MOBILITY TRACKING ALGORITHMS FOR CELLULAR NETWORKS BASED ON KALMAN FILTERING

by

Alper Yeşilyurt

B.S., in Electrical and Electronics Engineering, Boğaziçi University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Department of Electrical and Electronics Engineering

Boğaziçi University

2010

MOBILITY TRACKING ALGORITHMS FOR CELLULAR NETWORKS BASED ON KALMAN FILTERING

APPROVED BY:

Prof. Ayşın Bartan Ertüzün (Thesis Supervisor)	. . . .
Assoc. Prof. Mehmet Akar	. . . .
Assist. Prof. A. Taylan Cemgil	. . . .

DATE OF APPROVAL: 20.09.2010

ACKNOWLEDGEMENTS

I dedicate this thesis, with deepest respect and gratitude, to the memory of my deceased thesis supervisor Assist. Prof. Kerem Harmancı who taught, guided and encouraged me a lot for this thesis whenever I needed. I will always appreciate his sincere helps.

I would like to thank my thesis supervisor Prof. Ayşin Bartan Ertüzün whose guidance, supervision, attention and useful comments helped me a lot to conclude this thesis.

I would like to thank members of thesis jury, Assoc. Prof. Mehmet Akar and Assist. Prof. A. Taylan Cemgil, for their attention and help.

I would like to thank Assoc. Prof. M. Kıvanç Mıhçak for his help and guidance throughout my M.S. program.

I also appreciate help of my friends Caner Korkmaz and Abdullah Yakın. My dear family have always their part in any achievement of mine and this work is not an exception.

ABSTRACT

MOBILITY TRACKING ALGORITHMS FOR CELLULAR NETWORKS BASED ON KALMAN FILTERING

This thesis is an in depth theoretical and practical survey of dynamic mobility tracking systems specifically for cellular networks. A user mobility state model that is originally proposed for tracking targets in tactical weapons systems is discussed. This mobility model captures a large range of mobility by modeling acceleration(manuever) as driven by a discrete semi-Markovian command process and a Gaussian time-correlated random process. Linear and nonlinear observation models are presented. For nonlinear model, received signal strength indicator model of a cellular communication network is considered. Based on these models, tracking algorithms are presented and simulated in Matlab. Algorithms differ in the type of observation model they use(linear or nonlinear) and in the way they treat semi-Markovian manuever component and they employ variants of the Kalman filter namely: traditional Kalman filter, extended Kalman filter, uscented Kalman filter or adaptive Kalman filter. Linear observation command mode algorithm, nonlinear observation command mode algorithm and unscented Kalman filter based version of nonlinear observation command noise algorithm are introduced in this thesis. Matlab simulations and root mean square error statistics of simulations represented by sample mean and standard deviation of a number of root mean square error values show that treating command process as an additional state noise is a more accurate approach than treating it as a model variable within a mutliple model adaptive estimator. Extended and unscented Kalman filters are used to calculate predicted states and covariances when nonlinear observation model is used. Simulations and root mean square error statistics of simulations show that nonlinear structure of received signal strength indicator model causes large Gaussian approximation errors for posterior state probability and state and covariance prediction errors in extended and unscented Kalman filters, but smaller in uscented Kalman filter. Moreover, nonlinearity pronounces non-Gausianity of likelihood function used in the adaptive estimator as much as to terminate the estimator prematurely.

ÖZET

HÜCRESEL AĞLAR İÇİN KALMAN FİLTRE ÇIKARIMI TABANLI HAREKET TAKİBİ ALGORİTMALARI

Bu tez çalısmasında özellikle hücresel ağlar için geçerli olan dinamik hareket takip sistemleri teorik ve pratik olarak derinlemesine incelenmiştir. Orjinal olarak taktiksel silah sistemlerinde hedefleri takip etmek için öne sürülen hareket durum modeli tartışılmaktadır. Bu modelde ivme(manevra), yarı Markov emir süreci ve zaman iniltili rastgele Gauss süreci ile modellenerek geniş bir hareket yelpazesi göz önünde tutulmaktadır. Doğrusal ve doğrusal olmayan gözlem modelleri sunulmaktadır. Doğrusal olmayan model için, hücresel haberleşme ağlarında kullanılan alınan sinyal şiddeti göstergesi modeli kullanılmaktadır. Bu modellere bağlı olarak takip algoritmaları sunulmakta ve Matlab'da simule edilmektedir. Algoritmalar, kullandıkları gözlem modelinin tipine ve yarı Markov manevra birimine bakış açılarına göre farklılık göstermekte ve uygun Kalman filtre türleri kullanmaktadır: geleneksel Kalman filtre, genişletilmiş Kalman filtre, sezisiz Kalman filtre veya uyarlanabilir Kalman filtre. Doğrusal ve doğrusal olmayan gözlem komut model algoritmaları ile doğrusal olmayan gözlem komut gürültü algoritmasının sezisiz Kalman filtre versiyonu bu tezde ortaya atılmıştır. Matlab simulasyonları ve simulasyonların karekök ortalama kare hata istatistikleri(ortalama değer ve standart sapma), emir sürecini ilave durum gürültüsü olarak değerlendirmenin emir sürecini çoklu model uyarlanabilir filtrenin model değişkeni olarak değerlendirmeye göre daha doğru ve verimli olduğunu göstermektedir. Doğrusal olmayan gözlem modelinin kullanıldığı durumlarda genişletilmiş ve sezisiz Kalman filtreler tahmini durum ve kovaryansların hesaplanmasında kullanılmaktadır. Sonuçlar göstermektedir ki alınan sinyal şiddeti göstergesi modelinin doğrusal olmayan yapısı, durum olasılığının Gauss olarak yakınsanmasında ve genişletilmiş ve sezisiz Kalman filtrelerin tahmini değerleri kestirmesinde ciddi sorunlar oluşturmaktadır(sezisiz Kalman filtre daha iyi sonuç vermektedir.). Hatta uyarlanabilir filtrenin olasılık fonksiyonunu Gauss olmaktan o derece uzaklaştırmaktadır ki filtre erkenden sonlanmak zorunda kalmaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

$A[D_k]$	State transition matrix of the system at mode D_k
$d_{k,i}$	Distance between the mobile unit and the base station in cell i at time k
$\boldsymbol{d}_k$	Distance vector sequence
D	Finite set of command inputs
D_k	Current model of a multiple model system active at time k
D^j	jth model of a multiple model system at any time
$D_{k,l}$	lth model history through time k
D_k^j	The event that model D^j is in effect at time k
$E[.]$	Expectation
$E[.\|.]$	Conditional expectation
$f_k(.)$	Nonlinear function at time index k
$h_k(.)$	Nonlinear observation function at time index k
$\boldsymbol{I}_k$	Information set at time k containing $\boldsymbol{O}_k$ and $\boldsymbol{U}_k$
I_n	n x n identity matrix
K_k^i	Weights of sample points X_k^i
$M_{ss\|o}$	Conditional covariance matrix of vector $\boldsymbol{s}$ given vector $\boldsymbol{o}$
$M_{k+1\|k}$	Conditional covariance matrix of vector $\boldsymbol{s}_{k+1}$ given $\boldsymbol{O}_k$
M_{so}	Covariance matrix of vectors $\boldsymbol{s}$ and $\boldsymbol{o}$
n_s	Dimension of vector sequence $\boldsymbol{s}_k$
$\mathcal{N}(\mu, \sigma^2)$	Gaussian probability density function with mean μ and variance σ^2
$\boldsymbol{O}_k$	Series of vectors of vector sequence $\boldsymbol{o}_k$ up to time k
$p(.)$	Probability density function
$p(.\|.)$	Conditional probability density function
p_{ij}	Homogenous model transition probability from model D^i to model D^j at any time
$P\{.\}$	Probability
$P\{.\|.\}$	Conditional probability

q_k^j	Probability of model D^j or command j being active at time k
$q^{k,l}$	The conditional probability of the history $D_{k,l}$ given $\boldsymbol{O}_k$ active
r	Number of models of a multiple model system at any time and number of commands of the command process $\boldsymbol{u}_k$
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^{n_s}$	n_s dimensional real space
$\boldsymbol{s}$	vector
$\hat{\boldsymbol{s}}_{k\|k}^j$	Conditional estimate of state vector $\boldsymbol{s}_k$ when model D^j is
$\hat{\boldsymbol{s}}_{k\|k}$	Conditional estimate of vector $\boldsymbol{s}_k$ given $\boldsymbol{O}_k$
$\hat{\boldsymbol{s}}$	Estimate of vector $\boldsymbol{s}$
$\tilde{\boldsymbol{s}}$	Estimation error calculated as $\boldsymbol{s} - \hat{\boldsymbol{s}}$
$\tilde{\boldsymbol{s}}_{k+1\|k}$	Estimation error calculated as $\boldsymbol{s}_{k+1} - \hat{\boldsymbol{s}}_{k+1\|k}$
$\mathbf{s}_{\mathrm{k}}[\mathrm{i}]$	ith component of vector $\mathbf{s}_{\mathrm{k}}$
$\langle \boldsymbol{s}, \boldsymbol{q} \rangle$	Inner product of vectors $\boldsymbol{s}$ and $\boldsymbol{q}$
$\bar{\boldsymbol{s}}$	Mean of vector $\boldsymbol{s}$
$\\|\boldsymbol{s}\\|^2$	Norm of vector $\boldsymbol{s}$
$\boldsymbol{s}_k$	Vector sequence
t_k	Time instant at time index k
T_{k-1}	Sampling interval between t_k and t_{k-1}
$var(.)$	Variance
$\boldsymbol{w}_{k-1}[D_k]$	State model noise vector sequence of the system at mode D_k
X_k^i	Sample points calculated by unscented transform
δ_k	dirac-delta function at time k
λ_k^j	Likelihood function of model D^j at time k
$\vartheta_{k,i}$	RSSI received at the mobile unit from the base station in cell i at time k

EKF	Extended Kalman Filter
LMMSE	Linear Minumum Mean Square Error
LOCM	Linear Observation Command Mode

LOCN	Linear Observation Command Noise
MMSE	Minumum Mean Square Error
MSE	Mean Square Error
NOCM	Nonlinear Observation Command Mode
NOCN	Nonlinear Observation Command Noise
pdf	probability density function
PNOCM	Prefiltered Nonlinear Observation Command Mode
RMSE	Root Mean Square Error
RSSI	Received Signal Strength Indicator
UKF	Unscented Kalman Filter

1. INTRODUCTION

Target tracking is an element of a wider system that performs surveilance, guidance, obstacle avoidance or a similar function. In cellular communication networks, target(user) tracking is specifically used to provide seamless access to the network, quality-of-service (QoS) guarantees for mobile users and reliable location based services. Provision of these services require efficient mobility management systems that are based on robust mobility tracking algorithms. Mobility tracking is distinguished from geolocationing, which only requires instant position estimation. In mobility tracking, not only current location of the mobile station but also its current velocity and acceleration, that together constitute mobility state characteristics of the mobile, are estimated in real time. Estimation of velocity and acceleration parameters enables prediction of future mobility state of the mobile in advance. Mobility state prediction plays a vital role for many applications that require fast handoff since time and location of handoff can be predicted ahead of time [1].

Special importance of mobility tracking for cellular communication networks like GSM, CDMA, ATM etc., makes mobility tracking for cellular networks a popular topic. In any tracking system like those proposed for cellular networks, creating realistic target mobility models to represent dynamic motion of targets and establishing efficient tracking algorithms based on these models are key challenges. For mobility models, special importance is given to the concept of manuevering, deviation from the straight line constant velocity movement [3]. As far as studied, cornerstone works in this area starts with Singer [3] who proposed a time-correlated Gaussian manuever model for manuevering targets that manuever in a time-correlated manner. Then, Moose [4] proposed a two-dimensional semi-Markovian manuever model to represent sudden, large-scale manuevers resulting from pilot(deterministic user) or missile-guidance program commands. Gholson and Moose [5] further developed semi-Markovian concept to be used in three-dimensional spherical coordinates. Moose, Vanlandingham and Mccabe [6] combined time-correlated manuever model of Singer [3] with the semi-Markovian manuever model of Moose [4] for spherical target motion. Spherical target motion model includes range, bearing, and elevation coordinates and mainly used for target tracking in tactical weapon systems. Finally, Liu, Bahl and Chlamtac [7] customized the combined model presented in [6] to model two-dimensional mobility patterns of mobile users in a cellular network. As discussed in [7] and [1], this model can capture a wide range of dynamic and realistic user patterns for cellular networks. Cellular tracking system proposed in [7] uses the nonlinear received signal strength indicator(RSSI) model of a typical cellular network as the observation model.

Based on the cellular mobility models proposed in [7], Kalman filtering based tracking algorithms are proposed in [7] and [1]. Basically, algorithm proposed in [7] uses a simplified multiple model approach regarding the commanding manuever component , called command process, as the model control variable. In this approach, meanuever commands correspond to models. Based on this approach, algorithm proposed in [7] uses an extended Kalman filter(EKF) to estimate user mobility states. MT1 algorithm proposed in [1] uses the same approach as the algorithm proposed in [7] moreover uses an additional prefilter to reduce effect of shadowing noise in RSSI observations and a linear Kalman filter to estimate commands of command process using prefiltered RSSI observations. MT1 applies an EKF that uses prefiltered RSSI observations and command estimates to estimate user mobility states. MT2 algorithm proposed in [1] assumes a single model approach and regards the command process as an additional state noise instead of a model variable and applies an EKF accordingly.

In this thesis, theoretical foundation of general mobility tracking and cellular mobility tracking models and algorithms proposed in [7] and [1] are presented as an in depth survey. Besides, a linear observation model and a linear tracking algorithm based on this model are introduced to create a general cellular tracking framework. Simplified multiple model algorithm proposed in [7] is regarded as invalid and an extended multiple model algorithm that has parallelly operating EKFs is proposed for the nonlinear RSSI observation model. A popular, robust sampling based nonlinear Kalman filter called unscented Kalman filter(UKF) proposed by Julier, Uhlman and Hugh [2] is included to the MT2 algorithm proposed in [1] to improve performance of MT2 algorithm. These algorithms together with their original versions proposed in [7] and [1] are implemented and simulated in Matlab and their performances are compared. Potential problems related with the models and algorithms are presented. A detailed scope of this thesis is given below.

In general, estimation can be considered as filtering noise from the observation of the true signal whose value is to be estimated. Regarding this, Section 2 presents linear minumum mean square error(LMMSE) filtering. Basically, in LMMSE filtering, mean square error(MSE) between the signal and its estimate is to be minimized using linear filters. The most attractive feature of linear filters is that their simple linear and finite dimensional structure is easy to realize and implement computationally. Famous linear state estimator, Kalman filter, presented in Section 3, is LMMSE filter for dynamic systems of some special type.

Mobility tracking systems are nothing but dynamic systems that evolve with respect to time. In this view, Section 3 discusses general framework for analyzing dynamic systems. Discrete time state-space representation of dynamic systems and optimal(in the sense of MSE) Bayesian dynamic state estimation scheme are discussed. Special importance is given to the Kalman filter, the linear minumum MSE(MMSE) state estimator for dynamic systems that have linear model(state model and observation model) equations and white and independent model disturbances. If model disturbances are also Gaussian, Kalman filter becomes the optimal Bayesian MSE state estimator for dynamic systems.

A large number of dynamic systems in practice are nonlinear having at least one nonlinear model equation. Unfortunately, as given in Section 3, optimal MSE state estimator for nonlinear systems is impractical since it requires storage of an infinite dimensional probability density function(pdf) and numerical calculation of integration. Section 3 further discusses suboptimal MSE filtering for nonlinear systems. Two popular approaches, EKF and UKF are presented. Basically, these filters try to linearize the nonlinear model either analitically or probabilistically and utilize linear structure of traditioanl Kalman filter. At the end of Section 3, common adaptive estimation techniques for multiple model systems that have multiple models that switch in time according to a Markovian probalistic model are presented.

Each dynamic system has 2 models represented by 2 equations: state (process or signal) model and observation (measurement) model, which are elaborated in Section 3. Section 4 presents mobility models of the general cellular tracking system that are used to represent dynamic motion of users of cellular networks. Cellular tracking system contains a linear state model, a linear observation model and a nonlinear RSSI observation model.

Section 5 presents Kalman filtering-based two-dimensional cellular tracking algorithms based on the mobility models discussed in Section 4. Basically, one group of algorithms use linear observation model and other group of algorithms use nonlinear RSSI observation model. EKF and UKF are used to cope with the nonlinearity of the RSSI model. Besides, one group of algorithms treats semi-Markovian manuever variable as an additional state noise and other group of algorithms treats it as the mode variable of a multiple model system and uses adaptive estimation schemes presented in Section 3.

Section 6 presents simulation results obtained from independent Monte-Carlo simulations of tracking algorithms discussed in Section 5. Simulation results contain plots of actual versus estimated trajectories and root mean square error(RMSE) values. Simulation results are discussed in Section 7 that concludes the thesis. Potential future works are also presented in Section 7.

2. LINEAR MINUMUM MEAN SQUARE ERROR ESTIMATION

2.1. Introduction

In estimation theory, in general, a noisy version of a signal is observed and its true value is to be estimated [8]. As convention, the quantity to be estimated is called the "signal", noisy version of the signal is called the "observation" and estimated value of the signal or output of the filter is called the "estimate". In this view, estimation may be regarded as the filtering out noise from the observation to get the true value of the signal. That is why terms "filter" and "estimator" or "filtering" and "estimation" are used interchangebly throughout the text. Additionally, the signal, the observation and the noise are assumed to be random. The signal to be estimated may either be constant or a time-varying quantity like the state of a dynamic system that will be elaborated in Section 3.

In stochastic estimation systems, noise signals have some degree of uncertainity represented by their probability density functions. Because of the random disturbances generated by the noise signals, probabilistic methods are needed. In MMSE filtering schemes, expected value of the square of the error between the signal and its estimate is to be minimized. In this view, the optimal filter(estimator) is the one that minimizes MSE between the signal and its estimate. Such a filter is called MMSE filter. Besides, if the filter, being a function of the observations, linearly combines the observations to calculate the estimate, it is called linear MMSE or shortly LMMSE filter. As proven in [8], optimal MSE filter(not necessarily linear) is the conditional mean of the signal given the observations. Difficulty in realizing infinite dimensional conditional pdf makes LMMSE filtering very attractive because LMMSE filter is easy to realize and implement with simple, finite dimensional algebra and statistics as described in Section 2.2. Besides, Kalman filter is the LMMSE state estimator for dynamic systems as proven in [8] and in Section 3.3.

Basically, as given in [8], LMMSE estimation relies on the principle of orthogonality. The LMMSE estimate of a random variable in terms of observations is such that

(i) The estimate is unbiased – the estimation error has zero mean, and

(ii) The estimation error is uncorrelated from the observation(s);

that is, they are orthogonal since zero mean uncorrelated random variables are orthogonal. This is the principle of orthogonality: In order for the error to have minimum norm, it has to be orthogonal to the observations. This is equivalent to stating that the estimate has to be orthogonal projection of the signal into the space spanned by the observations.

