the same road might provide above-mentioned benefits for today's congested roads. The increase in the number of cars in traffic and frequent occurrence of different densities in different lanes are main reasons to look for a solution for multilane traffic density estimation. We can observe an example density difference from Figure 1.1 which is taken from British Cycling Federation [1]. Here, the densities of vehicles in the right lane and left lane of going upward roadway are different.

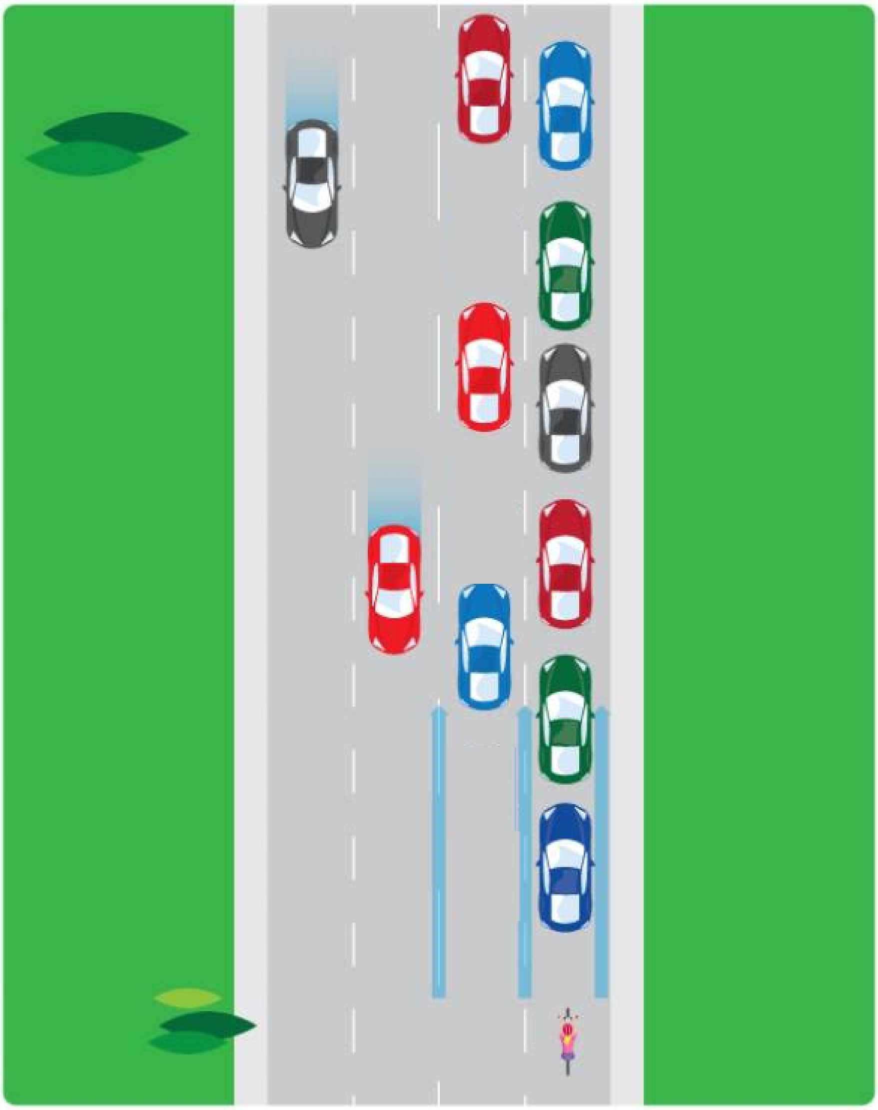


FIGURE 1.1: Multilane Traffic Flow Example [1]

Moreover, the importance of the multilane traffic density estimation can be seen from Figure 1.2 [2]. This traffic density map of İstanbul is taken from Yandex. There are three lanes in Kısıklı Street and then all lanes go different streets. The left lane goes to Kadıköy, the right one goes to Üsküdar and the central one also goes to Üsküdar but from different way. There are three branches leaving from Kısıklı Street and all have different colors in the map according to their speed. The speed of left branch is 10 km/h, the speed of central branch is 35 km/h, and the speed of left branch is 25 km/h.

However, when all three brunches were joint in the street just before leaving, the street has only one color and speed value that is 25 km/h in the map. As can be seen from the map, there is a direct relation from branches to the street and it might lead a delay in corresponding lane. In other words if the vehicles in the left brunch get stuck in the traffic, it also affects mostly left lane in the street. Therefore, traffic densities of different lanes in the street would be different.

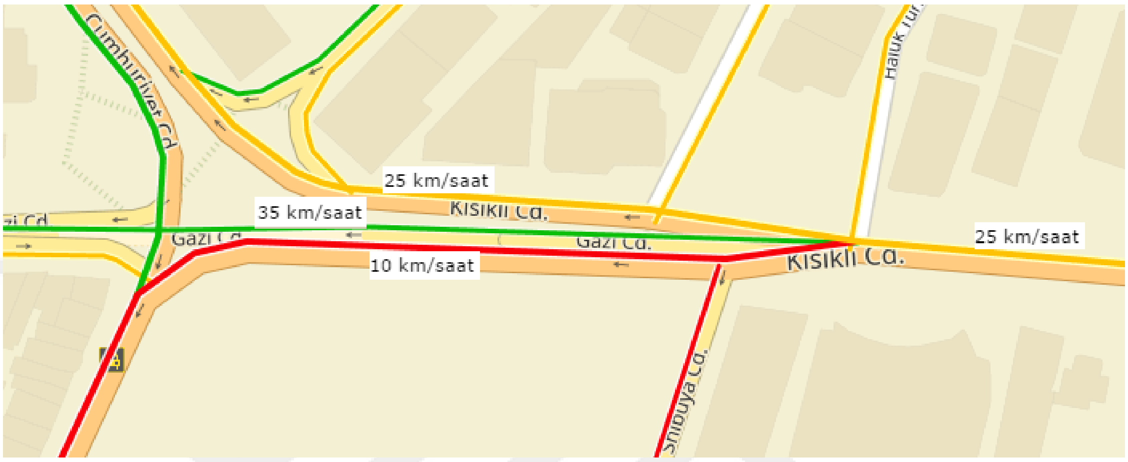


Figure 1.2: Multilane Traffic Density Map Example [2]

1.2 Methods to Find Probability Density Function and Cumulative Distribution Function

Each sample of the received data from the field can be treated as Normally Distributed, and the many realizations of the Normal Distributions are generated. The probability density function (PDF) of the received data can then be obtained either by bin allocation or histogram. Bin allocation involves splitting data into bins with certain step size. With this method, PDF value varies with the speed of vehicles on the road. As the number of vehicles with a velocity in the specified bin increases, the PDF value also increases. Alternative to this approach, Kernel Density Estimation (KDE) can be used to represent diverse traffic scenarios more explicitly and the estimation process via KDE is not too complicated since it makes use of the Gaussian Distribution. In this study, the system is modeled according to KDE, so it does not only determine PDF but also contributes the model of this study.

After determining the PDF, the next step would be to find the cumulative distribution function (CDF). There are numerous methods like Cramer-von Mises (CvM) and Kolmogorov-Smirnov (KS) tests to get the CDF. Since KS test is less affected by the existence of outliers [7], in this study, it is preferred to find CDF.

1.3 Traffic Density Estimation with Kernel Density Estimation

There are numerous ways to estimate the traffic density. However, Kernel Density Estimation, which is also known as Parzen Window Method [8], is one of the most effective approaches, because KDE can better differentiate between the cars that travel on the same road but with different speeds on different lanes [9]. Moreover, estimation approaches can be divided into two groups as parametric, which has a fixed number of parameters, and non-parametric, which has an increasing number of parameters as training data size becomes larger[10]. As we are dealing with continuously incoming data from the field, the latter, the non-parametric approach, is used with KDE. It is used because the combination is a better fit to the current problem, not to mention the mathematical advantages offered by the KDE [11].

In Statistics, KDE is a non-parametric method to estimate the PDF of a random variable. KDE is a fundamental data smoothing method that will be used to estimate mean values with peak detection algorithm. Emanuel Parzen and Murray Rosenblatt improved its widely known version and this approach is also used in Signal Processing [12] - [13]. Although this approach is invented for Statistics, it might be used in other disciplines. It is easily seen from this study that a method from signal processing can be applied to another field like traffic density estimation. Before this study, there was only one study [11] that used KDE to estimate roadway traffic density. Brief explanation of [11] will be given in Section 2.2 and its comparison between current study will be made in Section 2.3.

There are also some traffic density estimation related studies with KDE. Before going any further, it is better to explain these studies briefly. [14] evaluates Gaussian Mixture Model and the adaptive KDE for statistical anomaly detection in sea traffic. The normalcy modeling evaluation reaches an important result that KDE captures details of normal data, more accurately. However, according to anomaly detection results, there is no significant difference between the two approaches and the performances of both methods are suboptimal [14].

Another study with KDE [15] analyzes traffic accidents data in time and space. For the analysis process, it presents a spatio-temporal analysis method based on KDE to extract the time and location patterns of traffic accidents.

The model of [16] has two parts: traffic condition estimation and real-time routing algorithm. Based on location and speed information collected via sensors in the vehicles, the traffic condition is quantified by using a weighted KDE function. The combination of this estimation and road network leads a real-time routing to drivers [16].

1.4 The Approaches for Determination of Key Parameters of Traffic Density Estimation

For the estimation of the variances and the mean values, instead of linear Least Squares (LS) which is used in the beginning of the study [17], nonlinear LS (NLS) with separation of parameters approach is exploited. After determination of empirical CDF (eCDF) with KS test, the straightforward way of computing kernel weights is to take the derivative of the PDF. However, if the received data contains noise, it might not give desired outcome. Hence, instead of using derivatives of PDF, NLS approach is utilized.

Before the estimation of parameters via NLS, the mean values are estimated first by using a peak detection algorithm (PDA) over the smoothed version of the PDF curve. In order to estimate variance values and kernel weights, NLS is used successively. First, speed center's bandwidth and then its kernel weight are estimated. After these estimations, the next speed center's variance and its kernel weight are estimated, and so on.

In applying the NLS, two methods are exploited: linear search and Newton-Raphson (N-R) methods. While the former method generally gives the desired results in a longer time, the latter one gives sufficiently close results in considerably shorter amount of time [18]. Both methods are used to apply NLS approach successively and their results are matched with each other. In so doing, verification of the model is also done in addition to closeness of the estimated values to assumed and real example values as shown in Chapter 4.

1.5 Tracking between Estimated Data and New Data

For the same road, if new data arrives, estimating all data including old data is not efficient because of the computational heaviness and does not represent traffic properly due to the difference in the importance of new and old data. Together with the already existing data that is used for the estimation, we adopt the tracking of the estimated parameters are adopted, instead of using the new large data set and recalculating all the parameters needed to make a fresh estimate.

In accomplishing tracking part of the model, first an initial estimate is obtained from the new and small sized data. Then, to get an overall estimate, the old and the new estimations are combined. Moreover, a forgetting factor is utilized for the tracking of the old and the new estimates so that more weight is given to the newly arrived data rather than updating them according to their number of samples.

A scalar Kalman filter is applied to traffic density estimation in [19]. Present detector data in a 500-metre section of the Boulevard Peripherique in Paris is used to test the accuracy of proposed model in the study. Test results show well tracking capabilities and robustness against the change of application situations.

1.6 Thesis Outline

In the following chapter, Chapter II, historical background and literature review will be mentioned. In Chapter III, the simulation model will be presented. In Chapter IV, the estimation process will be illustrated with numerical calculations. In Chapter V, the tracking part of the model will be tested with different examples. In Chapter VI, the proposed model and its practical results will be assessed. Finally, in Chapter VII, a summary of conclusions, observations, and future work will be presented.

Chapter 2

Literature Review

2.1 Methodologies Used for Estimation of Traffic Density

In this section, some of the common studies about highway traffic density estimation will be mentioned. This thesis could not contain all articles about traffic density estimation methods. However, in order to give a general idea about this kind of studies, some articles and their methodologies will be briefly explained.

In [20] examines computer vision techniques for automatic video analysis from urban surveillance cameras. Since the development of technology, camera deployment and their quality in urban road networks has increased. Therefore, 3-D camera based models have become more popular, recently. In the study [20], a review of computer vision techniques used in traffic analysis systems is presented to analyze the underlying algorithms and methods. The authors also come up with their own method for traffic density estimation.

A similar study [21] uses airborne cameras or cameras located at very high positions to get the traffic flow of a large area. It derives traffic parameters on the basis of single car measurements by using image time series from these cameras. Also, robust road detection and tracking method based on a condensation particle filter is proposed in [22]. [23], [24], and [25] are also well-known and commonly applied models for traffic monitoring systems.

