GNSS Pseudorange Smoothing : Linear vs Non-linear Filtering Paradigm

Khurram Mazher
School of Science and Engineering
Lahore University of Management Sciences
khurram.usman@lums.edu.pk

Muhammad Tahir
School of Science and Engineering
Lahore University of Management Sciences
tahir@lums.edu.pk

Khurram Ali
Department of Electrical Engineering
COMSATS Institute of Information Technology
khurram.ali@ciitlahore.edu.pk

Abstract*—Improving the accuracy of position and velocity estimates from single frequency Global Navigation Satellite System (GNSS) receivers is of paramount importance due to their wide spread role in new and emerging real time applications requiring higher levels of accuracy. Single frequency receiver has only two independent measurements: code delay and carrier phase measurements. Pseudorange estimates from code delay measurements are unambiguous but noisy while those from carrier phase measurements are precise but suffer from integer ambiguity problem. In order to obtain high accuracy without explicitly solving for integer ambiguities, Carrier-Smoothed-Code (CSC) methods are very effective. These methods rely on combining the precise but ambiguous carrier phase measurements with noisy but unambiguous code phase measurements to get a smoothed and unambiguous estimate of the pseudorange for each satellite. Several CSC methods have appeared in the last few decades, almost all of them are based on Hatch filter which linearly combines code delay and carrier phase measurements. The main drawback of standard Hatch filter and its variants is the assumption that receiver is either static or is slowly moving and there are no cycle-slips. Beyond linear filtering solution, a non-linear filtering framework can also be devised for CSC process. A general framework to analyze and compare the performance of linear and non-linear filtering approaches towards CSC process especially in the presence of cycle-slips is missing. The main goal of this paper is to provide a significant contribution in this domain. It is shown that Hatch filter problem can be formulated in terms of Kalman filtering problem whose performance can be significantly improved after adequate tuning of process noise statistics. Effect of inaccuracies in estimating the measurement noise statistics on the final pseudorange estimates is also presented. A Gaussian sum non-linear filter is devised whose performance is compared with CSC methods based on linear filtering including Kalman filter. Simulation results, supported by real GNSS data, are provided showing the trade-off between better noise rejection, robustness against cycle-slips, convergence speed and computational complexity.

TABLE OF CONTENTS

978-1-4673-7676-1/16/$31.00

1. INTRODUCTION

Global Navigation Satellite System (GNSS) transmits Direct Sequence Spread Spectrum (DSSS) signals by a constellation of satellites. Using these signals, the receiver makes the distance measurements, called pseudoranges, from at least four satellites enabling itself to compute Position, Velocity and Time (PVT) estimates. The distance measurements can be made either on incoming signal code or on incoming carrier phase. Both of these measurements are affected by satellite noise and errors, receiver noise and atmospheric biases such as those introduced by ionosphere, troposphere and multipath [1]. The receiver noise and multipath errors in the carrier phase observables are very small as compared to code delay observables. However, carrier phase observables are ambiguous being affected by integer number of carrier cycles, called integer ambiguity. So, pseudorange observables from code measurement are noisy and unbiased, while those from carrier measurements are precise but biased due to unresolved integer ambiguity problem [2]. Integer ambiguity resolution is not a trivial task especially for a single frequency receiver operating in stand-alone mode. The best possible position estimate accuracy is obtainable from dual frequency receivers using carrier phase observables. However dual frequency receivers are very expensive and take about 20-40 minutes to reach higher level of accuracies [3]. This aspect makes their scope limited to very special positioning applications like geodetic surveys etc. On the other hand, novel emerging applications of GNSS are requiring higher levels of accuracies in real time operations. For instance, many emerging roadway applications require in-lane level vehicle's position estimates [4]. So, it becomes highly desirable to improve the accuracy of single frequency GNSS receiver which are natural and cost effective solutions for such applications.