2.2. Linear Minumum Mean Square Error Estimation for Vector Random Variables

Consider the vector valued random variables $\boldsymbol{s}$ and $\boldsymbol{o}$, which are not necessarily Gaussian and zero-mean. The "best linear" estimate of $\boldsymbol{s}$ in terms $\boldsymbol{o}$ is obtained as follows. The criterion for "best" is the MMSE; that is, find the estimator

$$\hat{\boldsymbol{s}} = G\boldsymbol{o} + b \tag{2.1}$$

that minimizes the scalar MSE criterion, which in the multidimensional case is the expected value of the squared norm of the estimation error,

$$J = E[(\boldsymbol{s} - \hat{\boldsymbol{s}})'(\boldsymbol{s} - \hat{\boldsymbol{s}})] \tag{2.2}$$

According to the previous discussion, the linear MMSE estimator is such that the estimation error

$$\tilde{\boldsymbol{s}} = \boldsymbol{s} - \hat{\boldsymbol{s}} \tag{2.3}$$

is zero-mean (the estimate is unbiased) and orthogonal to the observation $\boldsymbol{o}$. In other words, the estimate $\hat{\boldsymbol{s}}$ is the orthogonal projection of the vector $\boldsymbol{s}$ into the space spanned by the (random components of the) observation vector $\boldsymbol{o}$. The unbiasedness requirement is

$$E[\tilde{\boldsymbol{s}}] = \bar{\boldsymbol{s}} - (G\bar{\boldsymbol{o}} + b) = 0 \tag{2.4}$$

where $\bar{\boldsymbol{s}}$ and $\bar{\boldsymbol{o}}$ are mean of $\boldsymbol{s}$ and $\boldsymbol{o}$, respectively. This requirement yields

$$b = \bar{\boldsymbol{s}} - G\bar{\boldsymbol{o}} \tag{2.5}$$

The estimation error is then

$$\tilde{\boldsymbol{s}} = \boldsymbol{s} - \bar{\boldsymbol{s}} - G(\boldsymbol{o} - \bar{\boldsymbol{o}}) \tag{2.6}$$

The orthogonality requirement is, in the multidimensional case, that each component of $\tilde{\boldsymbol{s}}$ be orthogonal to each component of $\boldsymbol{o}$. The orthogonality requirement can thus be written as

$$E[\tilde{\boldsymbol{s}}\boldsymbol{o}'] = E\big[\big(\boldsymbol{s} - \bar{\boldsymbol{s}} - G(\boldsymbol{o} - \bar{\boldsymbol{o}})\big)\boldsymbol{o}'\big] \tag{2.7}$$

$$= E\big[\big(\boldsymbol{s} - \bar{\boldsymbol{s}} - G(\boldsymbol{o} - \bar{\boldsymbol{o}})\big)(\boldsymbol{o} - \bar{\boldsymbol{o}})'\big]$$

$$= M_{so} - GM_{oo} = 0 \tag{2.8}$$

where M_{so} is cross covariance of $\boldsymbol{s}$ and $\boldsymbol{o}$ and M_{oo} is covariance of $\boldsymbol{o}$. The subtraction of $\bar{\boldsymbol{o}}$ from $\boldsymbol{o}$ in the transition from (2.7) to (2.8) could be done in view of the property (2.4) that $\tilde{\boldsymbol{s}}$ is zero mean. The solution for the weighting matrix G is thus

$$G = M_{so}{M_{oo}}^{-1} \tag{2.9}$$

The existence of the above requires the invertibility of M_{oo}, i.e., no linear dependence between the observations (or, equivalently, no redundant observations). Combining (2.5) and (2.9) yields the expression of the linear MMSE estimator for the multidimensional case as

$$\hat{\boldsymbol{s}} = \bar{\boldsymbol{s}} + M_{so}{M_{oo}}^{-1}(\boldsymbol{o} - \bar{\boldsymbol{o}}) \tag{2.10}$$

The matrix MSE is given after simple manipulations,

$$E[\tilde{\boldsymbol{s}}\tilde{\boldsymbol{s}}'] = M_{ss} - M_{so}{M_{oo}}^{-1}M_{os} \tag{2.11}$$

Equations (2.10) and (2.11) are the fundamental equations of vectoral LMMSE estimation.

2.3. Mean Square Error Estimation of Gaussian Random Vectors

As stated before, in any case the optimal MSE estimate of $\boldsymbol{s}$ given $\boldsymbol{o}$ is conditional mean of $\boldsymbol{s}$

$$\hat{\boldsymbol{s}} = E[\boldsymbol{s}|\boldsymbol{o}] \tag{2.12}$$

Given any number and type of observations, conditional mean gives the optimal MSE estimator, linear or nonlinear. Most of the time, conditional pdf is infinite dimensional and evaluation of it is prohibitively complicated. For these practical reasons, a LMMSE estimator mentioned above is preferred, although it is not optimal most of the time. However, if $\boldsymbol{s}$ and $\boldsymbol{o}$ are jointly Gaussian, (2.12) becomes linear:

$$\hat{\boldsymbol{s}} = E[\boldsymbol{s}|\boldsymbol{o}] = \bar{\boldsymbol{s}} + M_{so}{M_{oo}}^{-1}(\boldsymbol{o} - \bar{\boldsymbol{o}}) \tag{2.13}$$

and the corresponding conditional covariance matrix is

$$M_{ss|o} = M_{ss} - M_{so}{M_{oo}}^{-1}M_{os} \tag{2.14}$$

This follows from the fact that conditional pdf of $\boldsymbol{s}$ given $\boldsymbol{o}$ is Gaussian with mean (2.13) and covariance (2.14). Since above equations have the same form as (2.10) and (2.11), it can be said that the best MSE estimator for Gaussian random variables is identical to the best linear MSE estimator for arbitrarily distributed random variables with the same first and second-order moments. In other words, the linear estimator (2.10) is the overall best if the random variables are jointly Gaussian; otherwise it is only the best within the class of linear estimators.

3. STATE ESTIMATION FOR DYNAMIC SYSTEMS WITH RANDOM INPUTS

3.1. Introduction

Mobility tracking systems are nothing but dynamic systems with random state and observation models. In this section, general methodology for estimating the state of dynamic systems driven by random noise signals is given. It is assumed that related models exist(Formulation of mobility models will be given in Section 4). Chapter 4,5,6,10 and 11 of [9] and chapter 1 and 2 of [10] are extentesively used in the derivations below. State-space approach is adopted for modeling dynamic systems and discrete-time formulation of the problem is considered. Thus, difference equations are used to model the evolution of the system over time, and observations are assumed to be available at discrete times. For dynamic state estimation, the discrete-time approach is both widespread and convenient [10].

The state-space approach focuses attention on the state vector of a system. The state vector contains all relevant information required to describe the system under investigation. For example, in tracking problems this information could be related to the kinematic characteristics of the target. Alternatively, in an econometrics problem, it could be related to monetary flow, interest rates, inflation, and so forth [10].

The observation vector represents (noisy) observations that are related to the state vector. The observation vector is generally (but not necessarily) of lower dimension than the state vector. The state-space approach is convenient for handling multivariate data and nonlinear/non-Gaussian processes and it provides a significant advantage over traditional time-series techniques for these problems [10].

In order to analyze and make infrences about a dynamic system, at least two models are required. First, a model describing the evolution of the state with time (the system or dynamic model):

$$\boldsymbol{s}_{k+1} = f_k(\boldsymbol{s}_{k,}\boldsymbol{u}_k, \boldsymbol{w}_k) \tag{3.1}$$

where f_k is a known, possibly nonlinear function of the state vector $\boldsymbol{s}_k$, known control input $\boldsymbol{u}_k$ and process noise sequence $\boldsymbol{w}_k$. Process noise caters for any mismodelling effects or unforseen disturbances in the target motion model. State vector $\boldsymbol{s}_k$ has dimension n_s, so $\boldsymbol{s}_k \in \mathbb{R}^{n_s}$; $\mathbb{R}$ is a set of real numbers. k is the time index and $k \in \mathbb{N}$; $\mathbb{N}$ is the set of natural numbers. Here index k is assigned to a continous-time instant t_k , and the "sampling interval" $T_{k-1} = t_k - t_{k-1}$ may be time-dependent (i.e., a function of k) Second, a model relating the noisy observation to the state(the observation model):

$$\boldsymbol{o}_{k+1} = h_{k+1}(\boldsymbol{s}_{k+1}, \boldsymbol{\psi}_{k+1}) \tag{3.2}$$

where h_{k+1} is a known , possibly nonlinear function and $\boldsymbol{\psi}_{k+1}$ is a observation noise sequence.

In dynamic state estimation, as a general solution, we seek filtered estimate of $\boldsymbol{s}_{k+1}$ based on the sequence of all available information from initial time up to time $k+1$:

$$\boldsymbol{I}_{k+1} = \{\boldsymbol{O}_{k+1}, \boldsymbol{U}_k\} \tag{3.3}$$

where $\boldsymbol{O}_{k+1} = \{\boldsymbol{o}_0, \dots, \boldsymbol{o}_{k+1}\}$ is the cumulative set of observations from initial time up to time $k+1$ and $\boldsymbol{U}_k = \{\boldsymbol{u}_0, \dots, \boldsymbol{u}_k\}$ is the set of known inputs from initial time up to time k.

3.2. The Optimal Bayesian Mean Square Error Estimator

In the Bayesian state estimation, it is assumed that above models are available in a probabilistic form and the noise sequences $\boldsymbol{w}_k$ and $\boldsymbol{\psi}_{k+1}$ will be assumed to be white, with known probability density functions and mutally independent. The initial target state is assumed to have a known pdf $p(\boldsymbol{s}_0|\boldsymbol{I}_0)$ and also to be independent of noise sequences. Given these assumptions, in the Bayesian approach to optimal dynamic state estimation one attempts to construct the posterior probability density function of the state, based on all available information $\boldsymbol{I}_{k+1}$, $p(\boldsymbol{s}_{k+1}|\boldsymbol{I}_{k+1})$. Since this pdf embodies all available statistical information, it may be regarded to be the complete solution to the estimation problem. In principle, an optimal (with respect to any criterion in general, but we use MMSE criterion in this text for optimality condition, so conditional mean of the state is calculated) estimate of the state may be obtained from the posterior pdf. A measure of the accuracy of the estimate may also be obtained .

For many problems an estimate is required every time an observation is received. In this case a recursive filter is a convenient solution. Recursive filtering approach means that received data can be processed sequentially rather than as a batch, so that it is not necessary to store the complete data set nor to reprocess existing data if a new observation becomes available.

As shown in [9], above assumptions (white and mutually independent noise sequences) enable construction of $p(\boldsymbol{s}_{k+1}|\boldsymbol{I}_{k+1})$ from $p(\boldsymbol{s}_k|\boldsymbol{I}_k)$ recursively. There is no need to store growing information set $\boldsymbol{I}_k$, which grows with k. The nongrowing information state $p(\boldsymbol{s}_k|\boldsymbol{I}_k)$ is a complete substitute for the past data in the pdf of any present and future quantity related to the system (Markovian property). The related recursive filter has the following form:

$$p(\boldsymbol{s}_{k+1}|\boldsymbol{I}_{k+1}) = \frac{1}{c} p(\boldsymbol{o}_{k+1}|\boldsymbol{s}_{k+1}) p(\boldsymbol{s}_{k+1}|\boldsymbol{I}_k) \tag{3.4}$$

$p(\boldsymbol{s}_{k+1}|\boldsymbol{I}_k)$ is called prediction density (also referred to as the prior pdf) of the state at time $k+1$ and given by the Chapman-Kolmogorov equation [9]:

$$p(\boldsymbol{s}_{k+1}|\boldsymbol{I}_k) = \int p(\boldsymbol{s}_{k+1}|\boldsymbol{s}_k, \boldsymbol{u}_k)\, p(\boldsymbol{s}_k|\boldsymbol{I}_k) d\boldsymbol{s}_k \tag{3.5}$$

and c is the normalization constant as given in [10]

$$c = p(\boldsymbol{o}_{k+1}|\boldsymbol{I}_k) = \int p(\boldsymbol{o}_{k+1}|\boldsymbol{s}_{k+1})\, p(\boldsymbol{s}_{k+1}|\boldsymbol{I}_k) d\boldsymbol{s}_{k+1} \tag{3.6}$$

The optimal filter given in (3.4) consists of essentially two stages: prediction and update. The prediction stage uses the system model to predict the state pdf forward from one observation time to the next. Since the state is usually subject to unknown disturbances (modeled as random noise), prediction generally translates, deforms, and broadens the state pdf. The update operation uses the latest observation to modify (typically to tighten) the prediction pdf.

The recursive propagation of the posterior density, given by (3.4) , is only a conceptual solution in the sense that in general it cannot be determined analytically. The implementation of the conceptual solution requires the storage of the entire pdf which is, in general terms, equivalent to an infinite dimensional vector. Only in a restrictive set of cases, one of which the Kalman filter that will be described in Section 3.3, the posterior density can be exactly and completely characterized by a sufficient statistics of fixed and finite dimension. Since in most practical situtaions the analytic solution of (3.4) and (3.5) is intractible, one has to use approximations or suboptimal Bayesian algorithms like those mentioned in sections 3.4 and and 3.5.

3.3. The Kalman Filter

In addition to previously mentioned Bayesian assumptions, if noise sequences have Gaussian densities of known parameters and model functions are linear, the Kalman filter becomes the optimal finite-dimensional algorithm for recursive Bayesian state estimation. This is a highly desired case for state estimation because linear structure of Kalman filter, requiring only up to second order moments of state and observation vectors, make it very easy to implement computationally. In linear-Gaussian case, posterior density at every time step becomes Gaussian and hence is exactly and completely characterized by two parameters, its mean and covariance where mean is the optimal estimate. In this case, as will be elaborated, Kalman filter linearly propagates mean and covariance as new observations arrive. If noise signals are not Gaussian but models are still linear, Kalman filter is still valid and becomes the best linear MSE (LMMSE) filter for dynamic systems although not optimal overall. The derivation of the Kalman filter using chapter (5) of [9] is given as follows.

Suppose (3.1) and (3.2) can be rewritten as:

$$\boldsymbol{s}_{k+1} = A_k \boldsymbol{s}_k + B_k \boldsymbol{u}_k + \boldsymbol{w}_k \tag{3.7}$$

$$\boldsymbol{o}_{k+1} = H_{k+1}\boldsymbol{s}_{k+1} + \boldsymbol{\psi}_{k+1} \tag{3.8}$$

where A_k (of dimension n_s x n_s), B_k (of dimension n_s x n_u) and H_{k+1} (of dimension n_o x n_o) are known matrices defining the linear functions. Random sequences $\boldsymbol{w}_k$ and $\boldsymbol{\psi}_{k+1}$ are mutually independent zero-mean white Gaussian, with covariances Q_k and R_{k+1}, respectively. Note that the system and observation matrices A_k, B_k and H_{k+1} , as well as noise covariances Q_k and R_{k+1} , are allowed to be time-variant, which means that signal and observation process are allowed to be non-stationary. For stationary processes, these matrices will be constant.

The recursive linear Kalman filter algorithm that yields the state estimate at $k+1$, $\hat{\boldsymbol{s}} = \hat{\boldsymbol{s}}_{k+1|k+1}$ and its covariance, $M_{ss|o} = M_{k+1|k+1}$ can be directly obtained from the static LMMSE estimation equations (2.10) and (2.11)

$$\hat{\boldsymbol{s}} = \bar{\boldsymbol{s}} + M_{so}{M_{oo}}^{-1}(\boldsymbol{o} - \bar{\boldsymbol{o}}) \cong E[\boldsymbol{s}|\boldsymbol{o}] \tag{3.9}$$

$$E[\tilde{\boldsymbol{s}}\tilde{\boldsymbol{s}}'] = M_{ss|o} = M_{ss} - M_{so}{M_{oo}}^{-1}M_{os} \tag{3.10}$$

by substituting means $\bar{\boldsymbol{s}}$ and $\bar{\boldsymbol{o}}$, covariances M_{ss}, M_{oo} and cross covariances $M_{so}$and M_{os} of state vector $\boldsymbol{s}$ and observation vector $\boldsymbol{o}$ with the predicted state $\hat{\boldsymbol{s}}_{k+1|k}$, predicted observation $\hat{\boldsymbol{o}}_{k+1|k}$, predicted state covariance $M_{k+1|k}$, predicted observation covariance S_{k+1} and predicted cross covariances that are given below. Note that if noise sequences are Gaussian, as explained in 2.5., the approximation in (3.9) becomes exact and filter gives the overall optimal MSE solution, conditional mean. Also note that MSE related with the error $\tilde{\boldsymbol{s}} = \boldsymbol{s} - \hat{\boldsymbol{s}}$ in (3.10) is equivalent to the covariance of $\boldsymbol{s}$ given $\boldsymbol{o}$ because the estimate is equivalent to the conditional mean. Derivation of the predicted mean and covariances and explicit update equations are given below:

The predicted state $\hat{\boldsymbol{s}}_{k+1|k}$ substituting $\bar{\boldsymbol{s}}$ follows by applying the operator of expectation conditioned on $\boldsymbol{O}_k$ on the state equation (3.7). Since the process noise $\boldsymbol{w}_k$ is white and zero mean, this results in

$$E[\boldsymbol{s}_{k+1}|\boldsymbol{O}_k] = \hat{\boldsymbol{s}}_{k+1|k} = A_k\hat{\boldsymbol{s}}_{k|k} + B_k\boldsymbol{u}_k \tag{3.11}$$

Subtracting above from (3.7) yields the predicted state error

$$\tilde{\boldsymbol{s}}_{k+1|k} = \boldsymbol{s}_{k+1} - \hat{\boldsymbol{s}}_{k+1|k} = A_k\tilde{\boldsymbol{s}}_{k|k} + \boldsymbol{w}_k \tag{3.12}$$

Note the cancellation of input $\boldsymbol{u}_k$ in (3.12), since it is known, it has no effect on the estimation error.

The predicted state covariance $M_{k+1|k}$ substituting M_{ss} becomes

$$E\left[\tilde{\boldsymbol{s}}_{k+1|k}\tilde{\boldsymbol{s}}_{k+1|k}{}^T|\boldsymbol{O}_k\right] = M_{k+1|k} = A_kM_{k|k}A_k^T + Q_k \tag{3.13}$$

The cross-terms in (3.13) are canceled due to the fact that $\boldsymbol{w}_k$ is zero mean and white and, thus orthogonal to $\tilde{\boldsymbol{s}}_{k|k}$.