Cell transmission model is used to derive switching mode model (macroscopic traffic flow model) in [26]. The traffic model is applied to estimate traffic densities at unmonitored locations. Different data sets of linear difference equations depends on mainline boundary data and congestion status of the cells in a highway sections. The model is a hybrid system that switches between these sets.

Gaussian mixture variant cardinalized probability hypothesis density filter is extended by employing digital road maps for road constraint targets in [27]. The model is mainly proposed for tracking case in traffic. This extension pave the way for more precise tracking, more reliable target number estimation and more stable results under Doppler targets.

Recursive traffic density estimation by using loop detectors' information is proposed in [28]. Lane-change effect is considered in traffic modeling. Also, Markov chain is used in the state space model to describe lane-change behavior. Finally, estimation is updated with Kalman filter as it will be used in the current study.

Road acoustics is another method to collect data from the field. Cumulative acoustic signal based information cues are utilized in [29]. Short-term spectral envelope features of such signals are extracted and their class-conditional probability distributions are modeled based on the speed values in the study. Traffic densities are estimated according to these determined classes by using acoustic signals. Bayes' classifier is further used to classify the acoustic signal segments spanning a duration of 5-30 seconds.

Data in the [30] is taken from the cumulative acoustic signal acquired from a roadside-installed single microphone. Adaptive Neuro-Fuzzy classifier is used to model the traffic density state based on the velocities of the vehicles and to classify the acoustic signal segments spanning duration of 20-40 seconds.

A density estimation method for signalised arterials based on cumulative counts from upstream and downstream detectors is proposed in [31]. Cumulative plots and probe integration for travel time estimation is executed for traffic density estimation to deal with counting errors. Stop line detectors and probe data are the main source of the taken information from the field.

There are also other techniques to collect data from roads, like taking information from users of navigation applications. Also there are lots of ways to estimate traffic density after the collection of real-time data. However, reliable real-time data and accurate traffic density estimation are not easy tasks. Therefore, methods to achieve these goals are crucial.

2.2 An Example Study of Traffic Density Estimation with KDE and CvM

The traffic data can be modeled as clusters, and each traffic cluster is represented by a mean, a bandwidth, and a coefficient that shows cluster's weight among all present clusters. Traffic density estimation via KDE has been studied in [11] by using CvM test for the determination of the kernel weights. However, the study in [11] assumes the variances and mean values of the traffic clusters as constants.

In [11], PDF values of Dirac Distribution and Gaussian Mixtures are compared by using CvM and localized cumulative distributions. Then, optimized kernel weights are estimated by minimizing the distance between floating cars' position centers and point-of-interests through quadratic programming. The idea of kernel-based optimization method to minimize the distance between localized cumulative distribution is taken from [32], then applied to detect congestion areas [11].

To sum up, [11] has tested the problem of congestion detection in roadway traffic and come up with a kernel-based density optimization approach to solve this problem. However, the performance of the model rely on lots of parameters like bandwidth of speed centers, mean values, trade-off factor on regularization. Solutions when these parameters are unknown have not been proposed.

2.3 Three Complementary Studies for Traffic Density Estimation and Tracking

Similar to [11], in the beginning of this study, the bandwidths and the speed centers are also counted as constants. The methodology of [17] is an alternative to [11] since it can determine the kernel weights more easily by using KDE, KS Test and linear LS approaches in fewer steps. The approach used in [11] has been published in IEEE 23th SIU Conference in 2015.

The initial study [17] has been extended in [33] by assuming that the variances and the mean values of the traffic clusters are also non-constants like kernel weights. This, however, requires the estimation of the mean values and the variances of each cluster. The parameter that is fixed by the approach is the number of the traffic clusters. However, if more clusters than the existing amount is made, then the kernel weights are simply found to be close to zero.

In [33], in the first step of estimations, a PDA over the smoothed version of the PDF was utilized for the estimation of mean values. This algorithm detects all data that constructs PDF and finds peak values, these peak values correspond to speed centers in clusters. NLS with separation of parameters approach was applied successively to estimate variance values and kernel weights. Since Gaussian Distribution is nonlinear but kernel weights are linear in the equation, first a mean value's bandwidth determined, then its kernel weight is estimated. This part of the project is accepted to publish in Turkish Journal of Electrical Engineering and Computer Sciences in 2016 [33].

In the complementary study of former two studies, in [34], tracking part of the work is introduced. In the tracking part, the tracking of former data estimations and newly coming data estimations of a road is calculated by using Scalar Kalman Filtering with scalar state - scalar observation generality level. Tracking makes the model simpler for the calculations when new data enters the system. Because it is not logical to calculate estimation of already estimated data again and again. Instead, new data is estimated and estimation of overall data is found by using old estimation and new estimation. The last part of the project is accepted to publish in IU-Journal of Electrical & Electronics Engineering and will be published Vol 16, No 2 in 2016 [34].

2.4 Comparison of Three Different Nonlinear Estimation Techniques on the Same Problem

Maximum Likelihood (ML) Approach, ML with Expectation Maximization (EM) Algorithm, and Nonlinear Least Square Estimation with separation of parameters techniques will be compared for the same problem. Channel estimation for Direct Sequence - Code Division Multiple Access (DS-CDMA) System is an important problem and this channel parameters were estimated by using above three methods.

DS-CDMA is a well-known multiplexing technique, in this technique, several users can access simultaneously to the same frequency band by modulating preassigned signature waveforms [35]. However, in order to use such multiplexing technique, multipath channel parameters should be estimated accurately. Since the system was adopted to 3G mobile cellular networks, different studies have been done with different approaches.

Before going further, let's introduce three approaches briefly. To begin with ML Estimation, it is one of the most popular approaches to obtain practical estimators. We can define it as the value of θ that maximizes the likelihood function. When the PDF is counted as a function of the unknown parameter, it is termed as the likelihood function. If a closed form expression cannot be found for the ML Estimation, a numerical approach is implemented by using either a grid search or an iterative maximization of the likelihood function (Newton-Raphson and Scoring Methods). However, as we will see in the [36] convergence to the ML Estimation is not guaranteed [18].

Another ML based approach is the Expectation Maximization. Although EM is iterative in nature, it is guaranteed under certain mild conditions to converge, and at the convergence to produce at least local maximum.The EM has the property of increasing the likelihood at each iteration. The EM needs to have two steps, first is expectation which is to determine the average log-likelihood of the complete data, the other one is maximization which is to maximize average log-likelihood function of the complete data [18].

The main idea for the Least Square Approach is to minimize the squared difference between the given data and the assumed signal or noiseless data. This method is widely used in practice due to its ease of implementation, amounting to the minimization of a least square error criterion. When we come to NLS Estimation, its determination is based on iterative approaches and suffers from the same limitations as ML Estimation determination with numerical methods. There are two ways to reduce complexity of NLS, transformation of parameters and separability of parameters which is also used in [4]

2.4.1 A Maximum Likelihood Approach for Estimating DS-CDMA Multipath Fading Channels

In the paper [36], a ML approach for near-far robust synchronization of asynchronous DS-CDMA systems operating over multipath channels is proposed. Near-far problem occurs when received signals from the concurrent users are either non-orthogonal or very dissimilar in power. Maintaining orthogonality in asynchronous system is too difficult, on the other hand strict power control is easier when the channel is fading rapidly. Therefore, in [36], near-far resistant methods for synchronization and channel estimation is studied on the power issue.

The used process for the [36] is that first formulation a ML Method for estimating all parameters in a multiuser multipath static channel, then modifying the estimator to estimate only the parameters for one user, and finally evaluating the resulting estimator for the case when several users are active and when the channel is allowed to be time variant.

The main problem with the ML Estimator is that it requires a numerical search over a large number of parameters. The search is thus computationally expensive and might converge to a local minimum. In the paper, the performance of the ML-based algorithm is made better by decreasing the number of parameters in the search. Also, in order to get rid of from local minimum and to reach global minimum, good initial guess is so important for this method.

Finally, proposed algorithms of this work perform well in a near-far environment and relatively close to the Cramer-Rao Lower Bound (CRLB). CRLB is the best unbiased estimator, where the mean of the estimator is equal to estimated value's mean ($E\left(\hat{\theta}\right) = \theta$), if it is attainable.

2.4.2 Channel Estimation for the Uplink of a DS-CDMA System

In the paper [3], channel estimation in the of a DS-CDMA system which operates at multipath environment is made by using ML based Expectation-Maximization algorithm. As mentioned above, one of the main problem in estimating channel parameter is that signals transmitted by different users are asynchronous and not truly orthogonal. In contrast to [36], this paper deals with maintaining orthogonality problem. The problem generates multiaccess interference (MAI) at the base station, therefore accuracy of the channel esimations are limited.

In this study, MAI is counted as colored Gaussian noise and the inverse of its covariance matrix is used as a whitening filter to mitigate the near - far effect. In computing the MAI covariance matrix, it is assumed that the interfering users have already been acquired and their channel parameters are known at the receiver.

Here, the main enhancement is that EM algorithm is used to decompose a multidimensional maximization problem into a sequence of one-dimensional searches. The proposed channel estimator in this work is compatible with the universal mobile telecommunication system recommendations and operates in an iterative fashion according to the space-alternating generalized EM algorithm. At each step, the parameters of a given path are estimated and the signal contribution of that path is canceled out from the received signal.

Compared with the ML estimator (like [36]), it is much simpler to implement as it reduces a maximization problem with many parameters to a succession of simple one-dimensional searches. Also, the channel parameters when a new user enters the system are estimated well. As we said in [36], the channel gains and delays can be measured with accuracy close to the CRLB even in the presence of a strong MAI. Normalized mean square channel estimation error (MSCEE) versus E_b/N_0 energy per bit to noise power spectral density ratio is illustrated in Figure 2.1. Performance losses come from to imperfect knowledge of the MAI covariance matrix.

2.4.3 A Robust Method for Estimating Multipath Channel Parameters in the Uplink of a DS-CDMA System

In the paper [4], the estimation of multipath channel parameters in the uplink of a DS-CDMA system is described by using nonlinear least squares cost function. The function is nonlinear with respect to the time delays and linear with respect to gains of the multipath channel. Therefore, separation of parameters property of NLS is used.

Since the uplink of a DS-CDMA system is asynchronous, MAI is again used in this work. The most difficult part of this procedure is the maximization problem. For this

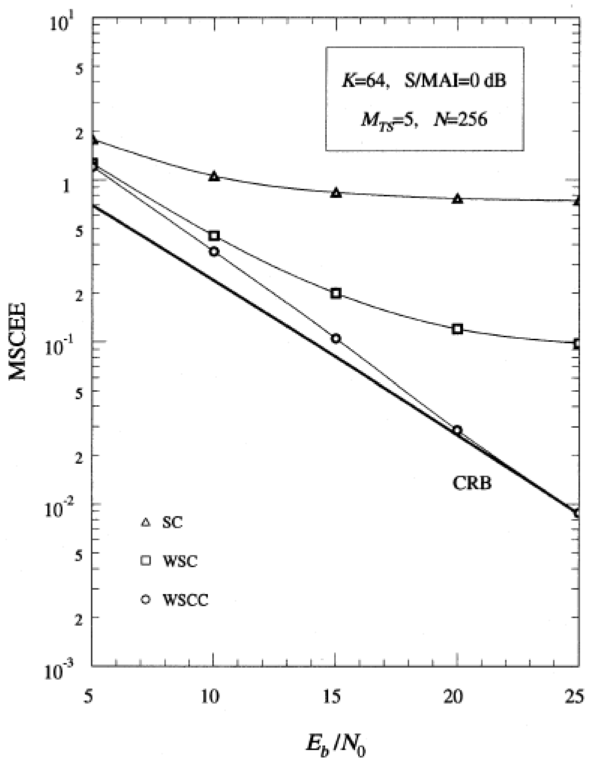


FIGURE 2.1: MSCEE versus E_b/N_0 with ML-EM Approach [3]

problem, performing a costly multidimensional search or applying an iterative Newton type method which can reach a local maximum are two common solution. Instead of using these two techniques, due to their drawbacks, the MAI covariance matrix is forced to become similar to identity matrix. Then the procedure have become easier.

All in all, simulations results show that the proposed method [4] exhibits a lower mean squared estimation error than the method of [3], at the expense of a negligible increase of the computational complexity. Figure 2.2 shows the enhancement when compared with Figure 2.1. Note that here NMSE is the same as MSCEE, both are normalized mean square channel estimation error with different abbreviations. Therefore, NLS estimation with separability of parameters property gives better results than both ML-based EM algorithm [3] and ML estimation techniques [36]. That's why NLS with separation of parameters approach is chosen in this study.