In order to get the best possible accuracy from single frequency receivers using both code and carrier observables without explicitly solving for integer ambiguities, Carrier-Smoothed-Code (CSC) techniques have been proposed in the literature [2]. These techniques benefit from precise carrier phase measurements by combining them with unambiguous code phase measurements to get a smoothed unambiguous pseudorange estimate for each satellite. There are few factors which determine the performance of any CSC method. These factors include steady-state pseudorange error, robustness to cycle slips, convergence speed and robustness against ionosphere divergence error. CSC methods can be devised either in *range domain* or in *position domain* [5]. While, *position domain* methods apply carrier smoothing to code observables within navigation solution computation, *range domain* techniques combine code and carrier phase observables to form smoothed pseudoranges for each satellite before computing

navigation solutions. *Position domain* methods are suitable only for receivers working in differential modes [6]. They also involve an intricate design of filter structures, such as Kalman filters incorporating double-differenced carrier phase measurements, for navigation solution computation. These structures also require quite good estimates of noise covariance of pseudorange and carrier phase measurements [7]. While some advantages of using position domain CSC methods over range domain CSC methods have been reported in the literature [8], a complete performance comparison of both in terms of all the factors reported above is still missing. In this paper, our focus is on comparison of two sub-domains for *range domain* CSC techniques namely linear vs nonlinear techniques. Linear techniques rely on linearly combining code and carrier phase measurements while non-linear techniques employ some nonlinear filter structure to get smoothed pseudorange. Comparison is performed in terms of steady-state pseudorange error, robustness to cycle slips and convergence speed. Ionosphere divergence error has not been chosen for comparison in this work. If smoothing intervals are small, this error is not significant in normal conditions i.e. in the absence of any ionosphere storm. There are also some techniques which can estimate this error with quite good accuracy which can be used to remove it before proceeding to smoothing algorithm e.g. the technique reported in [9].

State-of-the-art

Hatch posed the first ever proposal to combine code and carrier measurements using their linear combination in range domain [10]. The technique is usually termed as *Hatch Filter*. Basic idea behind this recursive method is to use the time differenced carrier phase measurements and raw code pseudoranges as current measurements. Use these measurements and previous estimate of smoothed pseudorange with proper weights to get current estimate of smoothed pseudorange. Selection of proper weights is very important for better performance therefore a number of variants of Hatch filter are proposed in the literature which try to optimize the weights. Hatch himself proposed an improvement in original algorithm by proposing epoch independent weights [11]. Lachapelle proposed a variant by reducing the weights by a constant from epoch to epoch [12]. Meyerhoff and Evans also proposed similar variants [13]. Apart from these, a number of other variants have also been proposed such as [14] and the techniques reported in [15]. Hatch filter and its variants use a very crude way of combining both observables which does not take into account any noise statistics on either code or carrier measurements. They are actually designed for static or low dynamic receivers in an environment which does not suffer from cycle-slip problem. If there are frequent cycle-slips, they will be converted to an equivalent error on smoothed pseudorange because in steady state Hatch Filter almost relies entirely on carrier phase measurements. So, in dynamic urban environments where carrier phase measurements are affected by blockages, foliage and cycle slips the performance of these methods deteriorates rapidly.

Alternatively, one can use a linear Kalman Filter, in range domain, for combining code and carrier measurements. We show that this type of Kalman filter actually reformulates Hatch filter equation into Kalman prediction and update equations by assuming linear model and Gaussian noise statistics for process and measurement models. A direct comparison of this structure with Hatch filter and its variants is missing. We actually show that using Kalman based smoothing we get significant performance advantage in terms of error reduction and convergence speed.

Another approach is to use nonlinear stochastic filter for smoothing which can take into account not only the nonlinearities in the models, if any, but also non-Gaussian noise statistics [16]. This situation arrives, for example, when a temporary signal blockage occurs and phase measurements are corrupted temporarily due to cycle slips. In this case, the integer ambiguity is changed from previous value. So, the probability density function (PDF) of carrier phase measurements becomes multimodal. Non-linear filter estimates are obtained by propagating posterior probability density functions in two steps: prediction and filtering. Then adopting some criterion for final smoothed pseudorange estimate e.g. maximum a posterior (MAP) or minimum mean square error (MMSE) etc.