The predicted observation $\hat{\boldsymbol{o}}_{k+1|k}$ substituting $\bar{\boldsymbol{o}}$ follows similarly by taking expected value of (3.8) conditioned on $\boldsymbol{O}_k$. Since observation noise is zero mean and white this becomes

$$E[\boldsymbol{o}_{k+1}|\boldsymbol{O}_k] = \hat{\boldsymbol{o}}_{k+1|k} = H_{k+1}\hat{\boldsymbol{s}}_{k+1|k} \tag{3.14}$$

Subtracting the above from (3.8) yields the predicted observation error, also called innovation or observation residual, $\boldsymbol{r}_{k+1}$

$$\tilde{\boldsymbol{o}}_{k+1|k} = \boldsymbol{o}_{k+1} - \hat{\boldsymbol{o}}_{k+1|k} = \boldsymbol{r}_{k+1} = H_{k+1}\tilde{\boldsymbol{s}}_{k+1|k} + \boldsymbol{\psi}_{k+1} \tag{3.15}$$

The predicted observation covariance S_{k+1}substituting M_{oo}, also called innovation covariance becomes

$$E[\boldsymbol{r}_{k+1}\boldsymbol{r}_{k+1}{}^T|\boldsymbol{O}_k] = S_{k+1} = H_{k+1}M_{k+1|k}H_{k+1}{}^T + R_{k+1} \tag{3.16}$$

The predicted cross covariance between the state and observation becomes

$$E\left[\tilde{\boldsymbol{s}}_{k+1|k}\tilde{\boldsymbol{o}}_{k+1|k}{}^{T}|\boldsymbol{O}_k\right] = M_{so} = M_{k+1|k}H_{k+1}^{T} \quad (3.17)$$

Then the filter gain $M_{so}M_{oo}{}^{-1} = W_{k+1}$ becomes

$$W_{k+1} = M_{k+1|k}H_{k+1}^{T}S_{k+1}{}^{-1} \quad (3.18)$$

Finally updated state $\hat{\boldsymbol{s}} = \hat{\boldsymbol{s}}_{k+1|k+1}$ in (3.9) and updated state covariance $M_{xx|o} = M_{k+1|k+1}$ in (3.10) becomes

$$\hat{\boldsymbol{s}}_{k+1|k+1} = \hat{\boldsymbol{s}}_{k+1|k} + W_{k+1}\boldsymbol{r}_{k+1} \quad (3.19)$$

$$M_{k+1|k+1} = M_{k+1|k} - W_{k+1}S_{k+1}W_{k+1}{}^{T} \quad (3.20)$$

3.4. The Extended Kalman Filter

Reality manifests itself as being very complex: nonlinear, non-Gaussian, non-stationary, and with continuous-valued target states. Therefore, in most practical situations, the optimal filters, like linear-Gaussian Kalman filter mentioned above cannot be applied. Instead, one is forced to use approximate or suboptimal solutions [10].

In one group of mobility models discussed in this text, observation equation is nonlinear. Hence, linear predicted mean and covariance calculations in the traditional Kalman filter discussed above can not be applied for these models to obtain optimal estimate. The EKF is one of the commonly used approximate solutions that can be applied when any of the model equations is nonlinear. The main future of the EKF is that it linearizes the nonlinear functions in the model equations so that linear equations of the Kalman filter can be used. EKF is derived for nonlinear systems with additive noise; that is, for spacial case of (3.1) and (3.2) given here:

$$\boldsymbol{s}_{k+1} = f_k(\boldsymbol{s}_k) + \boldsymbol{w}_k \quad (3.21)$$

$$\boldsymbol{o}_{k+1} = h_{k+1}(\boldsymbol{s}_{k+1}) + \boldsymbol{\psi}_{k+1} \tag{3.22}$$

where, for simplicity, it is assumed that there is no control input. Random sequences $\boldsymbol{w}_k$ and $\boldsymbol{\psi}_{k+1}$ are mutually independent, zero-mean and white with covariances Q_k and R_{k+1} , respectively.

3.4.1. Linearization

The EKF is based on the assumption that local linearization of the (3.21) and (3.22) by the first term of their Taylor Series expansion may be a sufficient description of the nonlinearity. Local linearizations $\hat{A}_k$ of f_k and $\widehat{H}_{k+1}$ of h_{k+1} are defined as Jacobians evaluated at $\hat{\boldsymbol{s}}_{k|k}$ and $\hat{\boldsymbol{s}}_{k+1|k}$, respectively; that is:

$$\hat{A}_k = [\boldsymbol{\nabla}_{\mathbf{s}_\mathrm{k}} f_k^{\;T}(\mathbf{s}_\mathrm{k})]^T|_{\mathbf{s}_\mathrm{k}=\hat{s}_{k|k}} \tag{3.23}$$

$$\widehat{H}_{k+1} = [\boldsymbol{\nabla}_{\mathbf{s}_{\mathrm{k+1}}} h_{k+1}^T(\mathbf{s}_{\mathrm{k+1}})]^T|_{\mathbf{s}_{\mathrm{k+1}}=\hat{s}_{k+1|k}} \tag{3.24}$$

where

$$\boldsymbol{\nabla}_{\mathbf{s}_\mathrm{k}} = \left[\frac{\partial}{\partial \mathbf{s}_\mathrm{k}[1]} \cdots \frac{\partial}{\partial \mathbf{s}_\mathrm{k}[\mathrm{n_s}]}\right]^\mathrm{T} \tag{3.25}$$

with $\mathbf{s}_\mathrm{k}[\mathrm{i}], \mathrm{i} = 1, \ldots, \mathrm{n_s}$, being the ith component of $\mathbf{s}_\mathrm{k}$. An element of say, $\widehat{H}_{k+1}$ is given by:

$$\widehat{H}_{k+1}[i,j] = \frac{\partial \mathrm{h}_{\mathrm{k+1}}[\mathrm{i}]}{\partial \mathbf{s}_{\mathrm{k+1}}[\mathrm{j}]}|_{\mathbf{s}_{\mathrm{k+1}}=\hat{s}_{k+1|k}} \tag{3.26}$$

where $\mathrm{h}_{\mathrm{k+1}}[\mathrm{i}]$ denotes the ith component of vector $\mathrm{h}_\mathrm{k}(\mathbf{s}_\mathrm{k})$.

3.4.2. Filter Equations

As shown below, except calculations of $\hat{A}_k$ and $\widehat{H}_{k+1}$ as given above, linear filter equations for the EKF is nearly the same as traditional Kalman filter derived above. The only difference is usage of $\hat{A}_k$ and $\widehat{H}_{k+1}$. Derivation of these equations are given in detail in [9]:

$$\hat{\boldsymbol{s}}_{k+1|k} = f_k(\hat{\boldsymbol{s}}_{k|k}) \tag{3.27}$$

$$M_{k+1|k} = Q_k + \hat{A}_k M_{k|k} \hat{A}_k^T \tag{3.28}$$

$$S_{k+1} = \widehat{H}_{k+1} M_{k+1|k} \widehat{H}_{k+1}^{\ T} + R_{k+1} \tag{3.29}$$

$$W_{k+1} = P_{k+1|k} \widehat{H}_{k+1}^{\ T} S_{k+1}^{\ -1} \tag{3.30}$$

$$\hat{\boldsymbol{s}}_{k+1|k+1} = \hat{\boldsymbol{s}}_{k+1|k} + W_{k+1}\left(\boldsymbol{o}_{k+1} - h_k\big(\hat{\boldsymbol{s}}_{k+1|k}\big)\right) \tag{3.31}$$

$$M_{k+1|k+1} = M_{k+1|k} - W_{k+1} S_{k+1} W_{k+1}^{\ T} \tag{3.32}$$

3.5. The Unscented Kalman Filter

If the nonlinearity in models (3.21) and (3.22) is very severe, approximation error due to linearization will be more pronounced and performance of the EKF will be degraded significantly. In this case, prediction errors resulting from linearization errors will make the EKF perform poorly. Even worser, if the nonlinear functions are discontinous, EKF cannot be applied at all. Additionally, most of the time calculation of Jacobians in each time step may become cumbersome and time-consuming.

The UKF proposed in [2] is one of the solutions proposed to overcome the described problems with the EKF. UKF is a sampling based approach that approximates the posterior density by a set of small number of deterministically chosen samples. Instead of linearization, it uses the unscented transform(UT) in Kalman filter framework to calculate predicted mean and covariances. UKF is applicable even when there is a discontinuity in nonlinear fucnstions $\mathbf{f}$ and $\boldsymbol{h}$, because no explicit calculation of Jacobians is necessary.

As proved in [2], the sample points completely capture the true mean and covariance of the posterior density and when propagated through a nonlinear transform, their sample mean and covariance capture the true mean and covariance up to the third order of nonlinearity (with errors introduced in the fourth and higher orders). Strong approximation features of UT make UKF perform better than EKF most of the time [2].

3.5.1. The Unscented Transform

The UT is a method for calculating the statistics of a random variable that undergoes a nonlinear transformation. Consider propagating a random vector $\boldsymbol{a}$, with mean $\bar{\boldsymbol{a}}$ and covariance P_a , through an arbitrary nonlinear function $g : \mathbb{R}^{\boldsymbol{n_a}} \rightarrow \mathbb{R}^{\boldsymbol{n_b}}$, to produce a random vector $\boldsymbol{b} = g(\boldsymbol{a})$. The first two moments of $\boldsymbol{b}$ are computed using UT as follows. First, $2n_{\boldsymbol{a}} + 1$ weighted sample points $(X_{\boldsymbol{i}}, K_{\boldsymbol{i}})$ are deterministically chosen so that they completely describe (capture) the true mean $\bar{\boldsymbol{a}}$ and covariance $\boldsymbol{M}_{aa}$ of $\boldsymbol{a}$. A scheme that satisfies this requirement is:

$$X_0 = \bar{\boldsymbol{a}} \qquad K_0 = \frac{\kappa}{(n_a+\kappa)} \qquad i = 0$$

$$X_i = \bar{\boldsymbol{a}} + \left(\sqrt{(n_a+\kappa)\boldsymbol{P}_a}\right)_i \qquad K_i = \frac{1}{2(n_a+\kappa)} \qquad i = 1, \dots, n_a$$

$$X_i = \bar{\boldsymbol{a}} - \left(\sqrt{(n_a+\kappa)\boldsymbol{P}_a}\right)_i \qquad K_i = \frac{1}{2(n_a+\kappa)} \qquad i = n_1 + 1, \dots, 2n_a$$

where κ is a scaling parameter (such that $\kappa + n_a \neq 0$) and $\left(\sqrt{(n_a+\kappa)M_{aa}}\right)_i$ is the ith row of the matrix square root$V$ of $(n_a+\kappa)M_{aa}$, such that $(n_a+\kappa)M_{aa} = V^T V$. The weights are normalized; that is, satisfy $\sum_{i=0}^{2n_a} K_i = 1$.

Now each sample point is propagated through the nonlinear function g**:**

$$Z_i = g(X_i) \;\; (i = 0,1, \dots, 2n_a) \tag{3.33}$$

and the first two moments of $\boldsymbol{b}$ are computed as follows:

$$\overline{\boldsymbol{b}} = \sum_{i=0}^{2n_a} K_i Z_i \tag{3.34}$$

$$M_{bb} = \sum_{i=0}^{n_a} K_i (Z_i - \overline{\boldsymbol{b}})(Z_i - \overline{\boldsymbol{b}})^T \tag{3.35}$$

3.5.2. Filter Equations

After calculating sample points X_k^i and their weigths K_k^i , $i = 0, \dots, N-1$ using UT, UT is further used to calculate predicted estimates and covariances from the sample points and their weights. Then, predicted statistics are used in the linear Kalman filter framework to calculate updated state estimate and covariance. Resulting equations that are derived in [10] are presented below. It is assumed that sigma points are calculated.

$$X_{k+1|k}^i = f_{\mathrm{k}}(X_k^i) \tag{3.36}$$

$$\hat{\boldsymbol{s}}_{k+1|k} = \sum_{i=0}^{N-1} K_k^i \,.\, X_{k+1|k}^i \tag{3.37}$$

$$M_{k+1|k} = Q_k + \sum_{i=0}^{N-1} K_k^i \left[X_{k+1|k}^i - \hat{\boldsymbol{s}}_{k+1|k}\right]\left[X_{k+1|k}^i - \hat{\boldsymbol{s}}_{k+1|k}\right]^T \tag{3.38}$$

$$\hat{\boldsymbol{o}}_{k+1|k} = \sum_{i=0}^{N-1} K_k^i \,.\, h_{k+1}(X_{k+1|k}^i) \tag{3.39}$$

$$M_{so} = \sum_{i=0}^{N-1} K_k^i \left[X_{k+1|k}^i - \hat{\boldsymbol{s}}_{k+1|k}\right]\left[h_{k+1}\left(X_{k+1|k}^i\right) - \hat{\boldsymbol{o}}_{k+1|k}\right]^T \tag{3.40}$$

$$M_{oo} = \sum_{i=0}^{N-1} K_k^i \left[h_{k+1}\left(X_{k+1|k}^i\right) - \hat{\boldsymbol{o}}_{k+1|k}\right]\left[h_{k+1}\left(X_{k+1|k}^i\right) - \hat{\boldsymbol{o}}_{k+1|k}\right]^T \tag{3.41}$$

$$S_{k+1} = M_{oo} + R_{k+1} \tag{3.42}$$

$$W_{k+1} = M_{so} S_{k+1}^{-1} \tag{3.43}$$

$$\hat{\boldsymbol{s}}_{k+1|k+1} = \hat{\boldsymbol{s}}_{k+1|k} + W_{k+1}(\boldsymbol{o}_{k+1} - \hat{\boldsymbol{o}}_{k+1|k}) \tag{3.44}$$

$$M_{k+1|k+1} = M_{k+1|k} - W_{k+1}S_{k+1}{W_{k+1}}^T \quad (3.45)$$

3.6. Adaptive Estimation For Multiple Model Systems

In the models considered above, the only uncertainities consisted of additive white noise signals with known statistical properties. In other words, the system model, consisting of the state transition matrix, the input gain and input(if any), the observation matrix and noise covariances were all assumed to be known. However, in some systems like multiple model systems, it is assumed that the system obeys one of a finite number of models. Such systems are called hybrid: They have both continuous noise uncertainities and discrete uncertainities - model or mode or operating regime uncertainities [9]. As stated in Section 5, multiple model systems are common in tracking systems that have Markovian manuevering target dynamics.

The purpose of this section is to present adaptive state estimation technique for multiple model systems that can adapt itself to model changes. A Bayesian framework is used: Starting with prior probabilities of each model being correct(i.e., the system is in a particular mode), the corresponding posterior probabilities are obtained [9]. The dynamic situation of switching models or mode jumping is considered, so the system undergoes transitions from one mode to another.

The dynamic multiple model system is modeled by the equations

$$\boldsymbol{s}_k = A[D_k]\boldsymbol{s}_{k-1} + \boldsymbol{w}_{k-1}[D_k] \quad (3.46)$$

$$\boldsymbol{o}_k = H[D_k]\boldsymbol{s}_k + \boldsymbol{\psi}_k[D_k] \quad (3.47)$$

where D_k denotes the mode or model at time k – in effect during the sampling period ending at k, $A[D_k]$ and $H[D_k]$ denote the transition and observation matrices of the system at mode D_k and $\boldsymbol{w}_{k-1}[D_k]$ and $\boldsymbol{\psi}_k[D_k]$ denote noise components of the system at mode D_k. That is, the structure of the system and/or the statistics of the noise signals might be different from model to model.

The mode at time k is assumed to be among the possible r modes

$$D_k \in \{D^j\}_{j=1}^{r} \tag{3.48}$$

The lth mode history – or sequence of models – through time k is denoted as

$$D_{k,l} = \{D^{i_{1,l}}, \dots, D^{i_{k,l}}\} \quad l = 1, \dots, r^k \tag{3.49}$$

where $i_{k,l} = 1, \dots r$ is the model index at time k from history l. Note that the number of histories increases exponentially with time. For example, with $r = 2$ one has at time $k = 2$ $r^k = 4$ possible sequences(histories).

It is assumed that the mode(model) switching – that is, the mode jump process – is a Markov process(Markov chain) with known transition probabilities

$$p_{ij} = P\{D_k = D^j | D_{k-1} = D^i\} \tag{3.50}$$

These mode transition probabilities will be assumed time-invariant and independent of the base state. In other words, this is a homogeneous Markov chain.

The event that model j is in effect at time k is denoted as

$$D_k^j = \{D_k = D^j\} \tag{3.51}$$

The conditional probability of the lth history is given by

$$q^{k,l} = P\{D_{k,l} | \boldsymbol{O}_k\} \tag{3.52}$$

The lth sequence of model through time k can be written as

$$D_{k,l} = \{D_{k-1,z}, D_k^j\} \tag{3.53}$$

where sequence z through $k-1$ is its parent sequence and D^j is its last element. Then, in view of the Markov property,

$$P\{D_k^j | D_{k-1,z}\} = P\{D_k^j | D_{k-1}^i\} = p_{ij} \tag{3.54}$$

where $i = z_{k-1}$, the index of the last model in the parent sequence z through $k-1$.

The conditional pdf of the state at k is obtained using the total probability theorem with an exponentially increasing number of terms

$$p(\boldsymbol{s}_k | \boldsymbol{O}_k) = \sum_{l=1}^{r^k} p(\boldsymbol{s}_k | D_{k,l}, \boldsymbol{O}_k) q^{k,l} \tag{3.55}$$

where, as derived in [9],

$$q^{k,l} = \frac{1}{c}\, p(\boldsymbol{o}_k | D_{k,l}, \boldsymbol{O}_{k-1}) p_{ij} q^{k-1,z} \tag{3.56}$$

where $i = z_{k-1}$ is the last model of the parent sequences. (3.56) shows that conditioning on the entire past history is needed even if the random parameters are Markov. Besides, according to (3.55), to each mode sequence a filter is to be matched. Hence, an exponentially increasing number of filters are needed to estimate the sate, which makes the optimal approach impractical.

3.6.1. The First Order Generalized Pseudo-Bayesian Multiple Model Estimator

The only way to avoid the exponentially increasing number of histories is to use suboptimal techniques. The generalized pseudo-Bayesian(GPB) approaches combine histories of models that differ in “older” models to decrease number of histories used in summation. For example, the first order GPB, denoted as GPB1, considers only the possible models in the last sampling period. This algorithm requires only r filters to operate in parallel.