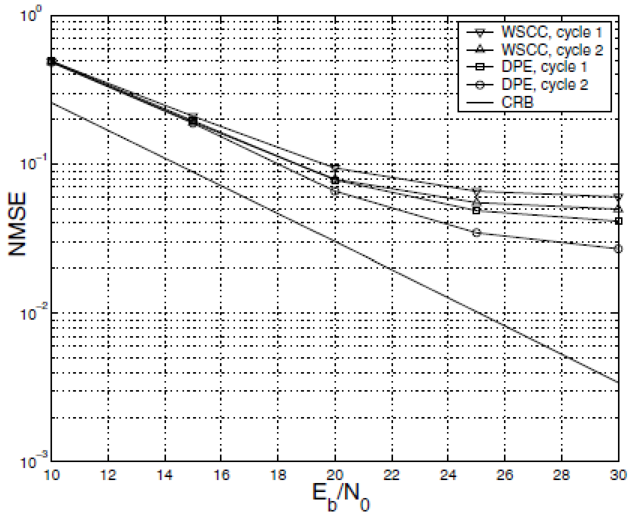


FIGURE 2.2: NMSE versus E_b/N_0 with NLS Approach [4]

Chapter 3

The Model

In this chapter, the model will be explained in detail. Figure 3.1 presents an illustration of the steps of the proposed method. Sections follow the procedure of the model. In the first section how to determine PDF values with KDE will be explained. Then, how to decide eCDF values with KS Test will be described. After determination of PDF and CDF, finding mean values with PDA will clarified. Afterwards, estimation of variance values and kernel weights with separability of parameters property of NLS Method will formulated. In the last section, tracking of the traffic density estimation with Scalar Kalman Filter, scalar state - scalar observation generality level will be explained. Also, two algorithms of estimation and tracking parts of the model will be given.

3.1 Finding Density Distribution with KDE

For a given N independent samples, let $\mathbf{x} \equiv \{X_1, ..., X_N\}$ comes from a continuous PDF f, which is defined on X. Gauss KDE can then be defined as follows [37]:

When the mean of each data sample is X_i and the corresponding variance is σ, then the Gauss Kernel PDF is

$$\hat{f}(x;\sigma) = \frac{1}{N} \sum_{i=1}^{N} \phi(x, X_i; \sigma), \qquad x \in \mathbb{R}, \tag{3.1}$$

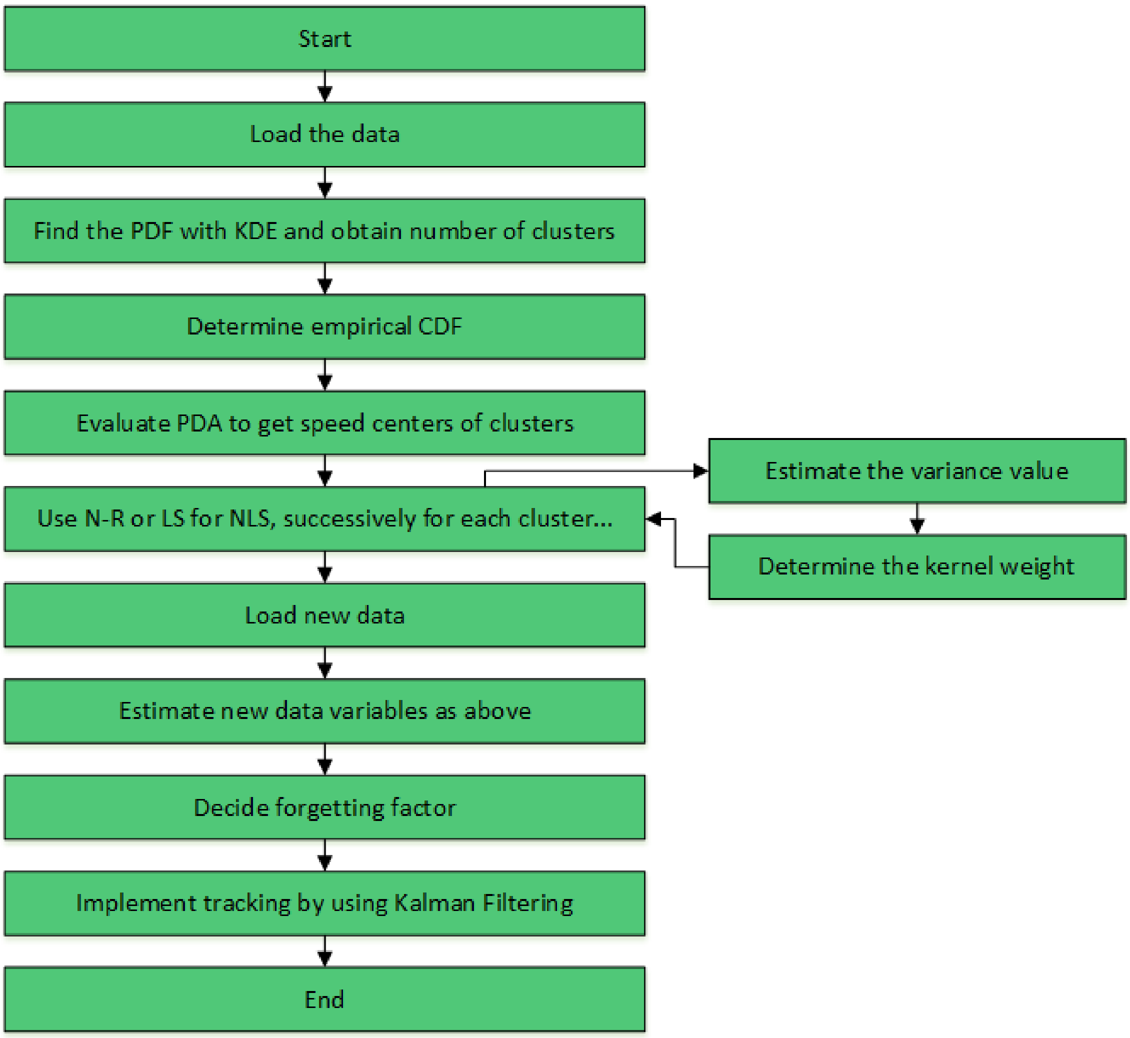


FIGURE 3.1: The steps of the model

where

$$\phi\left(x, X_i; \sigma\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-X_i)^2/(2\sigma)}. \tag{3.2}$$

When the contribution of the Gauss Kernels are different, then (3.1), can be re-expressed by including kernel weights, α_i as:

$$\hat{f}\left(x; \sigma, \alpha\right) = \frac{1}{N} \sum_{i=1}^{N} \alpha_i \phi\left(x, X_i; \sigma\right), \qquad x \in \mathbb{R}, \tag{3.3}$$

where

$$0 \leq \alpha_i \leq 1 \quad \text{and} \quad \sum \alpha_i = 1 \quad . \tag{3.4}$$

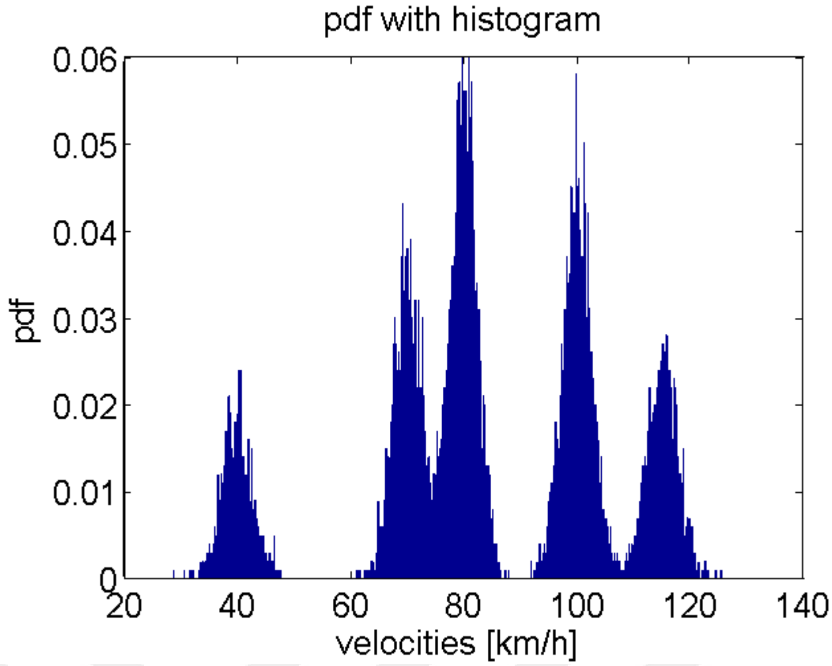


FIGURE 3.2: PDF plot via histogram

For a given set of data, PDF plots can be found in two ways: either through plotting or a closed-form expression. One of the easiest methods of plotting PDF is to use a histogram plot as seen in Figure 3.2. Alternatively, bin allocation is another method for PDF plot without using a closed-form expression as seen in Figure 3.3. However, these methods give only figures, and hence do not provide closed form expressions that can be utilized for further theoretical developments. On the other hand, if KDE is used to determine the PDF plots, such plots will be more convenient to visualize as in Figure 3.4 and will provide a better theoretical background for the estimation studies [37]. Since one of the main goals of KDE is to produce a smooth density surface over a 2-D geographical space, the smoothed version of the density is used to plot PDF with KDE [38].

3.2 Finding Empirical CDF with KS Test

For one sample test in KS Test, definition of the CDF and the hypothesis can be developed as follows: $X_1, X_2, \ldots, X_n$ are random variables for $x_1, x_2, \ldots, x_n$ continuous, independent and identically distributed (i.i.d) samples. When F is defined as CDF, $\hat{F}$ would be empirical CDF [39].

With F_0 is known, for all $x \in \mathbb{R}$ values, the hypothesis:

$$H_0 : F(x) = F_0(x) \tag{3.5}$$

$$\hat{F} = \frac{\#\left(i : x_i \leq x\right)}{n} \tag{3.6}$$

is empirical CDF. Then, the KS Test statistics D_ns are defined as follows:

$$D_n = \sup_x \left| \hat{F}\left(x\right) - F_0\left(x\right) \right| \tag{3.7}$$

where *sup* is the supremum of the CDF values.

KS Test initially examines the difference between the empirical and the real values, and then checks how much they match up with each other. For this model, the distribution of the speeds can be obtained and checked if they have normal distribution or not. The expression in (3.6) arranges all the data in order according to their values, and then rescales them, and finally when the biggest value is encountered, the CDF reaches to the unity. This approach can be criticized, as the number of operation would be huge when the number of samples are very large. However, this problem can be overcome by efficient and fast computer programs like MATLAB and high performing processors.

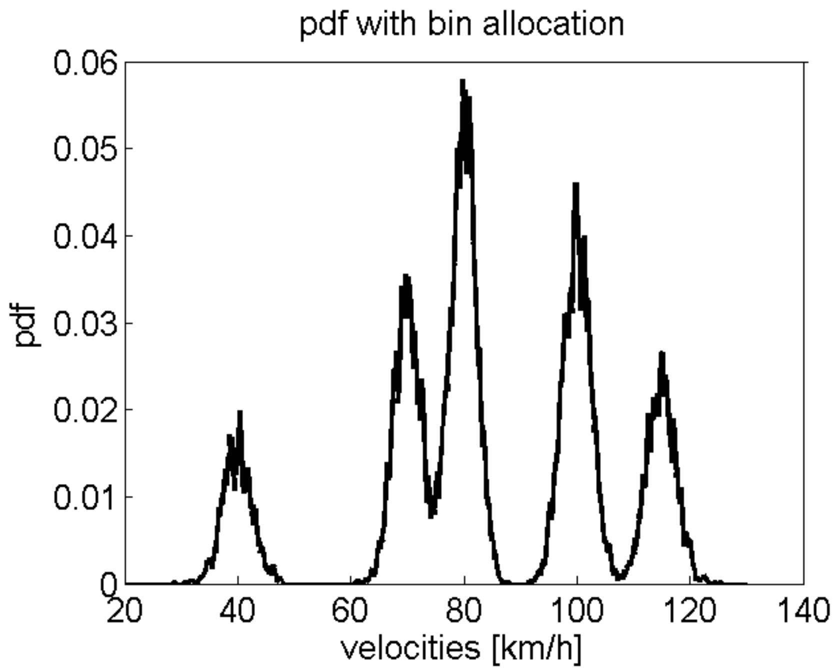


Figure 3.3: PDF plot via bin allocation

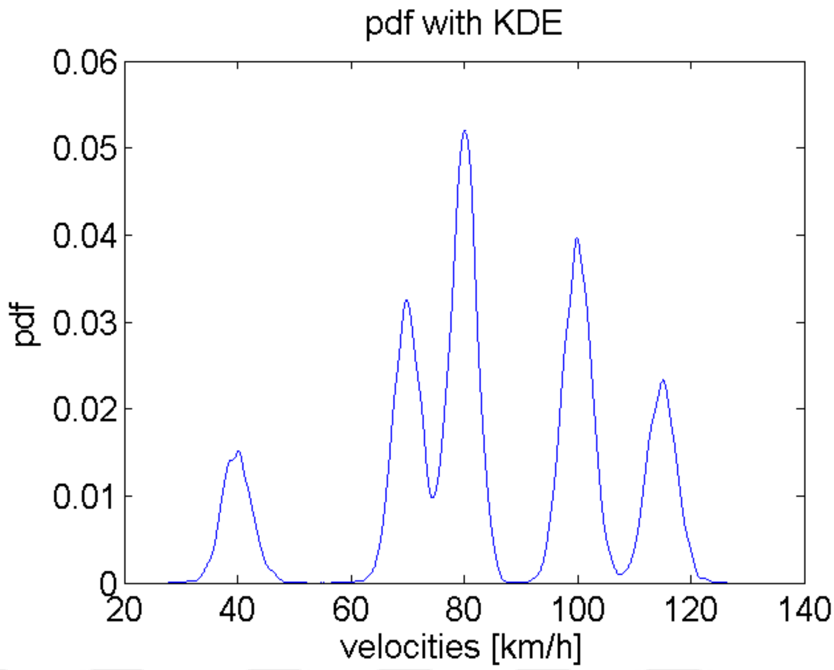


FIGURE 3.4: PDF plot with KDE

3.3 Determination of Speed Centers via PDA

The mean values of the speed clusters resemble separate regions for the PDF. Hence, first of all they should be determined. By using a peak detection algorithm applied to the PDF, mean values can be determined straightforwardly.