Contribution

In this paper, we address the problem of smoothing GNSS pseudorange from code and carrier phase measurements in both linear and non-linear paradigms. We choose some representative algorithms from both paradigms to compare their performance in terms of smoothed pseudorange Mean Square Error (MSE), cycle slip performance and convergence speed. We start by comparing the performance of Kalman filter and Hatch filter and its variants in terms of convergence speed and MSE in the presence of cycle slips. It is shown that using Kalman filter formulation, one can actually get more control on the smoothing behavior of algorithm. A discussion is carried out for proper tuning of Kalman filter parameters by analyzing their affect on filter's performance. Process noise covariance is shown to be the critical parameter whose value determines the trade-off between noise rejection and convergence speed. The affect of measurement noise inaccuracies is also discussed and a scheme is developed to online estimate the measurement noise covariance from Kalman residuals. Finally, we devise a Gaussian Sum non-linear filter for comparison with other linear filters. More specifically, we consider the following five algorithms for comparison:

1. Traditional Hatch Filter with time varying weights.
2. A variant of Hatch filter which provides an estimate based on subsequent measurements.
3. Traditional Kalman Filter implementing Hatch Algorithms.
4. Traditional Kalman Filter where measurement noise statistics are estimated online.
5. A stochastic nonlinear filter.

A simulation environment is set-up to simulate code and carrier phase measurements along with possibility of introducing a number of cycle slips at any desired position within the data. All algorithms are tested and their performance is evaluated based on the smoothed pseudorange mean square error as a function of variance on code phase measurements. Results indicate that nonlinear filter is more robust to perturbations due to signal outage and cycle slip problem but suffer from problem of very high computational complexity. All the algorithms are also tested with real Global Positioning System (GPS) data.

Another important contribution is to develop a framework for testing the nonlinear filter with real data which involves many implementation issues. For instance, we need to provide the filter with a rough estimate of integer ambiguity. We have shown that actually a very crude estimate of integer ambiguity using the average of difference of code and carrier measurements along with sufficient large number of modes of measurement likelihood function is sufficient for filter operation.

2. SIGNAL MODEL

GNSS signal received by the user terminal is a radio-frequency signal which for a single satellite can be written as

$$s(t) = \sqrt{2C}X(t-\tau(t))\cos(2\pi(f_o(t-\tau(t)) + \phi_o) \quad (1)$$

where C is the received signal power, $X(t)$ is the modulating signal of the satellite which is usually equal to the product of spreading code $c(t)$, sub-carrier $s_c(t)$ and navigation data $d(t)$, $\tau(t)$ is the propagation delay, f_o is the nominal carrier frequency and ϕ_o is a random unknown phase. User terminal can make the range measurements from the satellite by $\hat{\tau}(t)$ which is the estimate of $\tau(t)$. This estimate can be obtained by synchronization the incoming code with local code in code tracking loop. From this estimate, range measurements, *code delay observable* or *code pseudorange*, can be obtained as

$$\rho_c(t) = c\hat{\tau}_c(t) = r(t) + c\Delta t_{sr}(t) + I(t) + T(t) + \hat{\epsilon}_c(t) \quad (2)$$

where $\rho_c(t)$ is the measured code pseudorange, $\hat{\tau}_c(t)$ is the propagation delay estimate from code tracking loop, $r(t)$ is the true geometric range between user terminal and satellite, c is the speed of light, $\Delta t_{sr}(t)$ is the clock misalignment error between user and satellite clocks, $I(t)$ is the delay in electromagnetic wave propagation due to ionosphere, $T(t)$ is the propagation delay due to troposphere and $\hat{\epsilon}_c(t)$ contains error contributions due to other error sources such as receiver noise and multipath etc on code delay measurements.

Alternatively, user terminal can also find $\hat{\tau}(t)$ after synchronizing the incoming carrier with local carrier. This estimate will lead to new type of range measurements, called *carrier phase observables* or *carrier phase measurements*, which can be written as

$$\rho_\phi(t) = c\hat{\tau}_\phi(t) = r(t) + c\Delta t_{sr}(t) - I(t) + T(t) - \lambda N + \hat{\epsilon}_\phi(t) \quad (3)$$

where $\rho_\phi(t)$ is the carrier phase measurement at time t, $\hat{\tau}_\phi(t)$ is the propagation delay estimate from carrier tracking loop, λ is the wavelength of incoming radio-frequency carrier wave which is equal to 0.19 m for GPS C/A signal, N is the integer ambiguity associated with the incoming carrier and $\hat{\epsilon}_\phi(t)$ contains error contributions due to other error sources such as receiver noise and multipath etc on carrier phase measurements. There are three important differences between (2) and (3):