In GPB1, at time k the state estimate is computed under each possible current model, so a total of r possibilities(hypotheses) are considered. All histories that differ in "older" models are combined together. The total probability theorem is thus used as follows:

$$p(\boldsymbol{s}_k|\boldsymbol{O}_k) = \sum_{j=1}^{r} p\left(\boldsymbol{s}_k|D_k^j, \boldsymbol{o}_k, \boldsymbol{O}_{k-1}\right)q_k^j$$

$$= \sum_{j=1}^{r} p\left(\boldsymbol{s}_k|D_k^j, \boldsymbol{o}_k, \hat{\boldsymbol{s}}_{k-1|k-1}, M_{k-1|k-1}\right)q_k^j \tag{3.57}$$

where q_k^j is the probability of mode D^j being active at time k. Thus at time $k-1$ there is a single lumped estimate $\hat{\boldsymbol{s}}_{k-1|k-1}$ and the associated covariance that approximately summarizes the past $\boldsymbol{O}_{k-1}$. From this, one carries out the prediction to time k and the update at time k under r possibilities:

$$\hat{\boldsymbol{s}}_{k|k}^j = \hat{\boldsymbol{s}}\left[k|k; D_k^j, \hat{\boldsymbol{s}}_{k-1|k-1}, M_{k-1|k-1}\right] \qquad j = 1, \dots, r \tag{3.58}$$

$$M_{k|k}^j = M\left[k|k; D_k^j, M_{k-1|k-1}\right] \qquad j = 1, \dots, r \tag{3.59}$$

After the update, the estimates are combined with the weightings q_k^j, resulting in the new combined estimate $\hat{\boldsymbol{s}}_{k|k}$. In other words, in each cycle, each of r model-matched filters runs to produce mode-conditioned state $\hat{\boldsymbol{s}}_{k|k}^j$, the associated covariance $M_{k|k}^j$ and mode likelihood function λ_k^j. Likelihood function is used to calculate mode probabilityq_k^j as presented below. Finally, at the end of each cycle, r hypotheses are merged into a single hypothesis as follows:

$$\hat{\boldsymbol{s}}_{k|k} = \sum_{j=1}^{r} \hat{\boldsymbol{s}}_{k|k}^j \, q_k^j \tag{3.60}$$

$$M_{k|k} = \sum_{j=1}^{r} q_k^j \left\{M_{k|k}^j + \left[\hat{\boldsymbol{s}}_{k|k}^j - \hat{\boldsymbol{s}}_{k|k}\right]\left[\hat{\boldsymbol{s}}_{k|k}^j - \hat{\boldsymbol{s}}_{k|k}\right]'\right\} \tag{3.61}$$

q_k^j is derived in [9] and is given by

$$q_k^j = \frac{1}{c}\lambda_k^j \sum_{i=1}^{r} p_{ij}\, q_{k-1}^i \tag{3.62}$$

where p_{ij} is the known mode transition probability and likehood function λ_k^j and normalization constant c is given as follows:

$$\lambda_k^j = p\left(\boldsymbol{o}_k | D_k^j, \widehat{\boldsymbol{x}}_{k-1|k-1}, M_{k-1|k-1}\right) \tag{3.63}$$

$$c = \sum_{j=1}^{r} \lambda_k^j \sum_{i=1}^{r} p_{ij}\, q_{k-1}^i \tag{3.64}$$

4. CELLULAR TRACKING SYSTEM

4.1. Introduction

In tracking systems, target state typically consists of kinematic components (position, velocity, acceleration, and so forth) and measurements are noise-corrupted observations related to the target state. The kinematic observations are collected by various type of sensors and include target range, azimuth(bearing), elevation and range rate(extracted from Doppler frequency). State model represents dynamic motion of the target(evolution of the state vector) with respect to time and observation(measurement) model relates the observation(measurement) vector to the state vector. In this section, a special type of tracking system for cellular communication networks called “cellular tracking system” is presented. In general, this system consists of a state model and two observation models that form the basis for the cellular tracking algorithms discussed in Section 5. These models are presented in Sections 4.2 and 4.3.

4.2. State Model

The state model of the cellular tracking system is discussed in [1] and [7] and is a special combination of the time-correlated model given in [3] and the semi-Markovian model given in [4]. According to this combined model, movement of mobile users is similar to dynamic motion of manuevering targets in tactical weapon systems. This model can capture a wide range of mobility scenarios, including sudden stops and changes in acceleration. The state model is developed below.

The status of the mobile station at time t is defined by a state vector

$$\boldsymbol{s}(t) = [x(t), \dot{x}(t), \ddot{x}(t), y(t), \dot{y}(t), \ddot{y}(t)]^T \tag{4.1}$$

where $x(t)$ and $y(t)$ specify the position, $\dot{x}(t)$ and $\dot{y}(t)$ specify the velocity, and $\ddot{x}(t)$ and $\ddot{y}(t)$ specify the acceleration in the x and y directions in a two-dimensional grid. The state vector can be written more compatctly as

$$\boldsymbol{s}(t) = [\boldsymbol{x}(t), \boldsymbol{y}(t)]^T \tag{4.2}$$

where $\boldsymbol{x}(t) = [x(t), \dot{x}(t), \ddot{x}(t)]^T$ and $\boldsymbol{y}(t) = [y(t), \dot{y}(t), \ddot{y}(t)]^T$.

4.2.1. Manuever Modeling

In tracking environments, deviation of targets from straight line constant velocity movement is regarded as manuevering. Road turns, traffic lights, evasive manuevers, accelerations due to atmospheric turbulence, sudden stops and accelerations may be viewed as manuevers. Since acceleration accounts for the target deviations from constant velocity movement, it is termed as the target manuever variable [3].

The acceleration vector, target manuever variable, $\boldsymbol{a}(t) = [\ddot{x}(t), \ddot{y}(t)]^T$, is modeled as follows:

$$\boldsymbol{a}(t) = \boldsymbol{u}(t) + \boldsymbol{r}(t)\ , \tag{4.3}$$

where $\boldsymbol{u}(t) = \left[u_x(t), u_y(t)\right]^T$ is a discrete-valued Markovian process and $\boldsymbol{r}(t) = \left[r_x(t), r_y(t)\right]^T$ is a zero-mean Gaussian process chosen to cover the gaps between adjacent levels(states) of the process $\boldsymbol{u}(t)$.

The processes $u_x(t)$ and $u_y(t)$ are modeled as semi-Markov processes that take values from a finite set of acceleration levels(states) $L = \{l_1, \dots, l_m\}$. Thus, the process $\boldsymbol{u}(t)$ takes values in the set $D = L\ X\ L$. A semi-Markov process differs from a Markov process in that the duration of time in one state prior to switching to another state is itself a random variable.

$\boldsymbol{u}(t)$ is used to model sudden and unexpected target-controlled changes in the acceleration. It helps the target mobility model to view the target as if it is responding to deterministic acceleration commands given by the pilot or mobile user, that is why it is also termed as command process. Hence, commands correspond to discrete "states" of a semi-Markov process which are selected according to the transition probabilities of the Markov process and remain active according to a seperate random variable.

$\boldsymbol{r}(t)$ acts as time-correlated noisy acceleration component and covers gaps between the discrete acceleration states. Generally, it accounts for non-target-controlled environmental manuevers. This type of manuever capability can be satisfactorily specified by two quantities: the variance, or magnitude of the target manuever and the time constant, or duration of the target manuever.

$\boldsymbol{r}(t)$ is correlated in time; namely, if a target is accelerating at time t, it is likely to be accelerating at time $t + \tau$ for sufficiently small τ. For example, a lazy turn will often give rise to correlated acceleration inputs for up to one minute, evasive manuevers will provide correlated acceleration inputs for periods between ten and thirty seconds, and atmospheric turbulence may provide correlated acceleration inputs for one or two seconds. A typical representative model of the correlation function for one dimensional $r(t)$ is given by:

$$R_r(\tau) = E[r(t)r(t+\tau)] = {\sigma_1}^2 e^{-\alpha|\tau|} \;,\; \alpha \geq 0 \tag{4.4}$$

where ${\sigma_1}^2$ is the variance of the target acceleration and α is the reciprocal of the acceleration time constant. For example, $\alpha = 1/60$ for a lazy turn, $\alpha = 1/20$ for an evasive manuever, and $\alpha = 1$ for atmospheric turbulence.

The variance ${\sigma_1}^2$ of target acceleration is calculated as follows: The target can accelerate at a maximum rate $A_{max}(-A_{max})$ and will do each with a probability P_{max}. The target undergoes no acceleration with a probability P_0, and will accelerate between the limits $-A_{max}$ and A_{max} according to the appropriate uniform distribution. The variance ${\sigma_1}^2$ of the resulting acceleration probability density model is

$$\sigma_1{}^2 = \frac{A_{max}^2}{3}[1 + 4P_{max} - P_0]\ . \tag{4.5}$$

This model has been utilized in tracking simulations and has been shown to provide a satisfactory representation of the target's instantaneous manuever characteristics.

Utilizing the correlation function $R_r(\tau)$, the acceleration $r(t)$ may be expressed in terms of zero-mean white noise by the Wiener-Kolmogorov procedure(by passing a zero-mean white Gaussian random process through a single pole filter). The Laplace transform of $R_r(\tau)$ is given by :

$$R(s) = L\{R_r(\tau)\} = \frac{-2\alpha\sigma_1{}^2}{(s-\alpha)(s+\alpha)} = H(s)H(-s)W(s) \tag{4.6}$$

where

$$H(s) = \frac{1}{s+\alpha} \tag{4.7}$$

$$W(s) = 2\alpha\sigma_1{}^2 \tag{4.8}$$

The quantity $H(s)$ is the transform of the single pole filter for $r(t)$ and $W(s)$ is the transform of the white noise $w(t)$ that drives $r(t)$. Therefore the resulting equations in time domain are:

$$\dot{r}(t) = -\alpha r(t) + w(t) \tag{4.9}$$

where $\sigma_w^2(\tau)$, the correlation function of the white noise input, satisfies

$$\sigma_w^2(\tau) = 2\alpha\sigma_1^2\delta(\tau) \tag{4.10}$$

As a result, two dimensional $\boldsymbol{r}(t)$ has the form of

$$\dot{\boldsymbol{r}}(t) = -\alpha\boldsymbol{r}(t) + \boldsymbol{w}(t) \tag{4.11}$$

with the autocorrelation function

$$R_r(\tau) = E[\boldsymbol{r}(t)\boldsymbol{r}(t+\tau)^T] = {\sigma_1}^2 e^{-\alpha|\tau|} I_2 \;\; , \;\; \alpha \geq 0 \tag{4.12}$$

where I_2 is 2 x 2 identity matrix.

To sum up, target manuever variable $\boldsymbol{a}(t) = \boldsymbol{u}(t) + \boldsymbol{r}(t)$ is represented by a correlated gaussian noise with randomly switching mean. Random means correspond to states of command process $\boldsymbol{u}(t)$ and time-correlated behaviour of the $\boldsymbol{a}(t)$ between random means comes from $\boldsymbol{r}(t)$.

4.2.2. State Model Equations

From (4.3) and (4.11) we know that

$$\dot{a}_x(t) = \dot{r}_x(t) = -\alpha r_x(t) + w_x(t) \tag{4.13}$$

Above equation can be extended as follows to include command process $u_x(t)$

$$\dot{a}_x(t) = -\alpha r_x(t) - \alpha u_x(t) + \alpha u_x(t) + w_x(t) = -\alpha a_x(t) + \alpha u_x(t) + w_x(t) \tag{4.14}$$

Putting $\ddot{x}(t) = a_x(t)$ in (4.14) we get following linear system equation describing the state evolution in the x-direction:

$$\dot{\boldsymbol{x}}(t) = \tilde{A}_1 \boldsymbol{x}(t) + \tilde{B}_1 u_x(t) + \tilde{C}_1 w_x(t) \tag{4.15}$$

where

$$\tilde{A}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\alpha \end{bmatrix} \; , \; \tilde{B}_1 = \begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix} \; , \; \tilde{C}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.16}$$

Similarly, the state equation for the y-direction is given by

$$\dot{\boldsymbol{y}}(t) = \tilde{A}_1 \boldsymbol{y}(t) + \tilde{B}_1 u_y(t) + \tilde{C}_1 w_y(t) \tag{4.17}$$

Combining (4.15) and (4.17) yields the overall state equation

$$\dot{\boldsymbol{s}}(t) = \tilde{A}\boldsymbol{s}(t) + \tilde{B}\boldsymbol{u}(t) + \tilde{C}\boldsymbol{w}(t) \tag{4.18}$$

where

$$\tilde{A} = I_2 \otimes \tilde{A}_1\ ,\ \ \tilde{B} = I_2 \otimes \tilde{B}_1\ \ ,\ \ \tilde{C} = I_2 \otimes \tilde{C}_1 \tag{4.19}$$

and $\otimes$ denotes Kronecker matrix product.

By sampling the state once every T time units, the system can be characterized in terms of the discrete-time state vector $\boldsymbol{s}_n = \boldsymbol{s}(nT)$. The corresponding discrete-time state equation is given by

$$\boldsymbol{s}_{k+1} = A\boldsymbol{s}_k + B\boldsymbol{u}_k + \boldsymbol{w}_k \tag{4.20}$$

where

$$A = e^{\tilde{A}T}\ \ ,\ \ B = \int_t^{t+T} e^{\tilde{A}(t+T-\tau)}\tilde{B}d\tau \tag{4.21}$$

$$\boldsymbol{w}_k = \int_{kT}^{(k+1)T} e^{\tilde{A}(t+T-\tau)}\tilde{C}\boldsymbol{w}(\tau)d\tau\ . \tag{4.22}$$

The vector $\boldsymbol{w}_k$ is a 6 x 1 column vector. The process $\boldsymbol{w}_k$ is a discrete –time zero mean, stationary Gaussian process with autocorrelation function $R_{\boldsymbol{w}}(k) = \delta_k Q$, where $\delta_0 = 1$ and $\delta_k = 0$ when $k \neq 0$ which means that $\boldsymbol{w}_k$ is a white process. The matrix Q , the covariance matrix of $\boldsymbol{w}_k$, is given as follows:

$$Q = 2\alpha{\sigma_1}^2 I_2 \otimes Q_1(T) \tag{4.23}$$

where $Q_1(T) = [q_{ij}]$ is a symetric 3 x 3 matrix with upper triangular entries given as follows:

$q_{11} = (1 - e^{-2\alpha T} + 2\alpha T + 2\alpha^3 T^3/3 - 2\alpha^2 T^2 - 4\alpha T e^{-\alpha T})/(2\alpha^5)$,
$q_{12} = (e^{-2\alpha T} + 1 - 2e^{-\alpha T} + 2\alpha T e^{-\alpha T} - 2\alpha T + \alpha^2 T^2)/(2\alpha^4)$,
$q_{13} = (1 - e^{-2\alpha T} - 2\alpha T e^{-\alpha T})/(2\alpha^3)$,
$q_{22} = (4e^{-\alpha T} - 3 - e^{-2\alpha T} + 2\alpha T)/(2\alpha^3)$,
$q_{23} = (e^{-2\alpha T} + 1 - 2e^{-\alpha T})/(2\alpha^2)$,
$q_{33} = (1 - e^{-2\alpha T})/(2\alpha)$.

The matrices A and B in (4.21) are given by:

$$A = I_2 \otimes A_1(T) \ , \ B = I_2 \otimes B_1(T) \tag{4.24}$$

where

$$A_1(T) = \begin{bmatrix} 1 & T & a \\ 0 & 1 & b \\ 0 & 0 & e^{-\alpha T} \end{bmatrix} \ , \ B_1(T) = \begin{bmatrix} c \\ \alpha a \\ \alpha b \end{bmatrix} \ \text{with}$$

$a = (-1 + \alpha T + e^{-\alpha T})/\alpha^2$, $b = (1 - e^{-\alpha T})/\alpha$, $c = (1 - \alpha T + \alpha^2 T^2/2 - e^{-\alpha T})/\alpha^2$.

4.3. Observation Models

As observation models, linear model given in [3] and nonlinear RSSI model given in [1] and [7] are considered. In general, in linear model, observation vector contains x and y coordinates and in RSSI model observation vector contains three largest RSSI values related with three different neighboring base stations. An RSSI value is an index of distance between the mobile station and base station. Observation models are presented in sections 4.3.1 and 4.3.2 below.

4.3.1. Linear Observation Model

According to linear model, tracking sensor is assumed to measure target position along the dimensions being analyzed. For cellular tracking system, x and y coordinates are assumed to be measured and following observation model equation is provided:

$$\boldsymbol{o}_k = H\boldsymbol{s}_k + \boldsymbol{\psi}_k \tag{4.25}$$

where

$$H = [1\ 0\ 0\ 0\ 0\ 0; 0\ 0\ 0\ 1\ 0\ 0] \tag{4.26}$$

and $\boldsymbol{\psi}_k$is additive white noise with covariance matrix $R = \sigma_\psi^2 I_2$.

4.3.2. Nonlinear Observation Model

In a cellular network, the distance between the mobile unit and a reachable base station can be inferred from the RSSI of the pilot signal of the base station. The RSSI, measured in dB, is typically modeled as the sum of three terms: path loss, shadow fading, and fast fading. Fast fading is assumed to be sufficiently attenuated using a low pass filter. Therefore, the RSSI received at the mobile unit from the base station in cell i with coordinates (a_i, b_i) at time k is given by

$$\vartheta_{k,i} = \kappa_i - 10\gamma \log(d_{k,i}) + \psi_{k,i} \tag{4.27}$$

where κ_i is a constant determined by the transmitted power, wavelength, antenna height, and gain of cell i , γ is a slop index (typically $\gamma = 2$ for highways and $\gamma = 4$ for microcells in a city) , $\psi_{k,i}$ is a zero mean , Gaussian process with standard deviation σ_ψ typically from 4-8 dB, and $d_{k,i}$ is the distance between the mobile unit and the base station, given by

$$d_{k,i} = \sqrt{(x_k - a_i)^2 + (y_k - b_i)^2} \tag{4.28}$$

To locate the mobile station in the two-dimensional plane, three distance observations to neighboring base stations are sufficient. Thus, the observation vector consists of the three largest RSSIs denoted $\vartheta_{k,1}$, $\vartheta_{k,2}$, $\vartheta_{k,3}$, given as follows:

$$\boldsymbol{o}_k = \left(\vartheta_{k,1}, \vartheta_{k,2}, \vartheta_{k,3}\right)^T = h(\boldsymbol{s}_k) + \boldsymbol{\psi}_k \tag{4.29}$$

where $\boldsymbol{\psi}_k = \left(\psi_{k,1}, \psi_{k,2}, \psi_{k,3}\right)^T$ and

$$h(\boldsymbol{s}_k) = \boldsymbol{\kappa} - 10\gamma \log(\boldsymbol{d}_k) \tag{4.30}$$

where $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \kappa_3)^T$ and $\boldsymbol{d}_k = \left(d_{k,1}, d_{k,2}, d_{k,3}\right)^T$. The covariance matrix of $\boldsymbol{\psi}_k$ is given by $R = \sigma_\psi^2 I_3$.

5. MOBILITY TRACKING ALGORITHMS FOR CELLULAR TRACKING SYSTEM

5.1. Introduction

In this section mobility tracking algorithms based on the mobility models discussed in Section 4 are presented. Basically, there are five algorithms, two using the linear observation model given in (4.25) and three using the nonlinear RSSI observation model given in (4.29). All algorithms use the same state model given in (4.20). In all algorithms, semi-Markovian command process input $\boldsymbol{u}_k$ is assumed to be unknown. Two algorithms treat command process as additional state noise and others treat it as the mode-variable of a multiple model system with modes determined only by the discrete states(commands) of the command process.

The Table 5.1 lists the algorithms according to the mesurement model they use, the way they treat command process and the type of the Kalman filter they use. Algorithms are named according to the type of the observation model and the way command process treated. According to this convention, names of the algorithms are as follows respectively: linear observation command noise(LOCN) algorithm, linear observation command mode(LOCM) algorithm, nonlinear observation command noise(NOCN) algorithm [1] , nonlinear observation command mode(NOCM) algorithm and prefiltered nonlinear observation command mode(PNOCM) algorithm [1]. The term “prefiltered” in the name of the last algorithm refers to the prefilter used in the algorithm.