Before we start to use PDA algorithm, the derivative of CDF is determined, yielding the PDF, since its derivative is safe and is an easy way to find PDF values, statistically. Then, through a smoothing operation, the PDF is prepared for a peak detection algorithm, which then produces accurate speed center values.

Peak detection algorithm checks all data in data set, and finds peak values. From the peak detection algorithm, we obtain the corresponding mean values of the peak values in PDF. As seen in Figures 3.2, 3.3, and 3.4, peak values in PDF plots correspond to speed centers.

3.4 Estimation of Variance and Kernel Weights with Nonlinear LS Method

The next step is to find both variance values and kernel weights by using separability of parameters property of NLS. In the model, which is defined in (3.3), α is linear and σ is nonlinear with respect to the system. Then, the approach is to find LS for parameters α in terms of the σ, and then to determine σ. For this, the following equation (LS error) should be minimized for α [18];

$$J(\sigma, \alpha) = (x - H(\sigma)\,\alpha)^T\,(x - H(\sigma)\,\alpha) \tag{3.8}$$

Here, x values are F values of empirical CDF, so from now on F will be used instead of x, and α is estimated as follows:

$$\hat{\alpha} = \left(H^T(\alpha)\,H(\alpha)\right)^{-1} H^T(\alpha)\,F \tag{3.9}$$

Then, by replacing $\hat{\alpha}$ into above LS error (3.8), we get

$$J(\sigma, \hat{\alpha}) = F^T\left(I - H(\alpha)\left(H^T(\alpha)\,H(\alpha)\right)^{-1} H^T(\alpha)\right)F \tag{3.10}$$

Therefore, minimization of $J(\sigma, \hat{\alpha})$ is the same as maximization of the following equation over α

$$\max_{\alpha}\left[F^T H(\alpha)\left(H^T(\alpha)\,H(\alpha)\right)^{-1} H^T(\alpha)\,F\right] \tag{3.11}$$

where

$$H_i = \frac{1}{2}\left(1 + erf\,\frac{x - \mu_i}{\sigma\sqrt{2}}\right) \tag{3.12}$$

$$F = \sum \alpha_i H_i \tag{3.13}$$

$$F_i = \left(\alpha_i - \sum_{j=1}^{i-1} \alpha_j \right) H_i \tag{3.14}$$

Then, all α values are determined by

$$\alpha_i = \left(H_i^T H_i \right)^{-1} H_i^T F_i - \sum_{j=1}^{i-1} \alpha_j \tag{3.15}$$

Initially, the variance of the speed center is found with (3.11) by using linear search method or N-R Method. By then inserting this variance value into (3.15), the kernel weight of the corresponding speed center is found. Since the mean values are determined before the NLS method is invoked, the same procedure for finding the variance and kernel weight is repeated for each speed center, resulting in a successive estimation process. In the successive estimation process, each speed center's bandwidths and kernel weights are found one by one.

When the variances are determined, as expressed in (3.11), the equation should reach the maximum value with the variance value which makes the result of the equation greatest. A straightforward guaranteed approach is to perform linear search, however the analysis shows that more than 100 thousands times calculation is needed to find the correct value for each speed center's variance.

Alternative approach for the estimation of bandwidts is N-R Method. This method can determine the same variance values at a reasonable complexity. Although its calculation requires some computations as shown in Appendix A, it reaches sufficiently close variance value to linear search method in less than 10 iteration of calculation, generally just 5 or 6 iterations are sufficient for the convergence. Therefore, N-R method reaches the value faster than linear search method.

The summary of the algorithm for estimation of speed centers, variance values, and kernel weights is given in Algorithm 1.

Algorithm 1 Summary of the Algorithm for Estimation of Parameters

```
load the data
obtain PDF with KDE algorithm as given by (3.3)
obtain empirical CDF with "ecdf" function as formulated in (3.6)
take the derivative of eCDF and then smooth it
find peak values of smoothed PDF to get mean values with PDA
for each mean values do
    determine corresponding F values of speed center from eCDF
    Use Newton-Raphson Method:
    while error between two variance values is bigger than a threshold do
        calculate H values via (3.12)
        maximize (3.11) by changing variance values
        decide on variance values with (A.9)
    end while
    get variance value that maximizes (3.11)
    find the kernel weight with (3.15)
    OR Use Linear Search Method:
    for α^2 from 0 to 100 with step size 0.001 do
        calculate H values via (3.12)
        calculate (3.11) by changing variance values
    end for
    determine variance value that makes (3.11) maximum
    find kernel weight via (3.15)
end for
```

3.5 Tracking of Traffic Density Estimation with Scalar Kalman Filter

Tracking is very much needed when newly arrived data needs to be processed in addition to the past data. Instead of going back to the initial state of estimation of the parameters by using all the existing and the newly arrived data, the estimation of the parameters is just updated with the arrival of new data. Hence, the final estimates are like reaching a consensus between already estimated parameters and newly estimated ones. Moreover, the importance of the old and new data is not the same for the estimation, because new data has more emphasis on estimation and is seen as more probable to convey the current traffic scenario. Therefore, on the contrary to just reordering estimation results according to their number of samples in data sets, the use of a forgetting factor is necessary to improve the tracking capability in time varying parameter estimation [40]. Forgetting factor can be defined as the concept of forgetting in which older data is gradually scrapped by taking into consideration of more recent information [41]. The main idea behind this concept is to give less weight to older data and more weight to the new one [41].

In this study, Scalar Kalman Filter (KF) has been used for tracking. Its scalar state - scalar observation $(s[n-1], x[n])$ generality level is chosen as an approach. The scalar state and the scalar observation equations are as follows [18]:

$$s[n] = \lambda s[n-1] + u[n] \qquad n \geq 0 \tag{3.16}$$

$$x[n] = s[n] + w[n], \tag{3.17}$$

where λ is called the forgetting factor with $0 < \lambda < 1$, $u[n]$ is White Gaussian Noise (WGN) with $u[n] \sim \mathcal{N}(0, \sigma_u^2)$, $w[n] \sim \mathcal{N}(0, \sigma_w^2)$, and $s[-1] \sim \mathcal{N}(\mu_s, \sigma_s^2)$. $w[n]$ differs from WGN only in that its variance is allowed to change in time. Further assumption is the independence of $u[n], w[n]$, and $s[-1]$.

$s[n]$ is estimated based on the data set $\{x[0], x[1], ..., x[n]\}$ as n increases, and this process is simply a type of filtering. KF approach calculates the estimator $\hat{s}[n]$ subjected to the estimator for the previous sample $\hat{s}[n-1]$ and thus, it is recursive in nature [18].

With $n \geq 0$, the scalar KF equations for tracking are as follows:

$$\text{Prediction:} \quad \hat{s}[n \mid n-1] = \lambda \hat{s}[n-1 \mid n-1] \tag{3.18}$$

$$\text{Min Prediction MSE:} \quad M[n \mid n-1] = \lambda^2 M[n-1 \mid n-1] + \sigma_u^2 \tag{3.19}$$

$$\text{Kalman Gain:} \quad K[n] = \frac{M[n \mid n-1]}{\sigma_w^2 + M[n \mid n-1]} \tag{3.20}$$

$$\text{Correction:} \quad \hat{s}[n \mid n] = \hat{s}[n \mid n-1] + K[n](x[n] - \hat{s}[n \mid n-1]) \tag{3.21}$$

$$\text{Min MSE:} \quad M[n \mid n] = (1 - K[n])\, M[n \mid n-1]\,. \tag{3.22}$$

The summary of the algorithm for tracking of parameters is given below.

Algorithm 2 Summary of the Algorithm for Tracking of Parameters

load the data
estimate mean values, variance values, and kernel weights with Algorithm 1
load new data
estimate mean values, variance values, and kernel weights of new data with Algorithm 1
decide forgetting factor
determine prediction via (3.18)
calculate MMSE
calculate minimum prediction MSE via (3.19)
determine Kalman gain via (3.20)
made correction of the parameter via (3.21)
made MMSE correction via (3.22)

The explanation of all parts of the model is made in this chapter. Both estimation and tracking processes are introduced and algorithms for these processes of mean values, kernel weights, and variance values are given. Now, the next step is to check the model with examples.

Chapter 4

Numerical Calculations for Traffic Density Estimation

In this chapter, the proposed approach for estimation will be examined with an assumed system that has 5 speed centers and real time data of New York City. Also a table that includes four different traffic scenarios will be given. Then the successes and problems of the system will be observed. [1]

4.1 An Example Traffic Scenario with Five Speed Centers

A traffic scenario with 5 speed centers is assumed. The performance of the approach for estimated mean, variance, and kernel weight values will be assessed. The assumed scenario has the following typical parameters that are believed to form in a highway passing through a city:

$$\mu_1 = 40 \quad \mu_2 = 70 \quad \mu_3 = 80 \quad \mu_4 = 100 \quad \mu_5 = 115$$

$$\sigma_1^2 = 7 \quad \sigma_2^2 = 6 \quad \sigma_3^2 = 5 \quad \sigma_4^2 = 6 \quad \sigma_5^2 = 7$$

$$\alpha_1 = 0.1 \quad \alpha_2 = 0.2 \quad \alpha_3 = 0.3 \quad \alpha_4 = 0.25 \quad \alpha_5 = 0.15$$

[1]The estimation approach with examples presented in this chapter is accepted to publish in Turkish Journal of Electrical Engineering and Computer Sciences in 2016 [33].

The PDF of the assumed system is given in Figure 3.4, while the CDF plot is given in Figure 4.1.