In general, LOCN algorithm uses a linear Kalman filter. LOCM algorithm uses a Bayesian estimator to estimate command process and a linear Kalman filter to estimate mobility states. NOCN algorithm uses an EKF and an UKF. NOCM algorithm uses parallelly operating EKFs and takes the average of outputs of parallel filters. PNOCM algorithm uses a prefilter to filter out RSSI observations, a modified Kalman filter to estimate command process and an EKF to estimate mobility states. These algorithms are presented in detail in the following sections.

Table 5.1. Mobility tracking algorithms

Name of the Algorithm	Type of the Observation Model	Way Commad Process treated	Type of the Kalman Filter
LOCN Algorithm	Linear	State Noise	Linear
LOCM Algorithm	Linear	Mode Variable	Linear
NOCN Algorithm	Nonlinear	State Noise	Extended and Unscented
NOCM Algorithm	Nonlinear	Mode Variable	Extended
PNOCM Algorithm	Nonlinear	Mode Variable	Extended

5.2. Linear Observation Command Noise Algorithm

In the LOCN algorithm, command process $\boldsymbol{u}_k$ is treated as additional state noise and a linear Kalman filter based on the state model given in (4.20) and linear observation model given in (4.25) is applied. New state noise becomes $B\boldsymbol{u}_k + \boldsymbol{w}_k$ and covariance of the new state noise which is denoted by $\tilde{Q}$ is calculated as follows as defined in [1].

$$\tilde{Q} = Q + BE[(\boldsymbol{u}_k - E[\boldsymbol{u}_k])(\boldsymbol{u}_k - E[\boldsymbol{u}_k])^T]B^T \tag{5.1}$$

where the matrix Q is the covariance matrix of $\boldsymbol{w}_k$, Gaussian state noise. The discrete command process $\boldsymbol{u}_k$ consists of two zero-mean independent semi-Markov processes, so the covariance matrix of $\boldsymbol{u}_k$ is given below as defined in [1].

$$E[(\boldsymbol{u}_k - E[\boldsymbol{u}_k])(\boldsymbol{u}_k - E[\boldsymbol{u}_k])^T] = \sigma_u^2 I_2 \tag{5.2}$$

where σ_u^2 is the variance of u_x and u_y. Although the new state noise process, $B\boldsymbol{u}_k + \boldsymbol{w}_k$, is not white, the correlation between $\boldsymbol{u}_k$ and $\boldsymbol{w}_k$ is ignored. LOCN algorithm is depicted by the Figure 5.1.

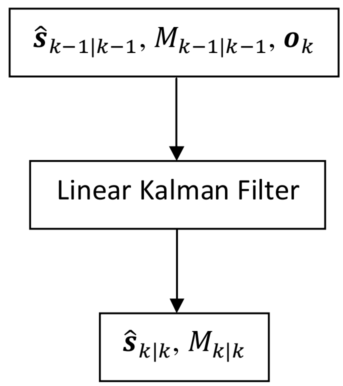

Figure 5.1. LOCN algorithm

The linear Kalman filter used in the LOCN algorithm is described by the Figure 5.2 below. Note that in the calculation of predicted covariance new state noise covariance $\tilde{Q}$ is used. Linear observation model given in (4.25) and state model given in (4.20) is used in the filter steps.

5.3. Linear Observation Command Mode Algorithm

In the LOCM algorithm, command process $\boldsymbol{u}_k$ is regarded as the mode variable of a multiple model system and a GPB1 adaptive multiple model state estimator based on the state model given in (4.20) and linear observation model given in (4.25) is applied. Since the only component that changes among the models is the realization of command process $\boldsymbol{u}_k$, $\boldsymbol{u}_k$ is regarded as the mode variable. Hence, each of the discrete states(commands) of $\boldsymbol{u}_k$ corresponds to a mode(model) and mode probability corresponds to command probability. Number of commands r determine number of modes and hence the number of mode-conditioned filters. As discussed in the modified Kalman filter part of [1], since all filter components other than command value are the same for all mode-conditioned filters of GPB1, GPB1 estimator of the LOCM algorithm can be reduced to a single linear Kalman filter augmented with a Bayesian estimator for command process as depicted in the Figure 5.3.

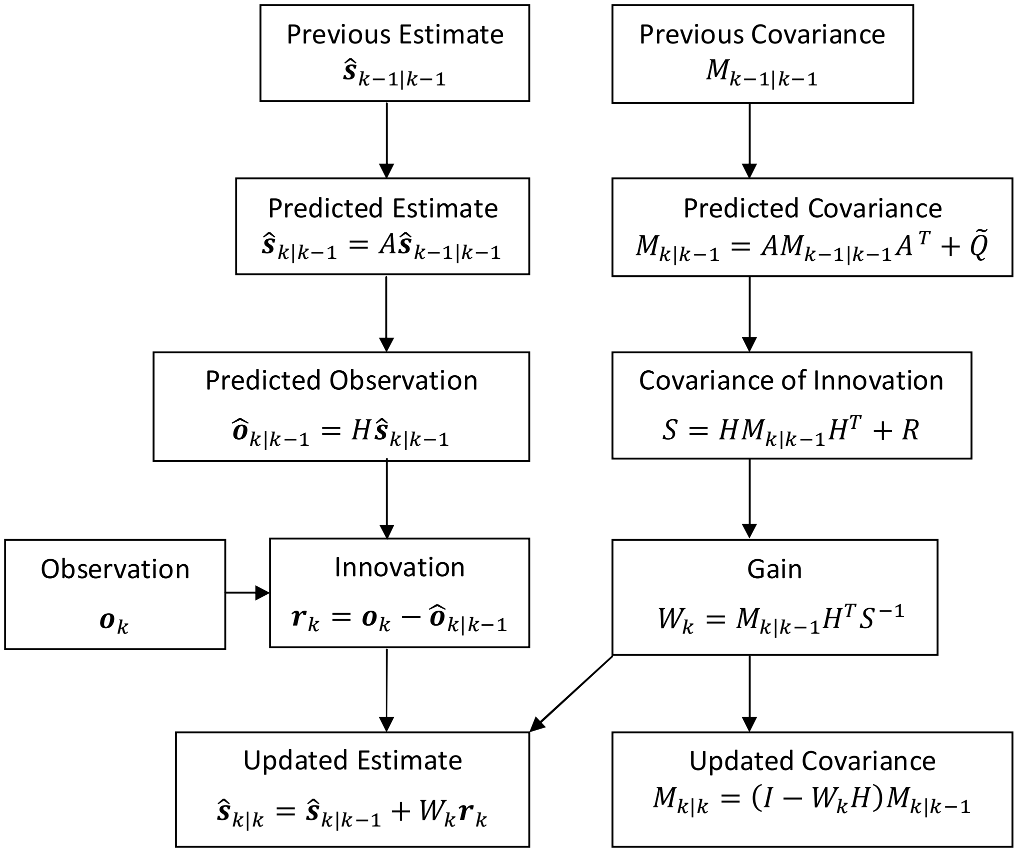

Figure 5.2. Linear Kalman filter of LOCN algorithm

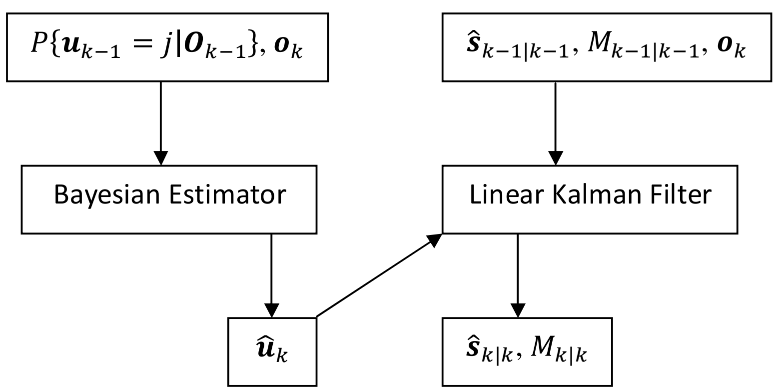

Figure 5.3. LOCM algorithm

As proposed [1], the Bayesian estimator generates command estimates from the previous command probabilites in three steps as depicted by the Figure 5.4. The normaliation constant c in step 2 is chosen such that $\sum_{l\epsilon D} P\{\boldsymbol{u}_k = l|\boldsymbol{O}_k\} = 1$. p_{jl} is homogenous command(mode) transition probability from command j to command l and approximated by a value p near unity for $j = l$ and by $(1-p)/(r-1)$ for $j \neq l$ as defined in [7]. For the likelihood function $f(\boldsymbol{o}_k|\boldsymbol{u}_k = l, \boldsymbol{O}_{k-1})$ in step 2, two Gaussian propositions given below which are defined in [1] and [7] respectively are used:

$$f(\boldsymbol{o}_k|\boldsymbol{u}_k = l, \boldsymbol{O}_{k-1}) = \mathcal{N}\left(H\left(A\hat{\boldsymbol{s}}_{k-1|k-1} + Bl\right), HM_{k|k-1}H^T\right) \tag{5.3}$$

$$f(\boldsymbol{o}_k|\boldsymbol{u}_k = l, \boldsymbol{O}_{k-1}) = \mathcal{N}\left(H\left(A\hat{\boldsymbol{s}}_{k-1|k-1} + Bl\right), H\left[AM_{k-1|k-1}A^T + Q\right]H^T + R\right) \tag{5.4}$$

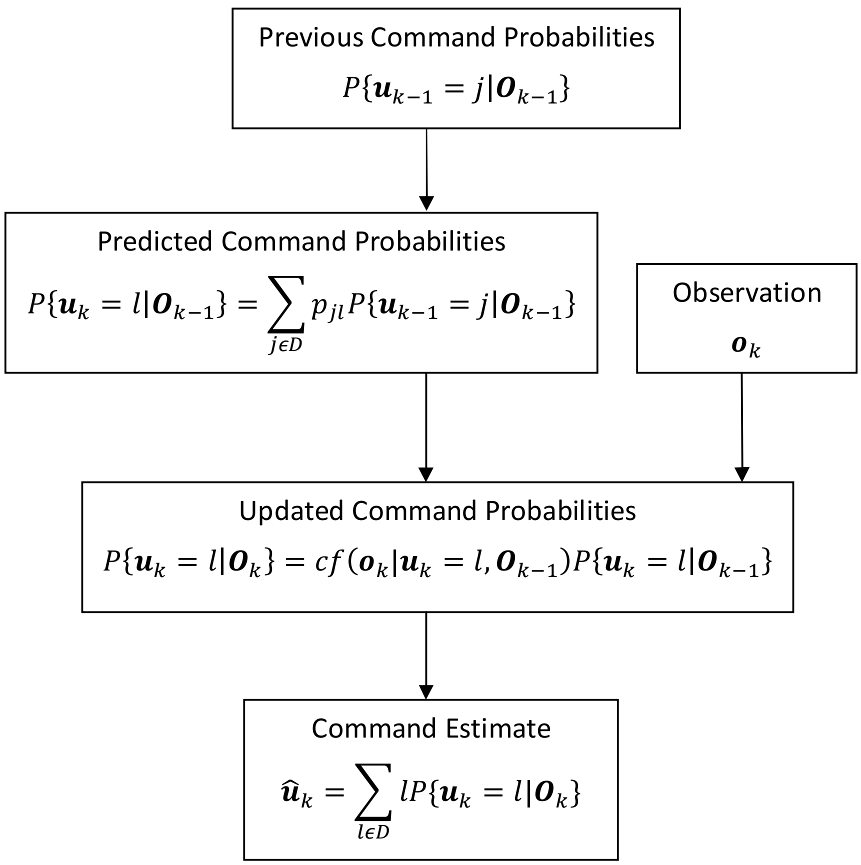

Figure 5.4. Bayesian estimator of LOCM algorithm

The linear Kalman filter of the LOCM algorithm given by the Figure 5.5 differs from the one of the LOCN algorithm by using command estimate $\hat{\boldsymbol{u}}_k$ in the predicted estimate calculation and using the state noise covariance matrix Q instead of $\tilde{Q}$ of LOCN algorithm in the predicted covariance calculation.

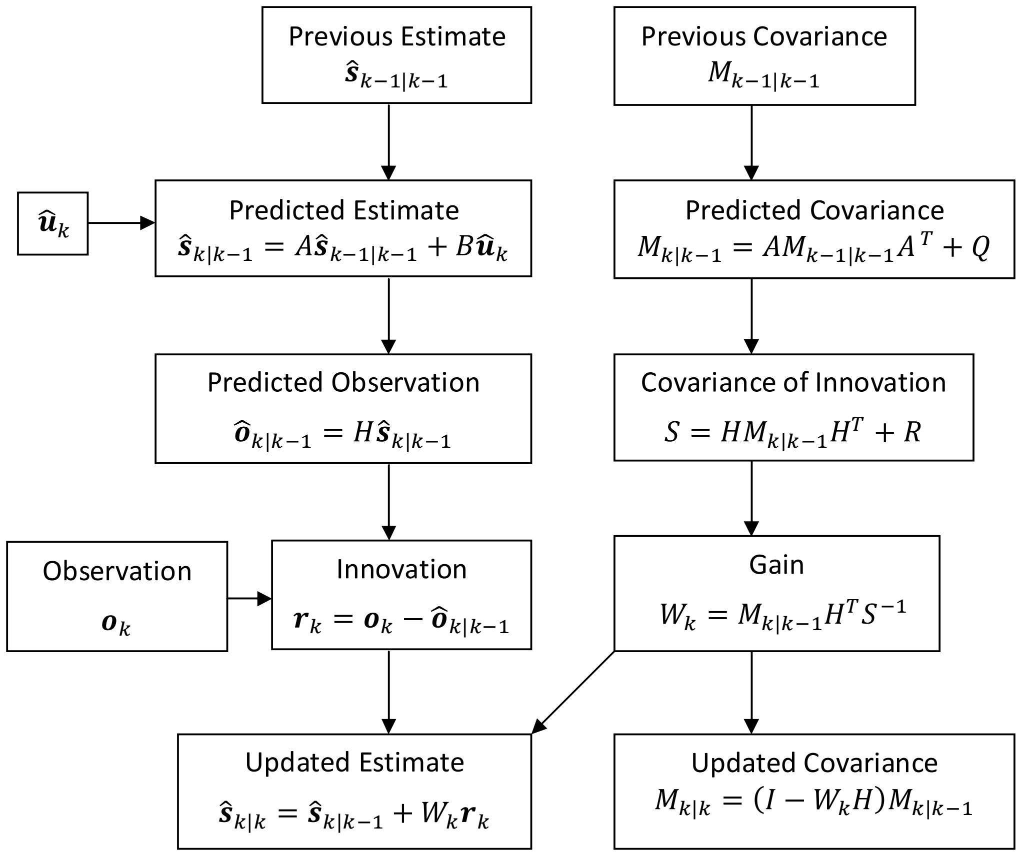

Figure 5.5. Linear Kalman filter of LOCM algorithm

5.4. Nonlinear Observation Command Noise Algorithm

Like LOCN algorithm, NOCN algorithm [1] treats command process $\boldsymbol{u}_k$ as additional state noise and covariance matrix of state noise is updated using (5.1) and (5.2). In this algorithm nonlinear RSSI model given in (4.29) is used as the observation model. To cope with the nonlinearity, either EKF or UKF is used. EKF version of NOCN algorithm is proposed in [1] and UKF version is introduced in this work. Figure 5.6 depicts the NOCN algorithm.

For UKF, samples are calculated and used in the Kalman filter framework exactly as described in Section 3.5. To use EKF, nonlinear RSSI observation model is linearized as follows as given in [1]:

$$\boldsymbol{o}_k = h(\boldsymbol{s}_k^*) + H_k \Delta \boldsymbol{s}_k + \boldsymbol{\psi}_k \tag{5.5}$$

where $\boldsymbol{s}_k^*$ is the nominal or reference vector and $\Delta\boldsymbol{s}_k = \boldsymbol{s}_k - \boldsymbol{s}_k^*$ is the difference between the true and nominal state vectors. As mentioned in Section 3.4, the nominal vector is obtained from the predicted state trajectory $\hat{\boldsymbol{s}}_{k|k-1}$, i.e., $\boldsymbol{s}_k^* = \hat{\boldsymbol{s}}_{k|k-1}$. Hence, the linearized observation matrix H_k is given by

$$H_k = \frac{\partial h}{\partial s}|_{s=\hat{s}_{k|k-1}} = -5\gamma \begin{bmatrix} h_{k,1} \\ h_{k,2} \\ h_{k,3} \end{bmatrix} \tag{5.6}$$

where $h_{k,i} = \frac{2}{d_{k,i}^2}[x_k - a_i \; 0 \; 0 \; y_k - b_i \; 0 \; 0]$ for $i = 1,2,3$.

Using $\tilde{Q}$ given in (5.1), H_k given in (5.6), state model given in (4.20) and nonlinear observation model given in (4.29), EFK of NOCN algorithm is described by the Figure 5.7.

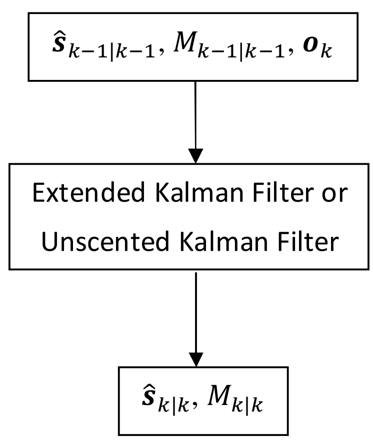

Figure 5.6. NOCN algorithm

5.5. Nonlinear Observation Command Mode Algorithm

Like LOCM algorithm, NOCM algorithm regards command process $\boldsymbol{u}_k$ as the mode variable of a multiple model system. In this algorithm, nonlinear RSSI model given in (4.29) is used as the observation model. Hence, a GPB1 adaptive multiple model state estimator having r mode-conditioned EKFs are applied. Note that linearized observation matrix, covariance of innovation and gain of the filters are different among the filters because they depend on the command process through the predicted estimate which directly depends on the command process as shown in the Figure 5.9. Hence, filters have more than one different component, thus they can not be reduced to a single Kalman filter augmented with a Bayesian estimator for command process like in LOCM algorithm. In [7], aforomentioned filter components are assumed to be the same for all the filters and a reduced form of GPB1 estimator that has an EKF with a Bayesian estimator for command process is proposed. However, such a reduction is valid only if the only filter component different among the filters is the command process value which is not the case for this nonlinear observation model. Hence, NOCM algorithm proposed here, as depicted in the Figure 5.8, applies a GPB1 filter having r mode-conditioned EKFs with the modes corresponding to the commands of the command process. Note that mode probability q_k^l corresponds to the command probability $P\{\boldsymbol{u}_k = l|\boldsymbol{O}_k\}$. Note also that filters also generate likelihood function values λ_k^l that are later used with the previous mode probabilities q_{k-1}^j to generate updated mode probabilities q_k^l.The mode-conditioned updated estimates and covariances generated by the filters are then averaged using these updated mode probabilities to calculate updated state estimate and covariance .