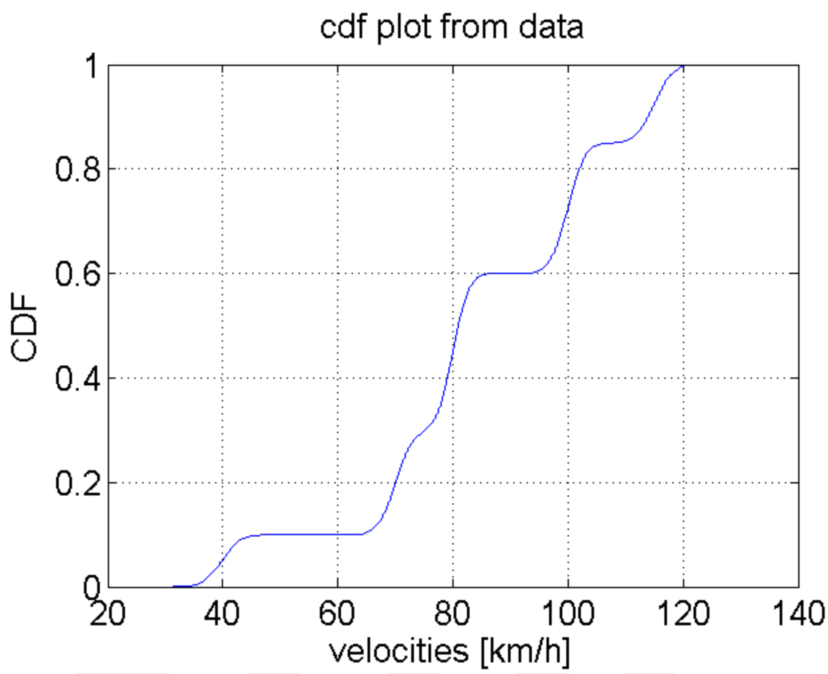


FIGURE 4.1: CDF plot for the example system

The mean values of the speed centers estimated via a peak detection algorithm are as follows:

$$\hat{\mu}_1 = 39.8921 \quad \hat{\mu}_2 = 69.7892 \quad \hat{\mu}_3 = 79.9655 \quad \hat{\mu}_4 = 100.0481 \quad \hat{\mu}_5 = 115.0541$$

We can see that the results are so close to real values as the MMSE (minimum mean square error) is 0.0125. The variances and kernel weights are estimated by using two methods as explained above. For linear search method, which takes long time and but provides more accurate results, the estimated variances are:

$$\hat{\sigma}_1^2 = 7.1350 \quad \hat{\sigma}_2^2 = 5.5230 \quad \hat{\sigma}_3^2 = 5.6700 \quad \hat{\sigma}_4^2 = 5.8980 \quad \hat{\sigma}_5^2 = 6.2970$$

$$\hat{\alpha}_1 = 0.0983 \quad \hat{\alpha}_2 = 0.1956 \quad \hat{\alpha}_3 = 0.3102 \quad \hat{\alpha}_4 = 0.2475 \quad \hat{\alpha}_5 = 0.1569$$

When the error amounts are analyzed, it is observed that kernel weights and speed centers have less error when compared to variance values. However, the estimation of variances is an intermediate step before the estimation of the kernel weights. Although variance estimation provides useful information about the traffic density, the speed centers and kernel weights are more critical in assessing the traffic density. We observe that the proposed approach can estimate the mean values and the kernel weights very close to the actual values. The MMSE for variance is 0.2399 and for kernel weights is 3.6030×10^{-5}. For N-R method, which reaches the result quickly, the results are as follows:

$$\hat{\sigma}_1^2 = 7.1350 \quad \hat{\sigma}_2^2 = 5.5232 \quad \hat{\sigma}_3^2 = 5.6695 \quad \hat{\sigma}_4^2 = 5.8983 \quad \hat{\sigma}_5^2 = 6.2963$$

$$\hat{\alpha}_1 = 0.0983 \quad \hat{\alpha}_2 = 0.1956 \quad \hat{\alpha}_3 = 0.3102 \quad \hat{\alpha}_4 = 0.2475 \quad \hat{\alpha}_5 = 0.1472$$

The MMSE for variance values found by using N-R method is 0.2399 and for kernel weights it is 2.8076×10^{-5}.

As can be seen from the estimated values, the proposed approach can accurately estimate the targeted parameters, as MMSE values are so small for the traffic density estimation. Since, the model adopts a successive approach, it is expected that an increase in error (error propagation) occurs when latter parameters in the order are estimated. However, for the sample system, the difference between α_5 and its real value is less than the difference between α_2 and α_3 and their real values. This is due to the fact that for every speed center, first the variance is found and then its kernel weight is estimated, thereby eliminating the error propagation.

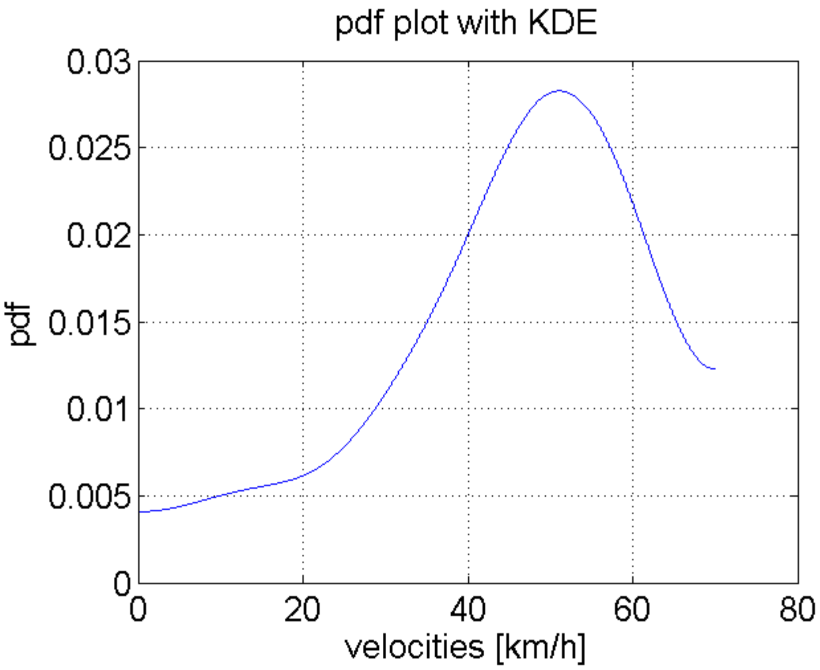


FIGURE 4.2: PDF plot for the real time speed data

4.2 The Estimation of A Real Time Data

As a second example, we examine the system with real data of New York City's traffic [42], which is obtained via sensor feeds at the major arterials and highways. Unfortunately the data has only one kernel weight. The PDF plot of the data is shown in the Figure 4.2, its empirical CDF plot and smoothed version of the CDF are seen in the Figure 4.3. The estimated speed center is 51.17875 by using peak detection algorithm. For linear search method, the variance value is 325.6390 and the kernel weights are 1, as expected. For N-R method, the variance value is 325.6430 and the kernel weight is 1. The variance value of the system is too large, because there is no other speed center and all the vehicles are around a single cluster.

As seen in 4.2, the speed center estimation matches with the peak velocity values in the PDF plot. Since, there is only one kernel, its weight is equal to 1. We could not find real data with more speed centers as the companies working in this area are mostly willing to show their final products like maps, and they are reluctant to share their data.

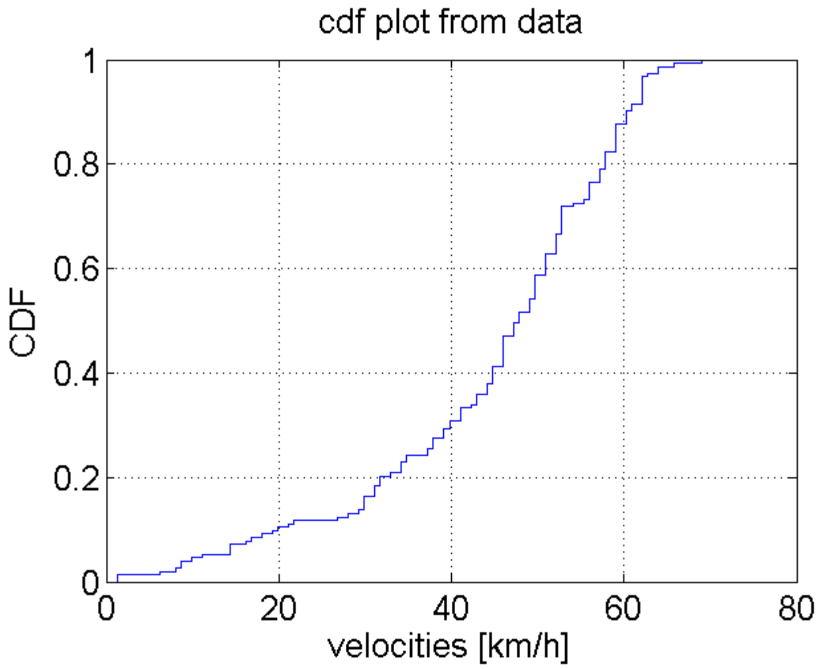
4.3 Traffic Density Estimation with Different Kernel Numbers

The model is also tested for systems that have different kernel numbers like 1, 3, and 7 and the results are shown in the Table 4.1. Although MMSE of variance values higher than other two parameters, the variance estimations are used for the estimation of kernel weights, which are more critical in the overall process.

Table 4.1: MMSE of traffic density estimation with different kernel numbers

Used Method	Peak Detection	Newton-Raphson		Linear Search	
# of Kernels	Mean	Kernel Weights	Variance	Kernel Weights	Variance
1	0.013572	0.000000	0.324786	0.000000	0.324900
3	0.034949	0.000072	0.223417	0.000107	0.159526
5	0.012502	0.000028	0.239866	0.000036	0.239853
7	0.032502	0.000030	0.638211	0.000030	0.631461

In this chapter, the proposed approach for estimation of three parameters is tested with three different examples. From results of different traffic scenarios, the model achieved accurate results in all.



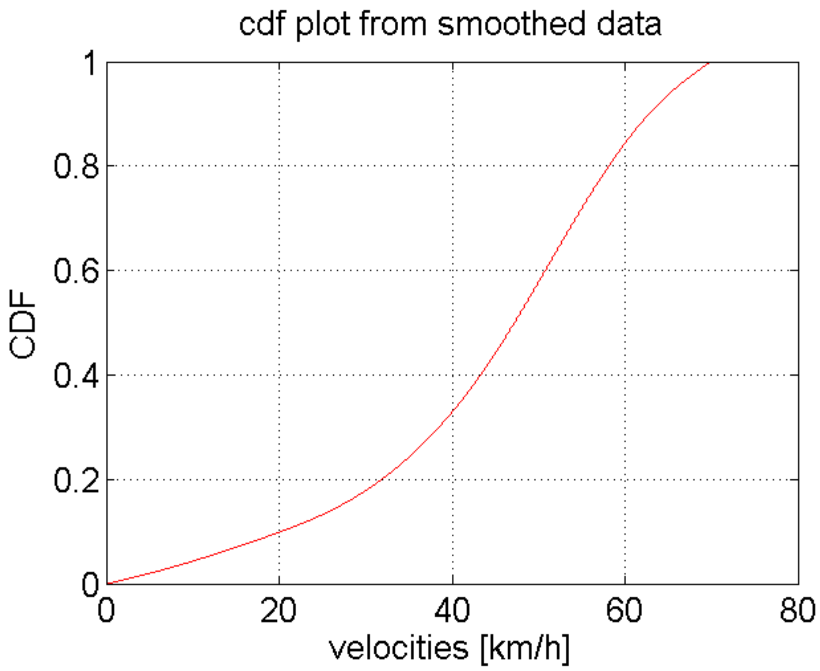


FIGURE 4.3: CDF plot and its smoothed version for the real time speed data

Chapter 5

Examples to Test Tracking Part of the Model

In this chapter, the proposed approach for tracking will be examined with an assumed system, and the advantages and drawbacks of the system will be observed. The system will be tested with 3 examples: the first one examines tracking with the change only in speed centers, the second one evaluates the tracking of kernel weights' changes, and finally the last one investigates what happens if the all variables have new different values. To simulate the given scenarios, Data Set 1 is produced and estimated firstly and then used in all three examples. For the first example, Data Set 2 and Data Set 3 are also created. For the second example, tracking of kernel weights is performed by using Data Set 1 and Data Set 4. In the last example, Data Set 1 and Data Set 5 are used for the estimations and tracking. [1]

Before we further proceed, we will state how some parameters in the tracking process are chosen. For example, the forgetting factor λ is calculated as a ratio of number of samples in the first data set and the number of all samples. By using the forgetting factor, the sample numbers are used implicitly in the tracking equation, however, the tracking is simply neither updating the overall estimation according to the number of samples nor giving equal weights to both old and new estimates. As the examples will show, the final estimates are closer to the estimates based on the newly arrived data rather than

[1] The estimation and tracking approaches with examples presented in this chapter is accepted to publish in IU-Journal of Electrical & Electronics Engineering and will be published Vol 16, No 2 in 2016 [34].

the estimates from the old data. Assuming that the traffic data is obtained via GPS data, and since the accuracy of GPS data is at least 95% according to GPS Standards [43], error values σ_u^2 and σ_w^2 are assumed to 0.05. For the tracking of mean, instead of $\hat{s}[n-1 \mid n-1]$, Data Set 1's mean estimation and instead of $x[n]$, new data set's mean estimation are used. For the kernel weight's tracking, kernel weight estimations of the aforementioned data sets are taken into calculations. The minimum mean square error of each variable is calculated separately since the system is scalar.