The mode-conditioned EKF of NOCM algorithm is described in detail by the Figure 5.9. The nonlinear function is linearized exactly in the same way as in the NOCN algorithm. Likelihood function $\lambda_k^l = f(\boldsymbol{o}_k|\boldsymbol{u}_k = l, \boldsymbol{O}_{k-1})$ is calculated using Gaussian proposition given in (5.4). Besides, normalization constant c and mode transition probability p_{jl} that are used in the calculation of updated mode probabilities are calculated as in the LOCM algorithm.

5.6. Prefiltered Nonlinear Observation Command Mode Algorithm

PNOCM algorithm is developed in [1]. As NOCM algorithm, it regards the command process as the mode variable of a mulitple model system and uses nonlinear RSSI measurement model given in (4.29). Basically, as depicted in the Figure 5.10, it uses a prefilter to reduce shadowing noise in RSSI observations and then calculate coarse positions from filtered observations. Coarse positions are then given as input to a GPB1 estimator(called modified Kalman filter in [1]) which is dedicated to estimate command process. Finally command estimates and filtered RSSI observations are used in an EKF to calculate state estimates and covariances. More detailed description of these modules of PNOCM algorithm are given below.

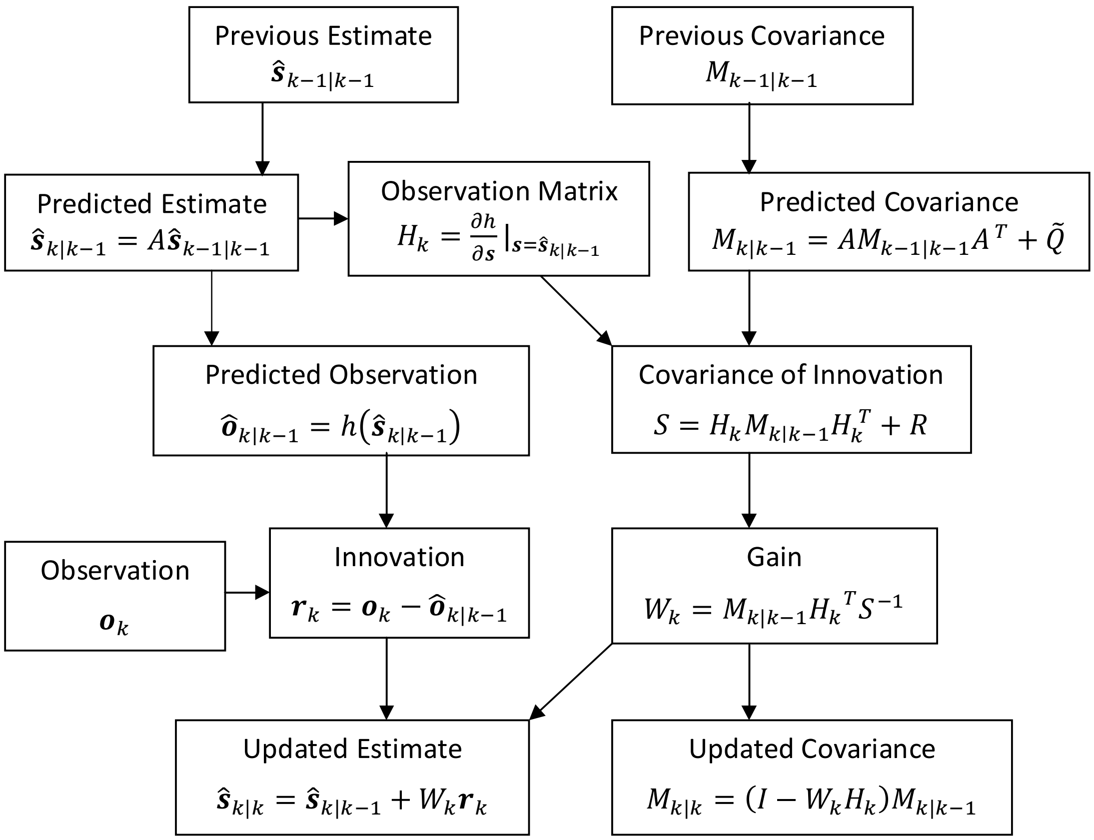

Figure 5.7. Extended Kalman filter of NOCN algorithm

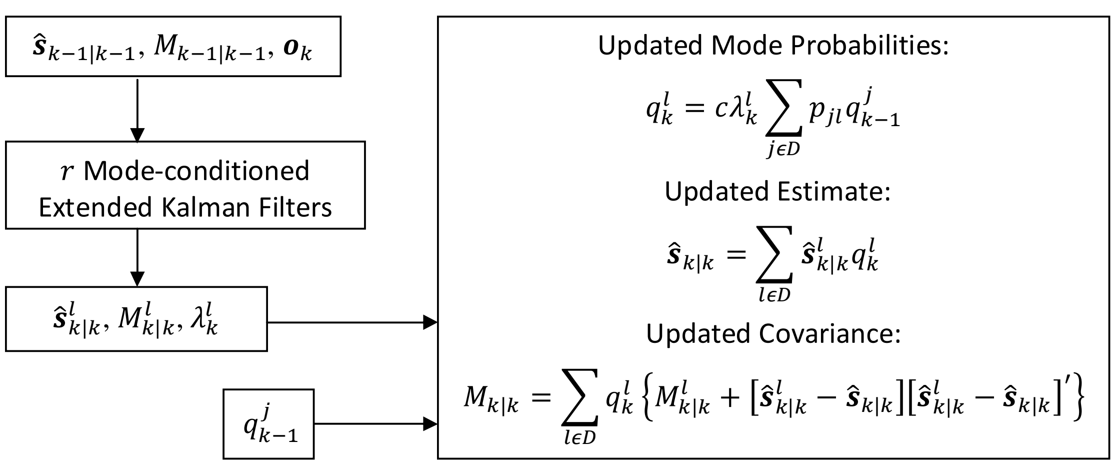

Figure 5.8. NOCM algorithm

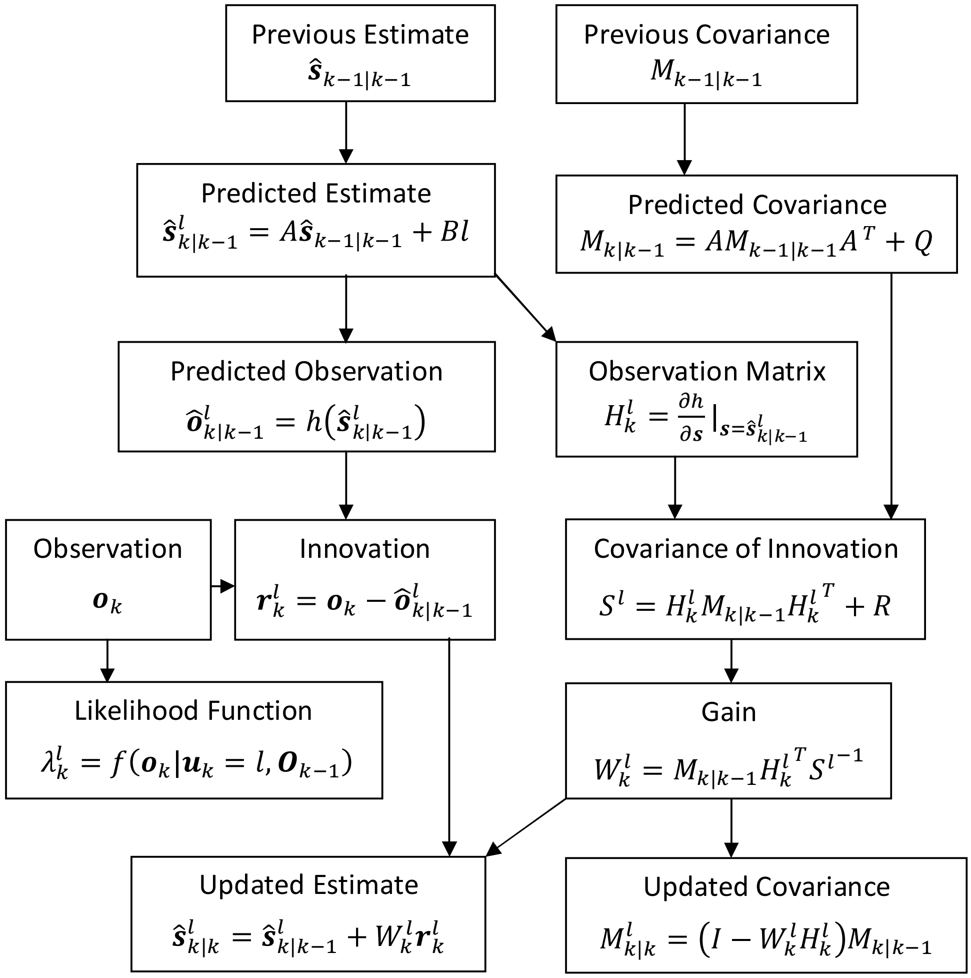

Figure 5.9. Mode-conditioned extended Kalman filter of NOCM algorithm conditioned on mode l

5.6.1. Prefilter

As depicted in the Figure 5.11, the prefilter consists of an averaging filter and a coarse position estimator and outputs averaged RSSI observation $\tilde{\boldsymbol{o}}_k$ and a vector of position estimates denoted by $\hat{\boldsymbol{o}}_k = [\hat{x}_k, \hat{y}_k]^T$, which are used as the observation data for the EKF and modified Kalman filter, respectively. The averaging filter reduces the shadowing noise considerably, without significantly modifying the path loss. The averaged RSSI observations are then used to generate coarse position estimates of the position coordinates.

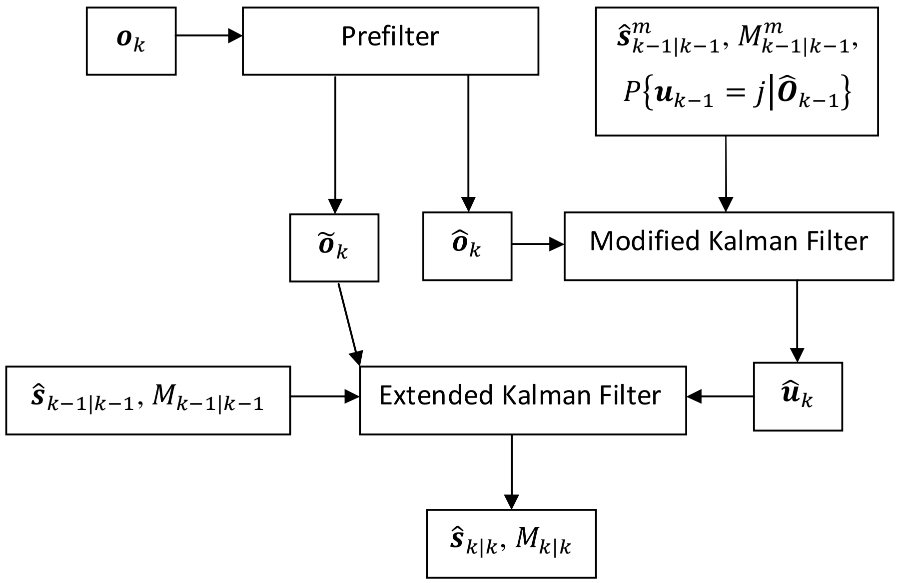

Figure 5.10. PNOCM algorithm

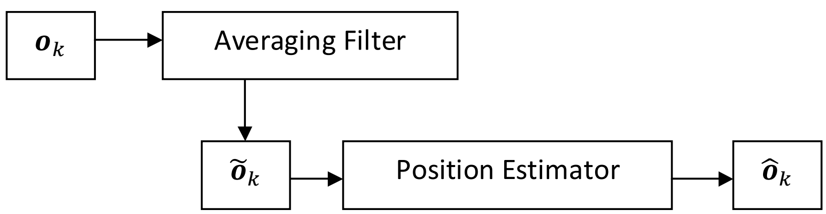

Figure 5.11. Prefilter of PNOCM algorithm

The observation vector $\boldsymbol{o}_k$ consists of the path loss and the shadowing component. The averaging filter reduces the shadowing component in the observations. Different filters can be used for this purpose. Applying a rectangular window, the output $\widetilde{\boldsymbol{o}}_k$ of the averaging filter is given as

$$\widetilde{\boldsymbol{o}}_k = \frac{1}{L}\sum_{i=k-L+1}^{k} \boldsymbol{o}_i \tag{5.7}$$

where L is the length of the window. For small L, the residual shadowing component is quite large and yields erroneous position estimates; however, for large L, the path loss is modified and induces errors in the position estimates. The suggested solution to this problem is to use a bank of averaging filters in series, each with small length L, instead of a single filter of larger length. In the filter bank arrangement, each filtger performs an averaging operation according to (5.7) and provides its output as input to the next filter in series. This averaging scheme preserves path loss and reduces shadowing noise to a satisfactory level. The averaged observations $\tilde{\boldsymbol{o}}_k$ are used by the position estimator to generate coarse position coordinates $\hat{\boldsymbol{o}}_k = [\hat{x}_k, \hat{y}_k]^T$, which are obtained as follows:

$$\begin{bmatrix} a_1 - a_2 & b_1 - b_2 \\ a_1 - a_3 & b_1 - b_2 \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix} = 0.5 \begin{bmatrix} -e^{\frac{\kappa_1 - \tilde{o}_k(1)}{5\gamma}} + e^{\frac{\kappa_2 - \tilde{o}_k(2)}{5\gamma}} + a_1^2 - a_2^2 + b_1^2 - b_2^2 \\ -e^{\frac{\kappa_1 - \tilde{o}_k(1)}{5\gamma}} + e^{\frac{\kappa_2 - \tilde{o}_k(3)}{5\gamma}} + a_1^2 - a_3^2 + b_1^2 - b_3^2 \end{bmatrix} \tag{5.8}$$

where (a_i, b_i) , $i = 1,2,3$ are the base station coordinates for cell i .

5.6.2. Modified Kalman Filter

As depicted in the Figure 5.12, the modified Kalman filter has exactly the same structure as the reduced GPB1 estimator of LOCM algorithm. It has a Bayesian estimator for command process which is integrated to a linear Kalman filter. Only the command estimate generated by the Bayesian estimator is used by the EKF module of the PNOCM algorithm. State estimate and covariance generated by the linear Kalman filter are used by the modified Kalman filter internally to calculate the next command estimate. It is important to note that the input to the linear Kalman filter is the two dimensional prefiltered observation $\hat{\boldsymbol{o}}_k$, rather than the three dimensional raw observation $\boldsymbol{o}_k$ and the observation matrix is the linear observation matrix given in (4.25) . Besides, the likelihood function defined in (5.3) which is assumed to be closer to the Gaussianity than the function defined (5.4) is used in the Bayesian estimator.

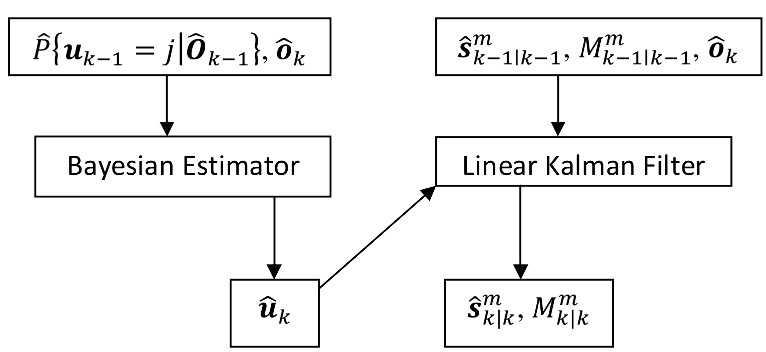

Figure 5.12. Modified Kalman filter of PNOCM algorithm

5.6.3. Extended Kalman Filter

The modified Kalman filter described above provides the mobility state estimates and discrete command estimates. However, the accuracy of the mobility state estimates is largely dependent on the performance of the prefilter. Since the coarse position estimates are used as the observations for the modified Kalman filter, the best the filter can do is to track the coarse position coordinates. As discussed in [1], any inaccuracy and error in the prefilter can cause the estimator to diverge. To avoid this problem, the command estimate of the modified Kalman filter is used by an EKF, depicted in the Figure 5.13, to produce the mobility state estimate and covariance. Note that the EKF takes the averaged RSSI $\tilde{\boldsymbol{o}}_k$ as observation. Besides, the matrix R_{res} is used in the calculation of the covariance of the innovation as the covariance of the residual noise in the averaged RSSI. A suitable matrix for R_{res} is ϵI_3 , where $\epsilon \approx 0.01$.

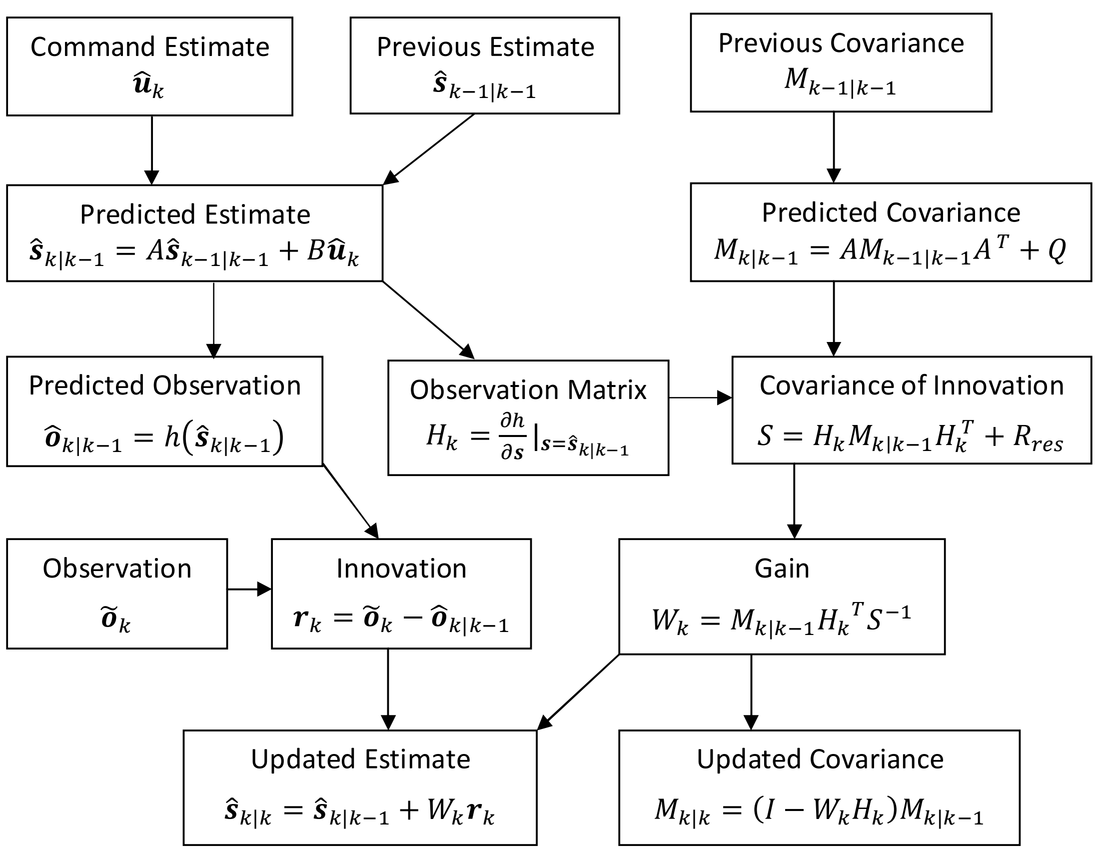

Figure 5.13. Extended Kalman filter of PNOCM algorithm

6. SIMULATION RESULTS

In this section, plots and RMSE statistics are presented for tracking algorithms presented in Section 5. Algorithms are implemented and simulated in Matlab. For each simulation, an independent Monte Carlo trajectory is generated using the state model given in (4.20) and estimated using any of the tracking algorithms discussed in Section 5. Estimation is performed in real-time for each sampling instant of a trajectory. In other words, at each sampling instant, current state is generated from the previous state using the state model, an observation is realized from the current state using the observation model and the state is estimated by any algorithm discussed in Section 5 using the observation, previous state estimate, previous state covariance and other parameters depending on the algorithm. The sample size of each trajectory is assumed to be 600 and sampling interval T is assumed to be 0.1 seconds as suggested in [1]. Initial values are determined as suggested in [9]: initial estimate and covariance are assumed to be zero vector and identity matrix respectively and initial state is chosen from a Gaussian distribution with mean equal to initial estimate and covariance equal to initial covariance.