5.1 Tracking with the Change only in Mean Values

A traffic scenario with 3 speed centers is assumed. The performance of the approach will be assessed for the estimated means, variances, and kernel weights as well as tracking. The assumed scenario for the first data set has the following parameters with the number of samples $N = 10000$:

$$\mu_1 = 50 \quad \mu_2 = 70 \quad \mu_3 = 100$$

$$\sigma_1^2 = 6 \quad \sigma_2^2 = 7 \quad \sigma_3^2 = 5$$

$$\alpha_1 = 0.3 \quad \alpha_2 = 0.5 \quad \alpha_3 = 0.2$$

It is needed to be emphasized that differently from [4], in addition to Gaussian distribution variance values, the contribution of GPS allowable error is added as variance. For the given example, uniformly distributed additional variance values are 2.5, 3.5, and 5, respectively. The estimation of Data Set 1's mean values via a peak detection algorithm are as follows:

$$\hat{\mu}_1 = 50.1489 \quad \hat{\mu}_2 = 69.9231 \quad \hat{\mu}_3 = 100.1197$$

The estimation results are very close to the real values as the MMSE is 0.0424. Also, the MMSE of each speed center's estimation of Data Set 1 are 0.0222, 0.0059, and 0.0143, respectively. The variances and kernel weights are estimated by using two methods as explained above. For N-R method, which reaches accurate results quicker, the estimated values are as follows:

$$\hat{\sigma}_1^2 = 9.5088 \quad \hat{\sigma}_2^2 = 10.2556 \quad \hat{\sigma}_3^2 = 22.8845$$

$$\hat{\alpha}_1 = 0.3038 \quad \hat{\alpha}_2 = 0.4674 \quad \hat{\alpha}_3 = 0.2288.$$

When the error values are analyzed, kernel weights and speed centers have less error when compared to variance's values. However, the estimation of variances is an intermediate step before the estimation of the kernel weights. Although variance estimation provides useful information about the traffic density, the speed centers and kernel weights are more critical in assessing multilane traffic density. The MMSE of kernel weights is 0.0019 and also the MMSE of each kernel weights of Data Set 1 are 1.4440×10^{-5}, 0.0011, and 0.0008, respectively. For the linear search method, which takes longer time but that generally provides more accurate results, the estimated variances and kernel weights are as follows:

$$\hat{\sigma}_1^2 = 9.5090 \quad \hat{\sigma}_2^2 = 10.2560 \quad \hat{\sigma}_3^2 = 22.8840$$

$$\hat{\alpha}_1 = 0.3002 \quad \hat{\alpha}_2 = 0.4619 \quad \hat{\alpha}_3 = 0.2379$$

with MMSE of 0.0029.

As can be seen from the estimated values, the proposed approach can accurately estimate the targeted parameters as MMSE values are acceptably small for the traffic density estimation.

For the first example, new data sets (Data Set 2 and Data Set 3) are produced with a 5 km/h increase in speed centers while keeping the other parameters unchanged. These data sets have their number of samples as $N = 1000$, and by doing so the results of tracking will be examined by repeating the same procedure. Here, the expectation is that the second tracking would be closer to the speed centers of new data set than the first tracking. For Data Set 2, estimation of mean values and the overall system's corrected speed centers are as follows:

$$\hat{\mu}_1 = 54.9107 \quad \hat{\mu}_2 = 74.8072 \quad \hat{\mu}_3 = 105.2616$$

$$\hat{\mu}_{cor1} = 53.9121 \quad \hat{\mu}_{cor2} = 71.9633 \quad \hat{\mu}_{cor3} = 103.1484$$

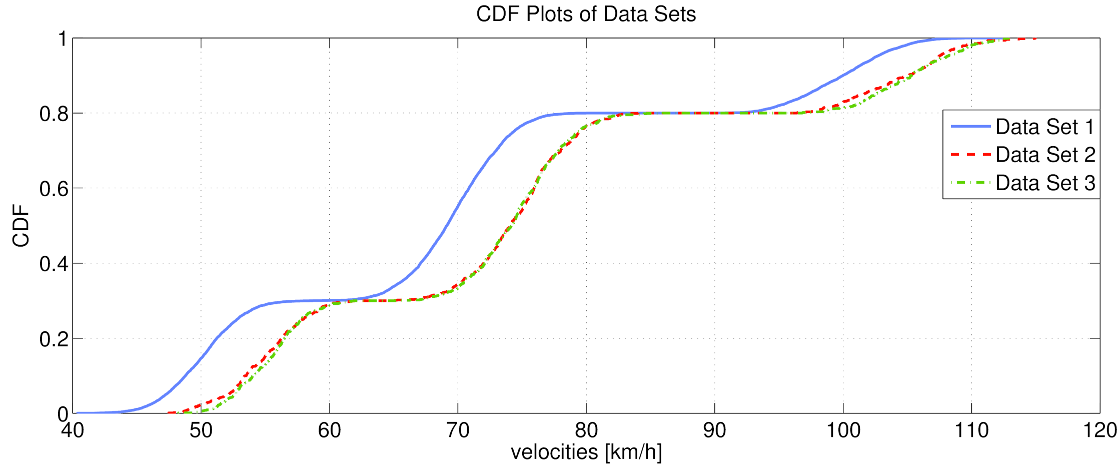


FIGURE 5.1: CDF plots of three data sets

The MMSE of mean estimation of Data Set 2 is 0.1561. Again, the estimation is very close to the real values. Also, the MMSE of each speed center's estimation of Data Set 2 are 0.0302, 0.0431, and 0.0828, respectively. As seen from results of tracking, corrected mean values are not linearly calculated by simply taking the number of samples in data sets. It is also observed that tracking represents real traffic scenarios better since the new data sets have more effect in the evaluation of the current estimates even though they bear less number of samples. It is obvious that if the initial estimates of the firstly received data are more accurate, then the results are closer to the real values. The estimation of the first speed center has more error than the estimation of the second speed center. Thus, the second one has corrected mean values closer to its former estimation (50.1489) than the corrected version of the first mean value's closeness to its former estimation (69.9231). Data Set 3 also has the same parameter values with Data Set 2. Estimation

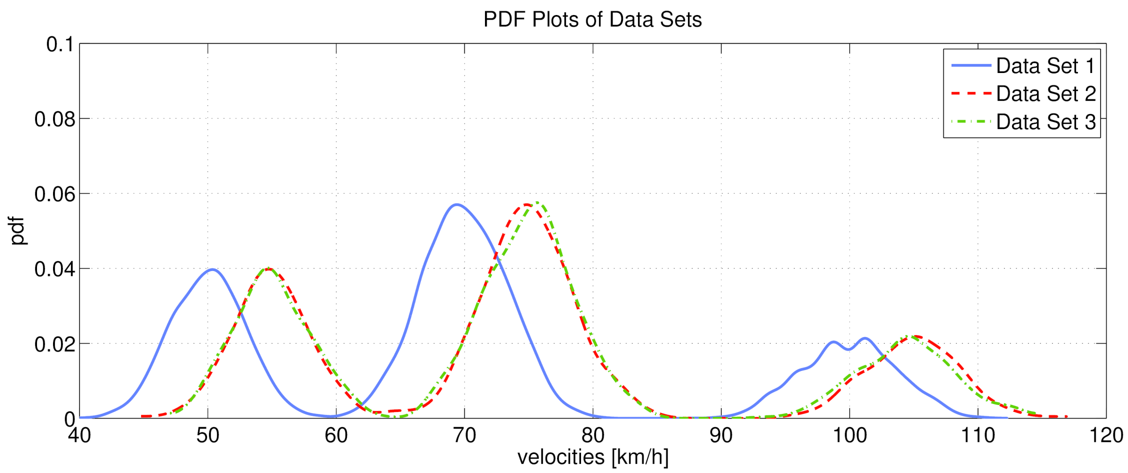


FIGURE 5.2: PDF plots of three data sets

of its mean values and corrected mean values of the overall system that consists of all three data sets including the results of first tracking are as follows:

$$\hat{\mu}_1 = 54.6585 \quad \hat{\mu}_2 = 75.3391 \quad \hat{\mu}_3 = 105.0514$$

$$\hat{\mu}_{cor1} = 54.2267 \quad \hat{\mu}_{cor2} = 74.7701 \quad \hat{\mu}_{cor3} = 104.6992.$$

The MMSE of mean estimation of Data Set 3 is 0.2342. As expected, new corrected values of speed centers are higher than the former ones. The illustration of the change in mean values and their kernel weights and variances of data sets can be observed in Figure 5.1 and Figure 5.2.

5.2 Tracking with the Change only in Kernel Weights

In the second example, only kernel weights will change and we will track their values. Data Set 4's kernel weights for $N = 1000$ number of samples are given as follows:

$$\alpha_1 = 0.5 \quad \alpha_2 = 0.4 \quad \alpha_3 = 0.1.$$

Since in our case N-R Method's MMSE is less than linear search one, estimation of kernel weights and their tracking are as follows:

$$\hat{\alpha}_1 = 0.5154 \quad \hat{\alpha}_2 = 0.3863 \quad \hat{\alpha}_3 = 0.0983$$

$$\hat{\alpha}_{cor1} = 0.3961 \quad \hat{\alpha}_{cor2} = 0.4027 \quad \hat{\alpha}_{cor3} = 0.1466.$$

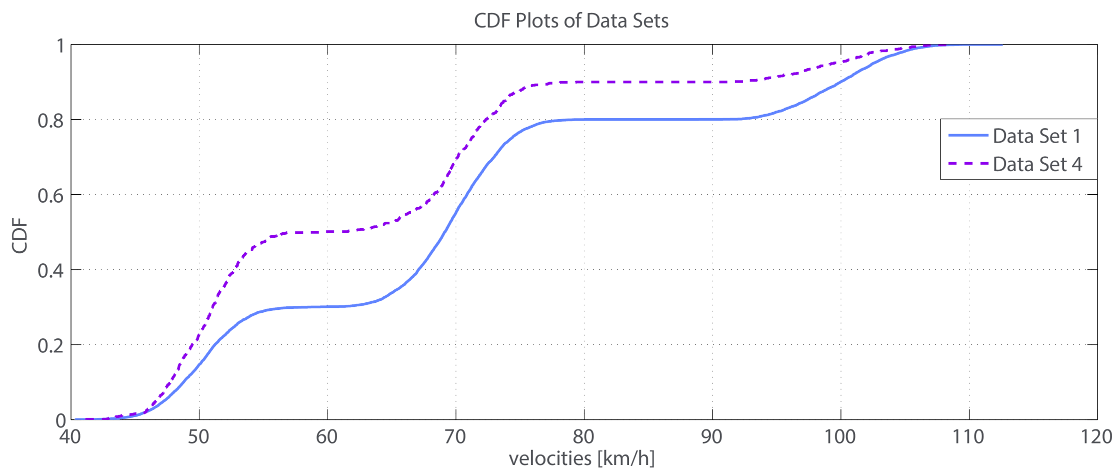


FIGURE 5.3: CDF plots of Data Set 1 and Data Set 4

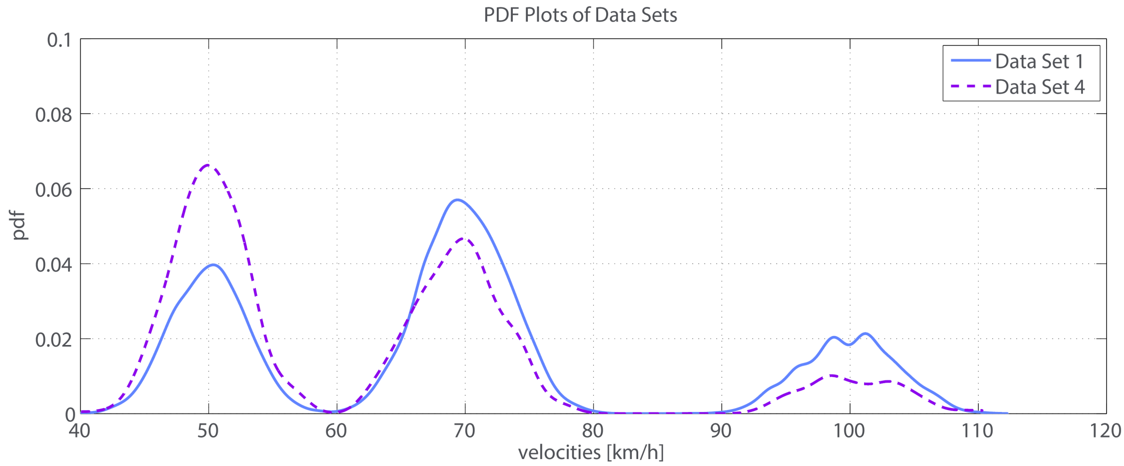


FIGURE 5.4: PDF plots of Data Set 1 and Data Set 4

The MMSE of kernel weights estimation of Data Set 4 is 4.2658×10^{-4}. As seen from corrected kernel weights results, the error of each values of first data set's estimation is correlated with final calculated values. The difference between Data Set 1 and Data Set 4 can be seen in Figure 5.3 and Figure 5.4.

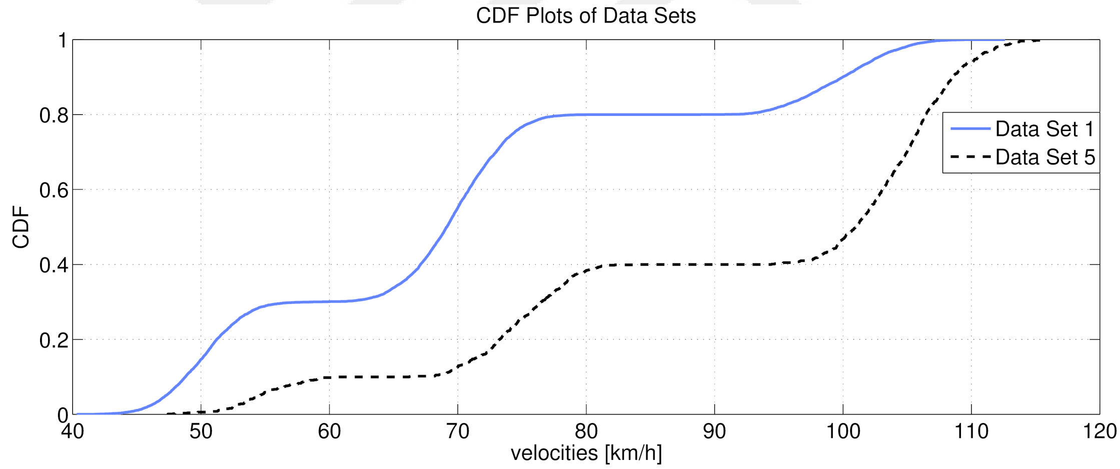


FIGURE 5.5: CDF plots of Data Set 1 and Data Set 5

5.3 Tracking with the Change in All Three Parameters

In the last example, values of all parameters of Data Set 1 will be changed, and their tracking will be calculated for $N = 1000$ number of samples. The estimation of mean values via PDA and the estimation of kernel weights and variance values with N-R Method including assumed sample system parameters are given in Table 5.1. Also tracking results of Data Set 5 are shown in the table. The MMSE of speed center estimation is

0.2609 and the MMSE of kernel weight estimation is 1.6213×10^{-4}. As seen from all examples, the system has performed well for not only the change of a single parameter but also for the change of all parameters. The PDF and CDF plots of Data Set 1 and Data Set 4 are shown in Figure 5.5 and Figure 5.6.

TABLE 5.1: Example 3's parameters, results of estimation and tracking

parameter name / parameter number	μ	σ^2	α	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\alpha}$	μ_{cor}	α_{cor}
1	55	5+2.75	0.1	54.7625	6.0102	0.0918	53.7797	0.1838
2	75	6+3.75	0.3	74.9063	12.3964	0.2985	72.0373	0.3523
3	105	7+5.25	0.6	105.4424	19.1120	0.6096	103.3024	0.4331

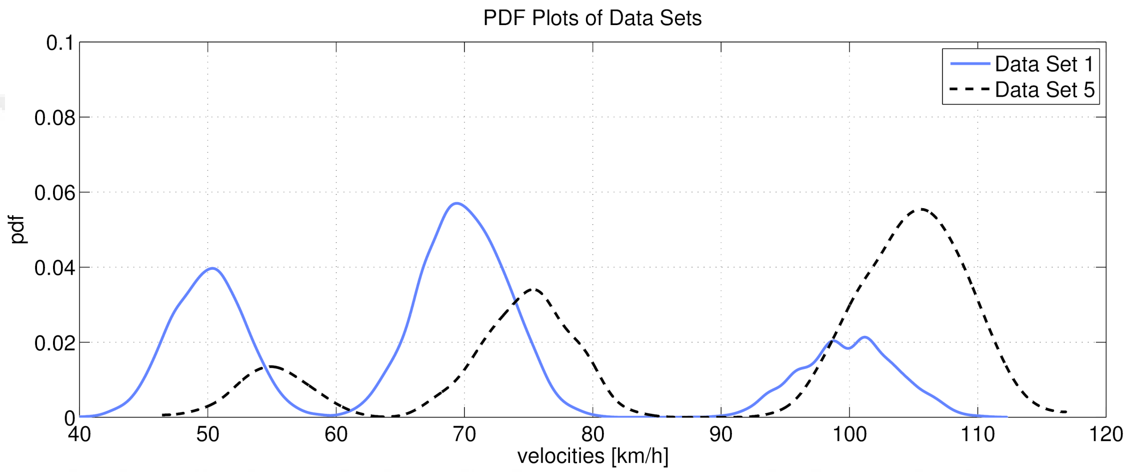


FIGURE 5.6: PDF plots of Data Set 1 and Data Set 5

In this chapter, the proposed approach for tracking is tested with three different examples. No matter whether there is a change in only one parameter or in all three parameters at the same time interval, the model has achieved successful tracking results. Also, the estimation results to get tracking values are sufficiently accurate. Numerical results will be evaluated in the next chapter.

Chapter 6

Assesment

The estimation model is examined with several scenarios in Chapter 4 and it achieved accurate results in all. Even when it is tested for low and high-speed values, it performed well for the estimations. On the contrary to [11], which only estimates kernel weights, this study deals with not only kernel weights but also speed centers and variance values. While estimating variance, N-R method reaches the results very quickly and its results do not have much difference between linear search method results.

The model is also tested for systems that have different kernel numbers like 1, 3, 5, and 7 and the results are shown in the Table 4.1. The estimations of mean values are found with peak detection after deciding CDF and then getting smoothed version of PDF which is derivated from CDF. The weights and bandwidths are estimated either by N-R or linear search methods, successively. These two approaches have reached sufficiently accurate and close results between each other.

The results in Chapter 4 show that the performance of estimation process is accurate. The error for mean estimation is less than 4%, while for kernel weights it is less than 1.1×10^{-4} when both methods are used. As for the variance, the MMSE error is higher but less than 0.7. The variance estimates are exploited for the estimation of kernel weights, which are more critical in the overall process. Therefore, the amount of error in mean values and kernel weights are more critical than variance values while evaluating the system performance. As seen in the Table 4.1, for different scenarios, the proposed approach performs consistently well.

The model with estimation and tracking is tested with three different examples and for all three, it performed well by giving out the desired results. This model includes GPS error and tracking, additionally. Chosen speed values are relatively middle and high speed levels for driving standards. The model was also examined in low and high speed values, and it performed well for estimations and their tracking.

While in estimating variances, N-R method reaches the results very quickly and its performance is comparable to the linear search method results. For example, for every estimation process of each variance value, linear search method needs more than 100 thousand multiplications to get maximum values in (3.11), while N-R method reaches maximum value in less than 10 iteration even though its evaluation needs some heavy computation. Scalar KF with scalar state - scalar observation generality level has achieved preferred outcomes instead of just linear calculations of two estimation results.

The simulation results in Chapter 5 indicate that the error of mean estimation for examples is less than 0.261, while it is less than 0.002 for kernel weights when N-R method is used. Here the variances are calculated as intermediate variables in estimating the kernel weights. These error rates are considered to represent multilane traffic condition accurately when compared with other studies in the literature such as [44] and [45]. We can use the following mean-percentage error formula that is used in [44] to compare error rates as

$$E_{MPE} = \frac{1}{M} \sum_{i=1}^{M} \left| \frac{\alpha_i \hat{\alpha}_i}{\alpha_i} \right|. \tag{6.1}$$

Here, M is the number of total speed centers and it is equal to 3 in the current study.

The error in [44] is around 13% and the error in [45] is around 10%. Meanwhile, the error rate of mean values in this study reaches a maximum value of 0.11% when (6.1) is used. If we modify the equation (6.1) for error rate of kernel weight estimation, its maximum becomes 2.04%. Thus, by using the proposed approach, an accurate multilane traffic density estimation and its tracking are realized.

A criticism to this study can be about the error propagation due to its successive approach. Yet, as explained above and seen in the results of the example, successive system model does not pave the way for error propagation since the mean values are estimated first, then for each mean value, its variance value is estimated, followed by its kernel weight estimation from CDF of KDE.

Another criticism to this study could be the requirement of more computational power when compared to the estimators used in the studies [44] and [45]. By using both N-R and Linear Search methods, average running time for parameter estimations of old data and new data, and their tracking is approximately 2 minutes and 10 seconds. However, here, estimation of the same parameters are made twice to compare these two methods. As mentioned in Chapter 1, although N-R Method has computational complexity, it is faster than linear search method. If we use only Newton-Rapson Method to estimate old data and new data, then perform the tracking, the average running time reduces to approximately 55 seconds. The computer used has a 1.70 GHz CPU, 4.00 GB RAM, and 128.00 GB memory.

Chapter 7

Conclusion

In traffic management, one of the most important factors and indicators is the performance of the transportation networks that is typically measured via the mobility on the roads. Therefore, traffic density estimation and its prediction play a crucial role in traffic related decisions. In this thesis, multilane traffic density estimation was resolved by estimating the mean values, bandwidths, and kernel weights of clusters, which represent a group of moving vehicles. Then, its tracking is determined in the case of the entrance of the new data to the system.

In the thesis, first probability densities are modeled by using Kernel Density Estimation. Secondly, empirical CDF is found by using Kolmogorov-Smirnov Test. Then, speed centers are found by using a peak detection algorithm. In the last step of estimation, variance values and kernel weights are found successively by using separation of parameters property of nonlinear Least Square Estimation. Here the variances or the kernel bandwidths are determined in two ways: linear search method and Newton-Raphson Method. Furthermore, for the same road, tracking of former estimation and new estimation with less amount of data is determined by using Scalar Kalman Filter with scalar state - scalar observation generality level. The roads' traffic density estimation is then updated with the newly calculated values.

The proposed model and the estimation procedure is tested with a sample system and it was observed that the proposed estimator achieve a good performance. Moreover, a real time traffic data is used to examine the system, and it fits with normal distribution hypothesis and also the speed center is found accurately. Due to lack of more than one kernel property of the data, we were not able to check the estimation of kernel weights and variance values for the real data. In addition, the model performed very well for simulated systems that have different number of speed centers, variance values and kernel weights.

The proposed model and the tracking procedure is also tested. Three different sample cases representing a) change in speed centers, b) change in kernel weights, and c) change in all parameters, i.e. mean values, kernel weights, and variance values are analyzed in order to validate the proposed model. It is observed that the proposed estimator and the tracking algorithm perform very well when compared with the state of the art.

This study can also be applied practically. As mentioned above, traffic density estimation of different lanes separately is so crucial to solve traffic-related problems. This model can be part of the dynamic traffic density and congestion deteciton and alternative routing suggestion in highways like D-100 highway in istanbul. Also, currently, most of the traffic map applications show all lanes with the same speed and the same color. However, with this model, they might evaluate their data more accurately and give their users more reliable information and better routing advices.

This model is tested a real-time data that has only one lane in the road. It would be better to have traffic data of more lanes to test. Moreover, it evaluates only loaded data, hence it should be used practically with a reliable data collection method to observe accurate estimations. This current study can further be extended to the prediction of the multilane traffic density for a given time interval, say daily or weekly.

Appendix A

Derivation of Newton-Raphson Method for the Estimation of Variance Values and Kernel Weights

Newton-Raphson Method is derived to reach the zero crossing point and its general formula is as [46]:

$$x_{i+1} = x_i - \frac{f\left(x_i\right)}{f'\left(x_i\right)} \tag{A.1}$$

where i is the iteration number.

However, in this model, we deal with maximization problem of (3.11), thus, in order to apply N-R method, we need to take the derivative of the (3.11) first, then we can implement N-R to find zero crossing point. Therefore, first and second derivatives of (3.11) have to be found.

Here, we perform the search on variance and we have two different components, H and F as given in (3.12) and (3.14). Since F is found from empirical CDF, we concentrate only on H. Let's use s instead of σ^2 into the equations.