For the discrete command processes $u_x(t)$ and $u_y(t)$ two sets of acceleration levels(commands) each taking on five possible levels of acceleration in units of m/s^2 as defined in [1] are used in simulations. The set $L = \{-5, -2.5, 0, 2.5, 5\}$ is capable of generating a wide range of dynamic motion and referred to the high mobility scenario. The other set $L = \{-0.5, -0.25, 0, 0.25, 0.5\}$ is referred to the low mobility scenario and more suitable for urban areas where smaller cells are used. As suggested in [1], the initial probability vector for commands is initialized to the uniform distribution and the dwell times in each state are uniformly distributed with a comman mean value of 20 sample points. As suggested in [7], transition probability between the same command states is assumed to be 0.95 and between different states 0.05/24 = 0.0021. Initial command value and its estimate are assumed to be zero vectors.

Other parameters are set for a typical cellular network as defined in [1]: The parameters determining the autocorrelation function of the Gaussian acceleration process $\boldsymbol{r}(t)$ are set $\alpha = 1000s^{-1}$ and $\sigma_1 = 1dB$, the covariance matrix R of ψ_k is determined by setting the parameter $\sigma_{\psi} = 6dB$ and the parameter κ_i is assumed to be zero for all cells i. Window size of an averaging filter and the number of filters used in series are assumed to be 5 and 8 respectively to obtain a sufficient noise filtering without losing too much path loss information.

Plots and RMSE values of LOCN, LOCM and NOCN algorithms are given below. The position, velocity and acceleration coordinates are specified in units of meters, meter/second, meter/$second^2$, respectively. Table 6.1 shows sample mean and standard deviation of 500 RMSE values computed using 500 independent Monte Carlo simulations. An RMSE for a simulation is calculated as:

$$\sqrt{\frac{1}{N}\sum_{n=1}^{N}[(\hat{x}_n - x_n)^2 + (\hat{y}_n - y_n)^2]} \tag{6.1}$$

where N is sample size of the trajectory(600 for all simulations), x_n and y_n are the true value of the state component(position, velocity or acceleration) at instant n in x and y are coordinates, respectively and $\hat{x}_n$ and $\hat{y}_n$ are their estimates in x and y are coordinates, respectively. NOCM and PNOCM algorithms could not be executed because of the problems mentioned in Section 7. Plots and RMSE values are discussed in Section 7.

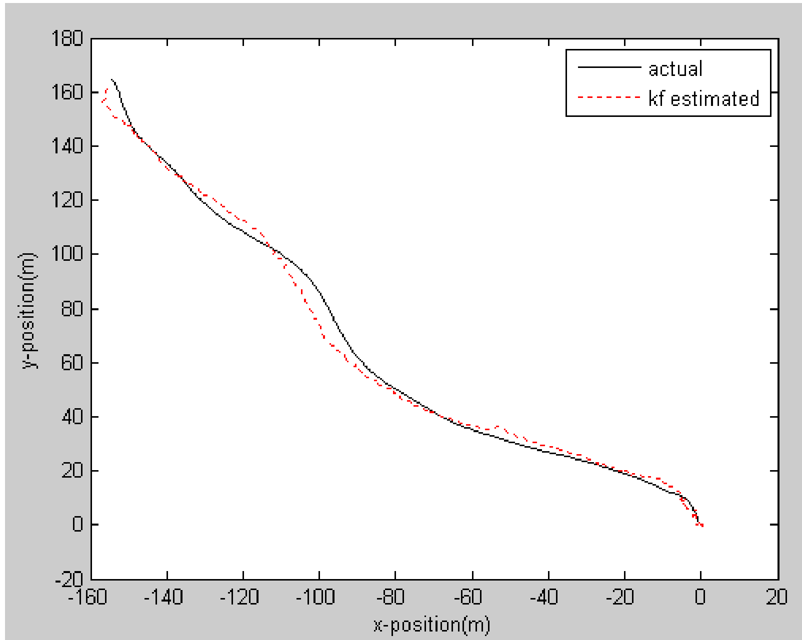

Figure 6.1. Actual positions(solid line) and LOCN-estimated positions(dotted line) of a sample trajectory for low mobility model

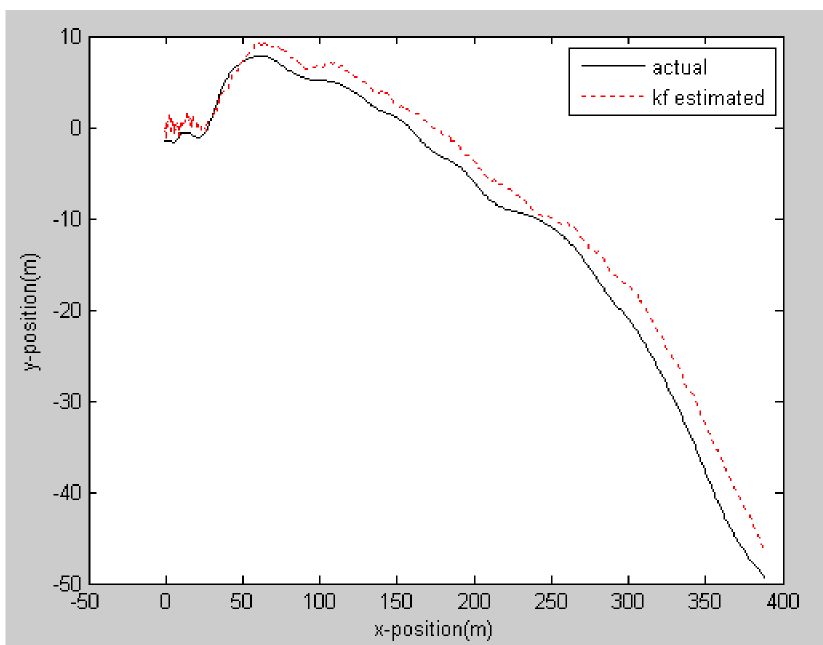

Figure 6.2. Actual positions(solid line) and LOCM-estimated positions(dotted line) of a sample trajectory for low mobility model

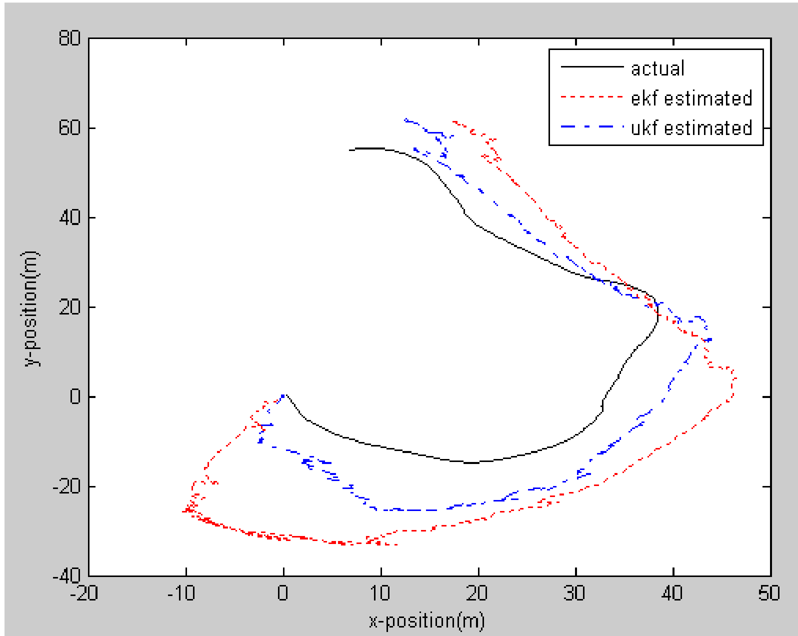

Figure 6.3. Actual positions(solid line) and NOCN-estimated positions(dotted line for EKF estimation, dashed line for UKF estimation) of a sample trajectory for low mobility model

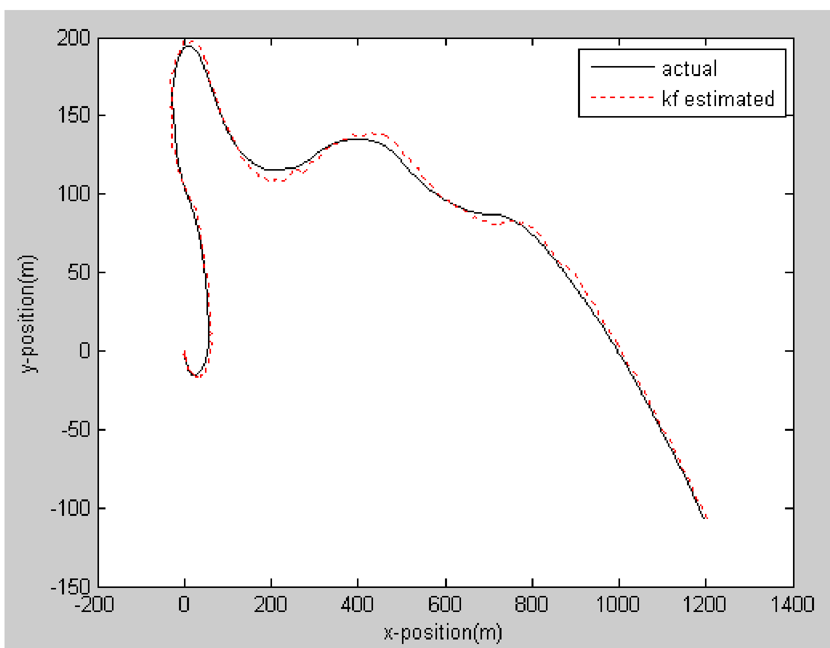

Figure 6.4. Actual positions(solid line) and LOCN-estimated positions(dotted line) of a sample trajectory for high mobility model

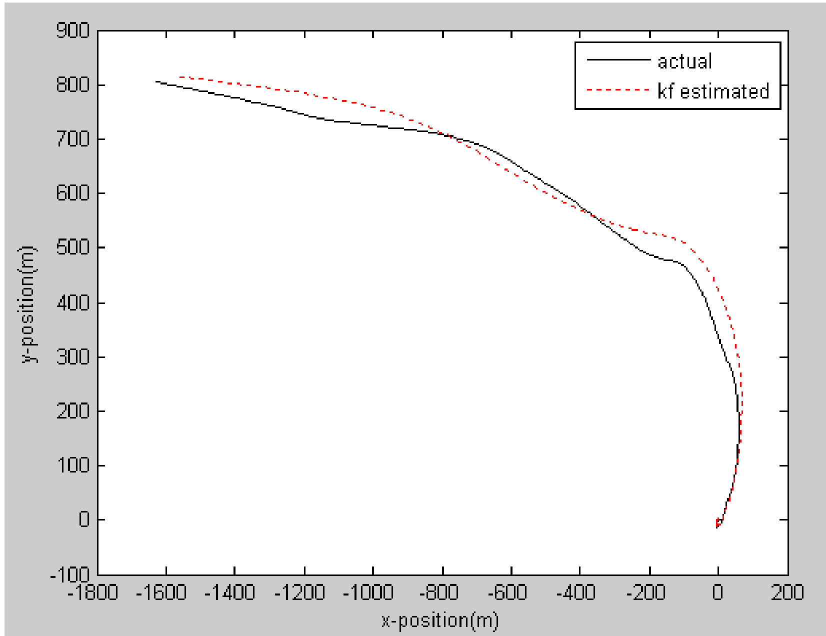

Figure 6.5. Actual positions(solid line) and LOCM-estimated positions(dotted line) of a sample trajectory for high mobility model

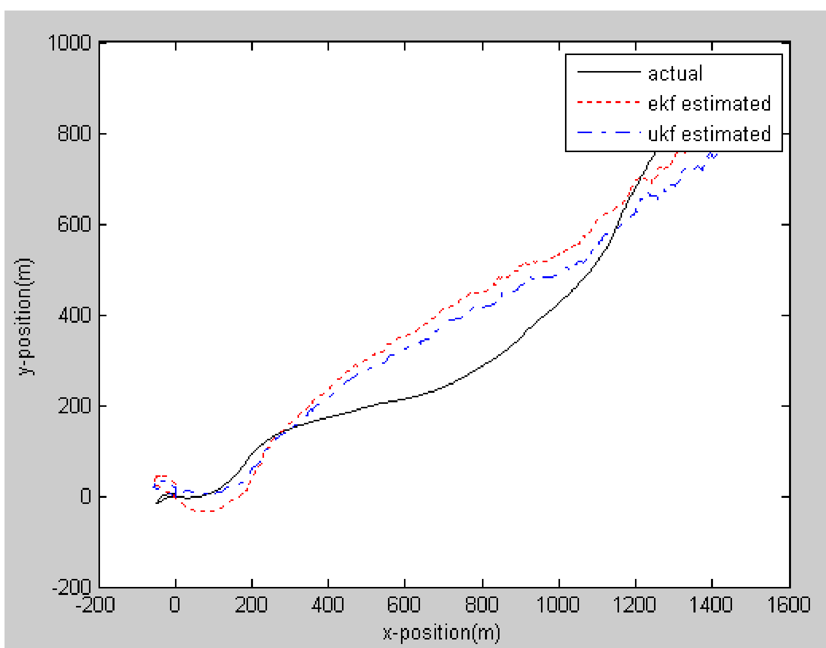

Figure 6.6. Actual positions(solid line) and NOCN-estimated positions(dotted line for EKF estimation, dashed line for UKF estimation) of a sample trajectory for high mobility model

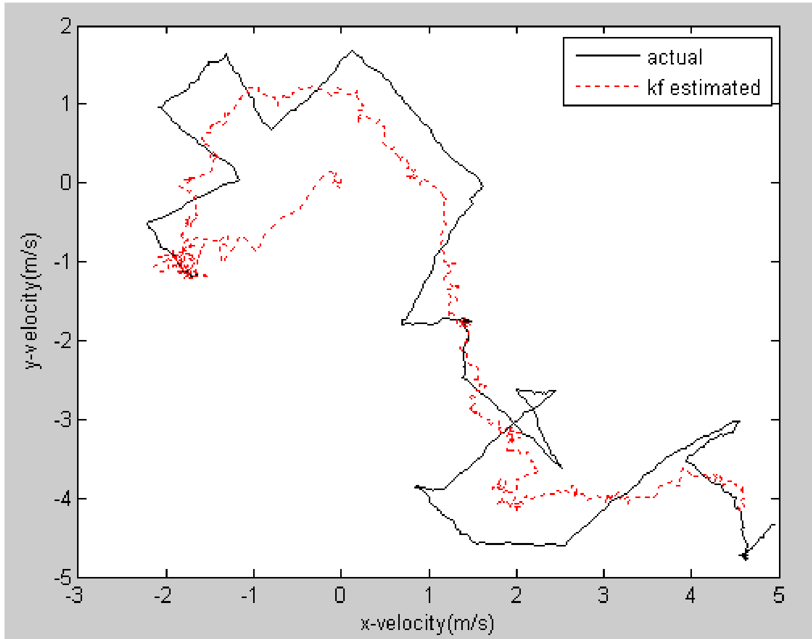

Figure 6.7. Actual velocities(solid line) and LOCN-estimated velocities(dotted line) of a sample trajectory for low mobility model

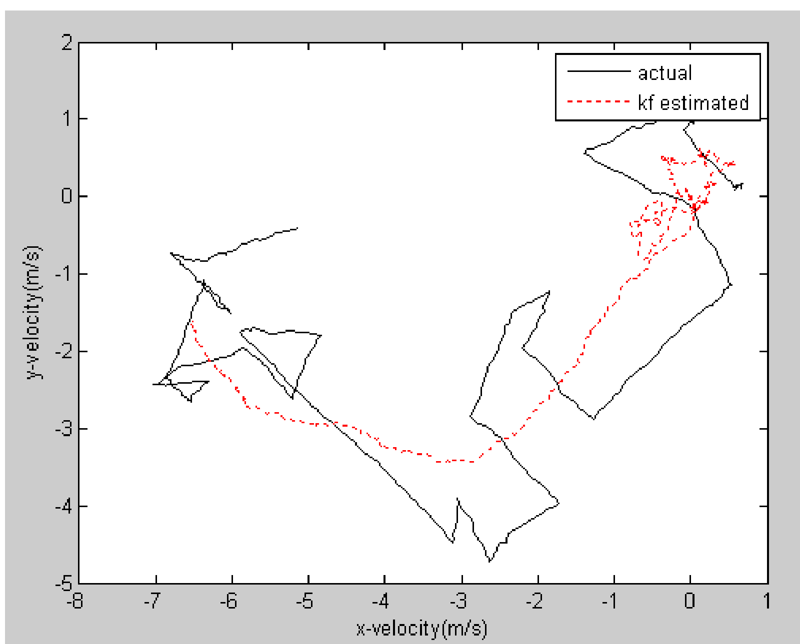

Figure 6.8. Actual velocities(solid line) and LOCM-estimated velocities(dotted line) of a sample trajectory for low mobility model