If we use short notation to identify (3.11), it can be written as:

$$B = H\left(H^T H\right)^{-1} H^T \quad \text{and} \quad A = F^T B F \tag{A.2}$$

Then, the first and second derivatives of A become

$$\frac{\partial A}{\partial s} = F^T B' F \tag{A.3}$$

and

$$\frac{\partial^2 A}{\partial s^2} = F^T B'' F \tag{A.4}$$

whereas the first and second derivatives of B are [47] - [48]:

$$\begin{aligned}\frac{\partial B}{\partial s} =& H' \left(H^T H\right)^{-1} H^T + H \left(H^T H\right)^{-1} H'^T \\ &- H \left(H^T H\right)^{-1} \left(H'^T H + H^T H'\right) \left(H^T H\right)^{-1} H^T\end{aligned} \tag{A.5}$$

and

$$\begin{aligned}\frac{\partial^2 B}{\partial s^2} =& H'' \left(H^T H\right)^{-1} H^T + 2H' \left(H^T H\right)^{-1} H'^T \\ &+ H \left(H^T H\right)^{-1} H''^T \\ &- H' \left(H^T H\right)^{-1} \left(H'^T H + H^T H'\right) \left(H^T H\right)^{-1} H^T \\ &- H \left(H^T H\right)^{-1} \left(H'^T H + H^T H'\right) \left(H^T H\right)^{-1} H'^T \\ &- H' \left(H^T H\right)^{-1} \left(H'^T H\right) \left(H^T H\right)^{-1} H^T \\ &- H' \left(H^T H\right)^{-1} \left(H^T H'\right) \left(H^T H\right)^{-1} H^T \\ &- H \left(H^T H\right)^{-1} \left(H'^T H\right) \left(H^T H\right)^{-1} H'^T \\ &- H \left(H^T H\right)^{-1} \left(H^T H'\right) \left(H^T H\right)^{-1} H'^T \\ &- H \left(H^T H\right)^{-1} \left(H''^T H + H'^T H'\right) \left(H^T H\right)^{-1} H^T \\ &- H \left(H^T H\right)^{-1} \left(H'^T H' + H^T H''\right) \left(H^T H\right)^{-1} H^T \\ &+ 2H \left(H^T H\right)^{-1} \left(\left(H'^T H + H^T H'\right) \left(H^T H\right)^{-1}\right)^2 H^T.\end{aligned} \tag{A.6}$$

Here the first and second derivatives of H, which was given in (3.12), are given by [49]:

$$\frac{\mathrm{d}H}{\mathrm{d}s} = -\frac{(x-\mu)\, e^{-\frac{x^2 - 2\mu x + \mu^2}{2s}}}{2^{\frac{3}{2}} \sqrt{\pi} s^{\frac{3}{2}}} \tag{A.7}$$

$$\frac{\mathrm{d}^2 H}{\mathrm{d}s^2} = \frac{(x-\mu)\left(3s - x^2 + 2\mu x - u^2\right) e^{-\frac{x^2 - 2\mu x + \mu^2}{2s}}}{2^{\frac{5}{2}} \sqrt{\pi} s^{\frac{7}{2}}}. \tag{A.8}$$

Then, the variance estimation becomes

$$s_{i+1} = s_i - \frac{A'}{A''} \tag{A.9}$$

Bibliography

[1] British Cycling Federation. Effective Traffic Riding - Filtering in Traffic. `https://www.britishcycling.org.uk/knowledge/skills/article/izn20130830-Effective-traffic-riding-part-2-0`, 2016. Accessed: 2016-08-10.

[2] Yandex. Real-Time Traffic Density Map of Istanbul. `https://yandex.com.tr/harita/-/CVhaVTiA`, 2016. Accessed: 2016-08-10.

[3] A. Amico, U. Mengali, and M. Morelli. Channel estimation for the uplink of a ds-cdma system. *Wireless Communications, IEEE Transactions on*, 2(6):1132–1137, 2003.

[4] V. Kekatos, A. A. Rontogiannis, and K. Berberidis. A robust method for estimating multipath channel parameters in the uplink of a ds-cdma system. In *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, volume 5, pages 3127–3131. IEEE, 2005.

[5] Hurriyet Daily News. Istanbul drowning in traffic chaos, report announces. `http://www.hurriyetdailynews.com/istanbul-drowning-in-traffic-chaos-report-announces.aspx?PageID=238&NID=44312&NewsCatID=341`, 2013. Accessed: 2016-08-10.

[6] Sabah. 'Dur-kalk'ta dünya lideri İstanbul. `http://www.sabah.com.tr/gundem/2013/10/02/durkalkta-dunya-lideri-istanbul`, 2013. Accessed: 2016-08-10.

[7] L. Jäntschi, S. D. Bolboaca, et al. Distribution Fitting 2. Pearson-Fisher, Kolmogorov-Smirnov, Anderson-Darling, Wilks-Shapiro, Kramer-von-Misses and Jarque-Bera statistics. *Bulletin UASVM Horticulture*, 66(2):691–697, 2009.

[8] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization.* John Wiley & Sons, 1992.

[9] R. Laxhammar, G. Falkman, and E. Sviestins. Anomaly detection in sea traffic - A comparison of the Gaussian Mixture Model and the Kernel Density Estimator. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 756–763, July 2009.

[10] K. P. Murphy. *Machine Learning: A Probabilistic Perspective.* MIT press, 2012.

[11] A. Tabibiazar and O. Basir. Kernel-Based Optimization for Traffic Density Estimation in ITS. In *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, pages 1–5, Sept 2011. doi: 10.1109/VETECF.2011.6093194.

[12] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33(3):1065–1076, 09 1962. doi: 10.1214/aoms/1177704472. URL `http://dx.doi.org/10.1214/aoms/1177704472`.

[13] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Ann. Math. Statist.*, 27(3):832–837, 09 1956. doi: 10.1214/aoms/1177728190. URL `http://dx.doi.org/10.1214/aoms/1177728190`.

[14] R. Laxhammar, G. Falkman, and E. Sviestins. Anomaly detection in sea traffic - a comparison of the gaussian mixture model and the kernel density estimator. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 756–763, July 2009.

[15] G. Dequan, L. Xiangzhen, Y. Chengyue, and Y. Zhang. Spatial patterns analysis of urban road traffic accidents based on gis. In *Automatic Control and Artificial Intelligence (ACAI 2012), International Conference on*, pages 1898–1901, March 2012. doi: 10.1049/cp.2012.1363.

[16] C. C. Lee, W. C. Lee, H. Cai, H. R. Chi, C. K. Wu, J. Haase, M. Gidlund, and B. Li. Traffic condition monitoring using weighted kernel density for intelligent transportation. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pages 624–627, July 2015. doi: 10.1109/INDIN.2015.7281807.

[17] M. Yılan and M. K. Özdemir. A Simple Approach to Traffic Density Estimation by using Kernel Density Estimation. In *Signal Processing and Communications*

Applications Conference (SIU), 2015 23th, pages 1865–1868, May 2015. doi: 10.1109/SIU.2015.7130220.

[18] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*, volume 1. Pearson Education, 2013.

[19] N. Bhouri, H. Hadj-Salem, M. Papageorgiou, and J. M. Blosseville. Estimation of traffic density on motorways. *Traffic Engineering & Control*, 29(11): 579–583, 1988. URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-0024107268&partnerID=40&md5=7bee056b592bc2b3d79c5962bd3591c8`. cited By 12.

[20] N. Buch, S. A. Velastin, and J. Orwell. A review of computer vision techniques for the analysis of urban traffic. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):920–939, Sept 2011. ISSN 1524-9050. doi: 10.1109/TITS.2011.2119372.

[21] P. Reinartz, M. Lachaise, E. Schmeer, T. Krauss, and H. Runge. Traffic monitoring with serial images from airborne cameras. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61(3-4):149–158, 2006. doi: 10.1016/j.isprsjprs.2006.09.009. URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-33751439998&partnerID=40&md5=fcf6016353e9ca02b57d489c60bb7e08`. cited By 0.

[22] Y. Wang, L. Bai, and M. Fairhurst. Robust road modeling and tracking using condensation. *IEEE Transactions on Intelligent Transportation Systems*, 9(4):570–579, Dec 2008. ISSN 1524-9050. doi: 10.1109/TITS.2008.2006733.

[23] K. T. Song and J. C. Tai. Image-based traffic monitoring with shadow suppression. *Proceedings of the IEEE*, 95(2):413–426, Feb 2007. ISSN 0018-9219. doi: 10.1109/JPROC.2006.888403.

[24] F. Bartolini, V. Cappellini, and C. Giani. Motion estimation and tracking for urban traffic monitoring. In *Image Processing, 1996. Proceedings., International Conference on*, volume 3, pages 787–790 vol.3, Sep 1996. doi: 10.1109/ICIP.1996.560850.

[25] J. Segen. A camera-based system for tracking people in real time. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 3, pages 63–67 vol.3, Aug 1996. doi: 10.1109/ICPR.1996.546795.

[26] L. Munoz, L. X. Sun, R. Horowitz, and L. Alvarez. Traffic density estimation with the cell transmission model. In *American Control Conference, 2003. Proceedings of the 2003*, volume 5, pages 3750–3755 vol.5, June 2003. doi: 10.1109/ACC.2003.1240418.

[27] M. Ulmke, O. Erdinc, and P. Willett. Gaussian mixture cardinalized phd filter for ground moving target tracking. In *Information Fusion, 2007 10th International Conference on*, pages 1–8, July 2007. doi: 10.1109/ICIF.2007.4408105.

[28] K. Singh and B. Li. Estimation of traffic densities for multilane roadways using a markov model approach. *IEEE Transactions on Industrial Electronics*, 59(11): 4369–4376, Nov 2012. ISSN 0278-0046. doi: 10.1109/TIE.2011.2180271.

[29] V. Tyagi, S. Kalyanaraman, and R. Krishnapuram. Vehicular traffic density state estimation based on cumulative road acoustics. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1156–1166, Sept 2012. ISSN 1524-9050. doi: 10.1109/TITS.2012.2190509.

[30] P. Borkar and L. G. Malik. Acoustic signal based traffic density state estimation using adaptive neuro-fuzzy classifier. In *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, pages 1–8, July 2013. doi: 10.1109/FUZZ-IEEE.2013.6622444.

[31] T. Tsubota, A. Bhaskar, and E. Chung. Traffic density estimation of signalised arterials with stop line detector and probe data. *Journal of the Eastern Asia Society for Transportation Studies*, 10(0):1750–1763, 2013.

[32] P. Krauthausen and U. D. Hanebeck. Regularized non-parametric multivariate density and conditional density estimation. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*, pages 180–186, Sept 2010. doi: 10.1109/MFI.2010.5604457.

[33] M. Yılan and M. K. Özdemir. Traffic Density Estimation via KDE and Nonlinear LS [accepted, will be published]. In *Turkish Journal of Electrical Engineering & Computer Sciences*, 2016.

[34] M. Yılan and M. K. Özdemir. Multilane Traffic Density Estimation and Tracking [accepted, will be published]. In *IU-Journal of Electrical & Electronics Engineering*, 2016.

[35] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt. *Spread spectrum communications; vols. 1-3.* Computer Science Press, Inc., 1985.

[36] E. G. Ström, G. Erik, and F. Malmsten. A maximum likelihood approach for estimating ds-cdma multipath fading channels. *Selected Areas in Communications, IEEE Journal on*, 18(1):132–140, 2000.

[37] Z. I. Botev, J. F. Grotowski, D. P. Kroese, et al. Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.

[38] Z. Xie and J. Yan. Kernel density estimation of traffic accidents in a network space. *Computers, Environment and Urban Systems*, 32(5):396–406, 2008.

[39] P. M. Djuric and J. Miguez. Model assessment with Kolmogorov-Smirnov statistics. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 2973–2976, April 2009. doi: 10.1109/ICASSP.2009.4960248.

[40] S. Song, J. S. Lim, S. Baek, and K.M. Sung. Gauss newton variable forgetting factor recursive least squares for time varying parameter tracking. *Electronics Letters*, 36 (11):988–990, May 2000. ISSN 0013-5194. doi: 10.1049/el:20000727.

[41] 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*, 43(1):31–55, 2005. doi: 10.1080/00423110412331290446. URL `http://dx.doi.org/10.1080/00423110412331290446`.

[42] NYC OpenData. Real-Time Traffic Speed Data. `https://data.cityofnewyork.us/Transportation/Real-Time-Traffic-Speed-Data/xsat-x5sa`, 2014. Accessed: 2016-01-15.

[43] US DoD. Global positioning system standard positioning service performance standard. *Assistant secretary of defense for command, control, communications, and intelligence*, 2008.

[44] L. Munoz, S. Xiaotian, R. Horowitz, and L. Alvarez. Traffic density estimation with the cell transmission model. In *American Control Conference, 2003. Proceedings of the 2003*, volume 5, pages 3750–3755 vol.5, June 2003. doi: 10.1109/ACC.2003.1240418.

[45] S. Xiaotian, L. Munoz, and R. Horowitz. Highway traffic state estimation using improved mixture kalman filters for effective ramp metering control. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 6, pages 6333–6338 Vol.6, Dec 2003. doi: 10.1109/CDC.2003.1272322.

[46] S. C. Chapra and R. P. Canale. *Numerical Methods for Engineers*. McGraw-Hill, 6th edition, 2010.

[47] E. Bodewig. *Matrix Calculus*. Elsevier, 2014.

[48] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. Siam, 2000.

[49] M. D. Weir, J. Hass, and G. B. Thomas. *Thomas' Calculus*. Pearson Education, 12th edition, 2010.