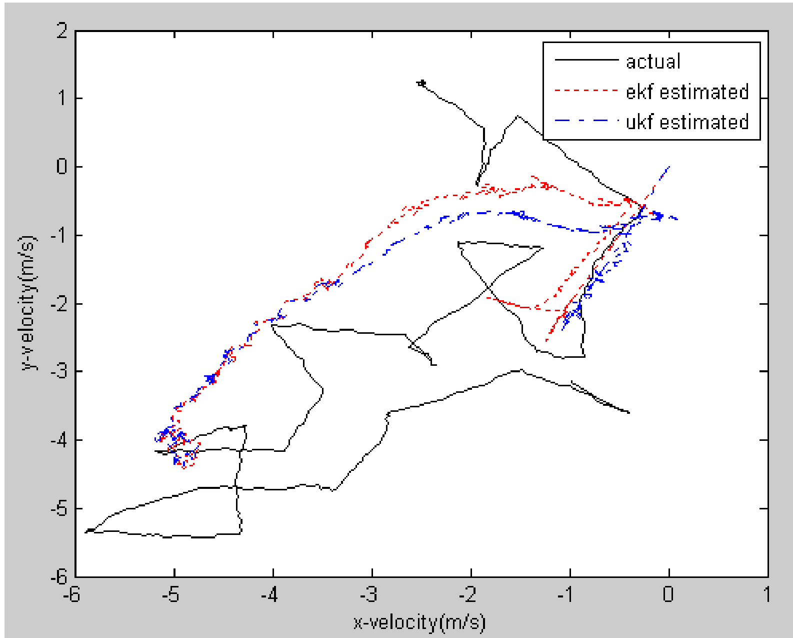

Figure 6.9. Actual velocities(solid line) and NOCN-estimated velocities(dotted line for EKF estimation, dashed line for UKF estimation) of a sample trajectory for low mobility model

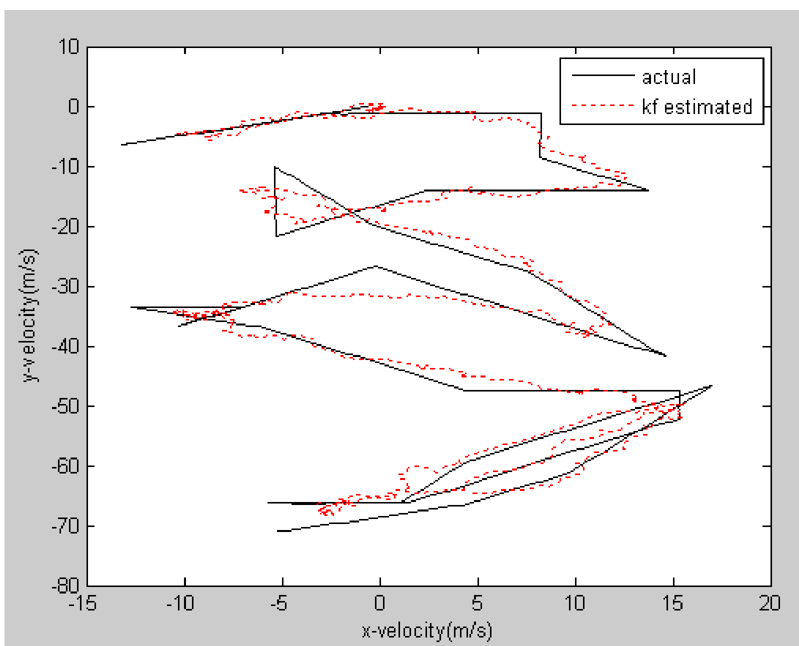

Figure 6.10. Actual velocities(solid line) and LOCN-estimated velocities(dotted line) of a sample trajectory for high mobility model

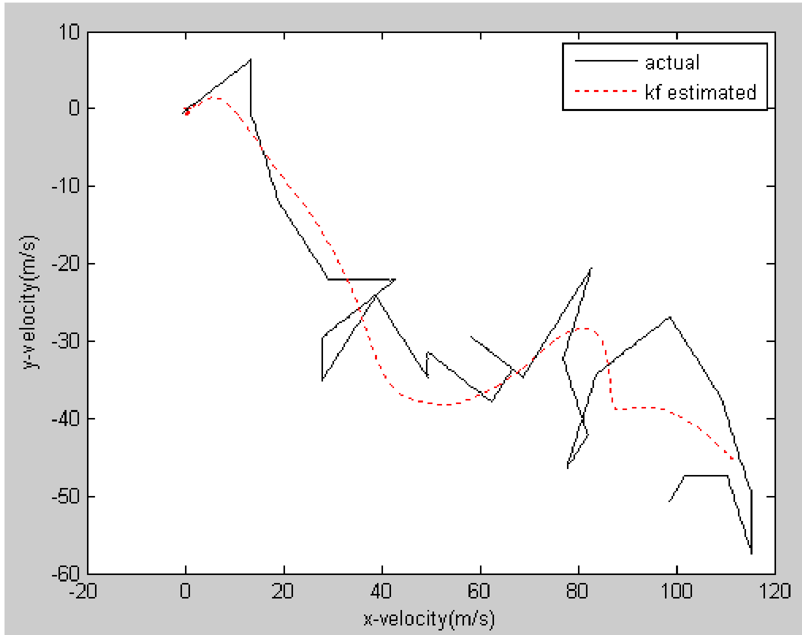

Figure 6.11. Actual velocities(solid line) and LOCM-estimated velocities(dotted line) of a sample trajectory for high mobility model

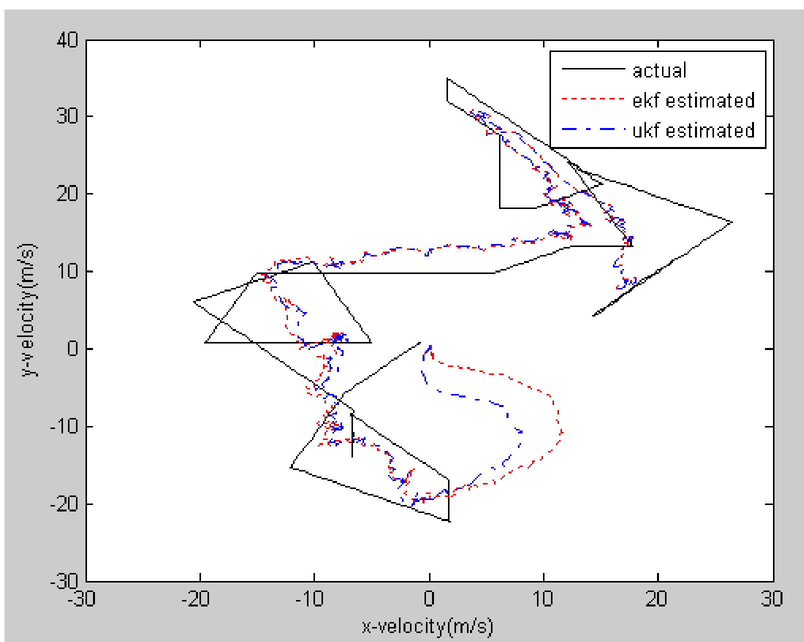

Figure 6.12. Actual velocities(solid line) and NOCN-estimated velocities(dotted line for EKF estimation, dashed line for UKF estimation) of a sample trajectory for high mobility model

Table 6.1. RMSE statisctics for tracking algorithms

Tracking Algorithm	Low Mobility Scenario		High Mobility Scenario	
	mean	st.dev.	mean	st.dev.
LOCN position(m)	2.61	0.25	4.41	0.38
LOCN velocity(m/s)	1.59	0.28	8.41	1.83
LOCN acceleration(m/s^2)	1.49	0.03	5.83	0.69
LOCM position(m)	6.11	1.44	45.19	8.78
LOCM velocity(m/s)	1.74	0.31	14.85	2.34
LOCM acceleration(m/s^2)	1.49	0.03	5.25	0.29
NOCN EKF position(m)	60.31	74.41	492.70	675.33
NOCN UKF position(m)	49.47	73.30	443.77	635.34
NOCN EKF velocity(m/s)	3.65	2.31	31.55	20.99
NOCN UKF velocity(m/s)	3.35	2.21	33.15	23.86
NOCN EKF acc. (m/s^2)	1.50	0.03	5.91	1.50
NOCN UKF acc. (m/s^2)	1.49	0.03	5.63	1.00

7. CONCLUSIONS

In this thesis, after presenting basics of linear MMSE filtering and dynamic state estimation based on Kalman filtering, a special type of tracking system for cellular communication networks called cellular tracking system presented in [7] is discussed in depth. This system is originally proposed in [6] for tracking targets in tactical weapons systems in three dimensional space(range,azimuth and elevation coordinates) and captures a large range of mobility by modeling acceleration(manuever) as driven by a discrete semi-Markovian command process proposed in [4] and a Gaussian time-correlated random process proposed in [3]. Then, in [7], this combined system is customized for two dimensional(x and y coordinates) mobility modeling and tracking in cellular communication networks having a nonlinear RSSI observation model given in (4.29). A linear observation model given in (4.25) that provides x and y position coordinates is introduced in this thesis and included to the cellular tracking system.

Based on this cellular tracking system, five mobility tracking algorithms are presented in Section 5. These algorithms use the same linear state model represented by the equation (4.20). In principal, they differ in the observation model they use and in the way they treat semi-Markovian manuever component and they employ variants of the Kalman filter(linear, extended or unscented) accordingly. For observation model, LOCN and LOCM algorithms use linear model and NOCN, NOCM and PNOCM algorithms use nonlinear RSSI model. LOCN and NOCN algorithms treat Markovian manuever component as an additional state noise and only update covariance matrix of the state noise. Calculation of the covariance of the new state noise occurs according to the equations (5.1) and (5.2) that are proposed in [1]. LOCM, NOCM and PNOCM algorithms treat the tracking system as a multiple model system with the Markovian manuever as the model variable and employ GPB1 adaptive estimator which is presented in Section 3.6.1. In this approach, manuever commands represent models and command transition probabilities correspond to mode transition probabilities. PNOCM algorithm applies an additional prefilter to reduce effect of shadowing noise in the RSSI model considerably wihout significantly modifying the path loss. LOCM and PNOCM algorithms and EKF version of NOCN algorithm are proposed in [1]. LOCN and NOCM algorithms and UKF version of NOCM algorithm are introduced in this thesis.

Algorithms are implemented and simulated in MATLAB and RMSE statistics for LOCN, LOCM and NOCN algorithms are presented in Table 6.1. Note that high mobility error statistics are larger than those of low mobility for all algorithms. This is due to larger actual states and estimates that cause larger absolute estimation errors in high mobility scenario in which target moves faster and further. Error statistics are further discussed below.

According to the statistics, LOCN algorithm which uses a single linear Kalman filter performs quite well as expected. This is due to lack of any approximation that could cause serious filter errors. The only approximation is that although the new state noise, $B\boldsymbol{u}_k + \boldsymbol{w}_k$, is not white, the correlation between the noise components is ignored and new state noise is approximated as a white noise. From RMSE values of LOCN algorithm, it seems that this approximation does not introduce serious errors. In fact, when Morkavian manuever is removed from the state model which makes state noise exactly white, an error mean of 1.3 and error variance of 0.3 are obtained which are very close to the original LOCN statistics in the sense of RMSE.

LOCM algorithm differs from LOCN algorithm by including a Bayesian estimator for command process and using command estimate in the state prediction stage of a linear Kalman filter. In other words, LOCM algorithm does not regard command process as an additional state noise, instead, it applies a simplified multiple model approach. Models are driven by command process because the only thing that differs among the models is the discrete command process values. That is why a simplified GPB1 estimator that consists of linear Kalman filter augmented with a Bayesian command estimator could be used instead of a bank of filters as in an ordinary GPB1 estimator like in NOCM algorithm.

Recall that in a GPB1 estimator, like one used in LOCM, only the possible models in the last sampling period are considered. It is assumed that at time $k-1$ there is a single lumped estimate $\hat{\boldsymbol{s}}_{k-1|k-1}$ and the associated covariance that approximately summarizes the past $\boldsymbol{O}_{k-1}$. In other words, histories of models that differ in "older" models are combined to decrease number of models used in average state calculation.

Another approximation in LOCM algorithm is Gaussian assumption of the likelihood function $f(\boldsymbol{o}_k|\boldsymbol{u}_k = l, \boldsymbol{O}_{k-1})$. Two propositions for this pdf given in (5.3) and (5.4) are applied in simulations. Given RMSE statistics are obtained using second model which is proposed in [7] since first model proposed in [1] causes resulting filter to completely diverge from actual trajectories in simulations. According to RMSE statistics, Gaussian assumption of likelihood function together with the multiple model assumption of GPB1 estimator cause LOCM to introduce larger approximation errors than LOCN, especially in high mobility model. Hence, according to the RMSE statistics, for linear state model given in (4.20) and linear observation model given in (4.25), treating command process as an additional state noise is a better approximation than treating it as a model variable in a GPB1 adaptive estimator.

NOCN algorithm applies a similar approach to the LOCN algorithm by considering the command process as an additional state noise and calculates the covariance matrix of the new state noise in the same way as the LOCN algorithm. However, different from LOCN algorithm, NOCN uses nonlinear RSSI observation model. To handle nonlinear observation model, NOCN uses either EKF or UKF. EKF approximates nonlinear equation with its first order Taylor Series expansion which is linear with respect to current state so that linear Kalman filter equations could be utilized. UKF, instead of functional linearization, uses probabilistic approximation. UKF uses UT to represent previous posterior state pdf by a deterministically chosen sample points and to calculate predicted state and observation mean and covariances using chosen sample points and their weights. Predicted means and covariances are then used in the traditional Kalman filter framework to calculate current state estimate(mean) and covariance. RMSE statistics show that UKF works better than EKF in the framework of NOCN algorithm.

However, according to RMSE statistics, both functional linearization of EKF and probabilistic approximation of UKF do not work well. This is because nonlinear structure of RSSI observation function considerably pronounces non-Gausianity of posterior state pdf and causes EKF and UKF to introduce large approximation errors in prediction steps.

These approximation errors together cause large filter biases, wrong covariances, wrong gains and so large estimation errors, even worser, leaad to complete filter divergence sometimes. In fact, when a different nonlinear observation equation that precludes logarithm and square root in the RSSI function is applied, much smaller RMSE values are obtained.

NOCM algorithm applies adaptive GPB1 multiple model estimator to the nonlinear RSSI observation model by considering the command process as the mode variable like LOCM algorithm. However, different from LOCM, NOCM algorithm does not use a reduced form of GPB1 with a single Kalman filter augmented with a Bayesian command estimator due to more than one different filter components(not only commands) in the mode-conditioned filters. Hence, NOCM applies an ordinary GPB1 estimator with r mode-conditioned EKFs. Likelihood function outputs of parallel EKFs are used in the mode probabability calculation. Then, mode-conditioned estimates and covariances are averaged using mode probabilities to calculate updated state estimates and covariances. For the likelihood function, Gaussian proposition given in (5.4) which is proposed in [7] is used. In all simulations, however, this function always behaves as non-Gaussian causing normalization constant used in the mode probability calculation to go to infinity for any observation and command. Hence, NOCM simulations terminate prematurely just after starting simulations. That is why implementation of NOCM in Matlab could not be run and plots and RMSE statistics of NOCM are not given.

PNOCM Algorithm applies a prefilter to reduce shadowing noise in RSSI observations and calculates coarse positions from filtered observations. Coarse positions are given as input to a reduced GPB1 filter(called modified Kalman filter) which has the same structure as of LOCM algorithm. Modified Kalman filter is dedicated to estimate command values. Command estimates are then used in an EKF to find updated state estimates and covariances. PNOCM algorithm claims that the modified Kalman filter structure uses the likelihood function given in (5.3) which is proposed in [1] because it is closer to Gaussianity than the likelihood function given in (5.4) which is proposed in [7]. However as in NOCM algorithm, in all simulations, likelihood function of PNOCM behaves as non-Gaussian and simulations terminate prematurely just after the start. Hence, PNOCM algorithm lacks the Matlab plots and RMSE statistics.

7.1. Future Works

Recall that in LOCN and NOCN algorithms, new state noise is assumed to be $B\boldsymbol{u}_k + \boldsymbol{w}_k$ and correlation between noise components $\boldsymbol{u}_k$ and $\boldsymbol{w}_k$ is ignored during the calculation of the covariance of the new state noise according to the equations (5.1) and (5.2) that are proposed in [1]. A more realistic approach that considers the correlation between these noise components may be developed for calculating covariance of the new state noise $B\boldsymbol{u}_k + \boldsymbol{w}_k$ to obtain better performance.

In LOCM, NOCM and PNOCM algorithms, in principal, tracking system is assumed to be a multiple model system that dynamically switches between models or modes and command process $\boldsymbol{u}_k$ is assumed to be the mode variable. In this approach, likelihood functions are used to calculate mode probabilities. Two Gaussian propositions given in (5.3) and (5.4) proposed in [1] and [7] respectively are used for the likelihood function of LOCM algorithm. A new likelihood function that performs better than these Gaussian propositions for linear models of LOCN algorithm may be developed. For NOCM and PNOCM algorithms the same Gaussian likelihood function propositions are used and they do not work at all in simulations. For these algorithms, a new likelihood function that more realistically considers the nonlinear structure of the RSSI observation model may be developed.

In algorithms that use nonlinear RSSI observation model, EKF is mainly used to handle nonlinearity. Only in NOCN algirithm UKF is used in addition to EKF. However, although UKF performs better than EKF in simulations of NOCN algorithm, both EKF and UKF result in very large RMSE statistics in NOCN simulations and EKF do not work at all in NOCM and PNOCM simulations. Other suboptimal, nonlinear solutions such as higher order EKF, iterated EKF or particle filters may be used for these algorithms to obtain better performances.

REFERENCES

1. Zaidi Z. R. and Mark B. L., "Real-Time Mobility Tracking Algorithms For Cellular Networks Based on Kalman Filtering", *IEEE Transactions on Mobile Computing,* Vol. 4, No. 2, pp. 195-208, March/April 2005.

2. Julier S. J., Uhlman J. K. and Durant-White H. F., "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators", *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, pp. 477-482, March 2000.

3. Singer R. A., "Estimating Optimal Tracking Filter Performance for Manned Manuevering Targets", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-6, No. 4, pp. 473-483, July 1970.

4. Moose R. L., "An adaptive state estimation solution to the manuevering target problem", *IEEE Trans. Automat. Contro.,* vol. AC-20, June 1975.

5. Gholson N. H. and Moose R. L., "Maneuvering Target Tracking Using Adaptive State Estimation", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-13, No. 3, pp. 310-317, May 1977.

6. Moose R. L., Vanlandingham H. F. and McCabe D. H., "Modeling and Estimation for Tracking Manuevering Targets", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-15, No. 3, pp. 448-456, May 1979.

7. Liu T., Bahl P. and Chlamtac I., "Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 6, pp. 922-936, August 1998.

8. Shanmugan K. S. and Breipohl A. M., *Random Signals: Detection, Estimation and Data Analysis,* John Wiley & Sons, New York, 1988.

9. Bar-Shalom Y., Rong Li X. And Kirubarajan T., *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, New York, 2001.

10. Ristic B., Arulampalam S. and Gordon N., *Beyond The Kalman Filter: Particle Filters For Tracking Applications*, Artech House, Boston and London, 2004.