1. Multipath and receiver noise affect code and carrier measurements differently depending on the precision of the process inside code and carrier tracking loops. Considering that carrier wavelength is much smaller than ranging code chip duration, the precision obtainable using carrier phase measurements is much higher as compared to code delay measurements. So, generally, the ratio $\hat{\epsilon}_c(t)/\hat{\epsilon}_\phi(t)$ is very large indicating that carrier phase measurements are much more precise than their code counterparts.
2. $\tau_\phi(t)$ is estimated inside carrier tracking loop by *integrated carrier phase* which is found not only from the current measured fractional part of the phase but also an integer part which comes from a counter which is counting integer number of carrier wave cycles since satellite lock-on. So, the total number of integer carrier wave cycles between user and satellite at the time of lock-on must also be known. This quantity is called *integer ambiguity*, denoted as N in (3). Generally, it is unknown which needs to be resolved for only carrier phase based positioning. This quantity is unique for each satellite and needs to be tracked continuously. It remains constant as long as user terminal is tracking and counting full cycles continuously since the satellite was first locked on. This quantity is changed when the user receiver loses the lock momentarily due to signal blockage etc. As a result, it misses the count of some integer number of cycles. The phenomenon is known as *cycle slip*. Cycle slips is detrimental phenomenon and can cause large errors in final estimates especially for CSC methods.
3. Finally, there is opposite sign to ionosphere delay term in (3) w.r.t. (2). Ionosphere being highly dispersive medium delays incoming code and carrier signals differently. In fact, code waveform is delayed while carrier signal is *advanced* by the same amount; hence an opposite sign in (3). This fact leads to an error in estimates of CSC methods called *code carrier divergence error* which becomes significant if larger smoothing intervals are used for CSC methods.

In order to minimize the errors due to ionosphere and troposphere, single frequency receivers operating in non-differential mode has to rely on some mathematical model for these mediums whose parameters are broadcast by the satellites. Assuming that these errors have been minimized using some method as described in [1], [2] and [17], we can write the simplified model for *error free* code and carrier observables. At discrete-time epoch n, code pseudorange, $\rho_c(n)$, can be written as

$$\rho_c(n) = r(n) + c\Delta t_{sr}(n) + \epsilon_c(n) = \rho(n) + \epsilon_c(n) \quad (4)$$

where $\rho(n) = r(n) + c\Delta t_{sr}(n)$ is the error free pseudorange and $\epsilon_c(n)$ contains the effect due to receiver noise and residual ionosphere and troposphere errors. Assuming that these errors are very small, we can model $\epsilon_c(n)$ as zero mean Gaussian random variable with variance σ_c^2 i.e. $\epsilon_c(n) \sim \mathcal{N}(0, \sigma_c^2)$. The value of σ_c^2 depends on Signal to Noise Ratio (SNR) and nature of code tracking loop. For example, for common delay locked loop, its value can be in the range 10-100 meters depending on the type of correlator and loop noise bandwidth for a fixed value of SNR, while for Kalman filter based tracking architectures it will seldom exceed 1-5 m range.

Also, for carrier phase measurements, we can write

$$\rho_\phi(n) = \rho(n) - \lambda N + \epsilon_\phi(n) \quad (5)$$

where $N = 0, \pm 1, \pm 2, \cdots$ is unknown integer ambiguity and $\epsilon_\phi(n)$ includes residual errors on carrier phase measurements. This quantity can also be modeled as zero mean Gaussian random variable with variance σ_ϕ^2 i.e. $\epsilon_c(n) \sim \mathcal{N}(0, \sigma_\phi^2)$. Due to the reasons already explained, typically, the ratio $\sigma_c^2/\sigma_\phi^2 > 100$ [16].

3. ALGORITHMS DESCRIPTION

This section provides details of various CSC algorithms considered in this paper. Basic idea behind any CSC method is to get an estimate of smoothed pseudorange using some combination of code and carrier observables at each epoch i.e.

$$\rho_{sm}(n) = f(\rho_c(n), \rho_\phi(n), \rho_{sm}(n-1)) \quad (6)$$

where $\rho_{sm}(n)$ is the smoothed pseudorange at epoch n and $f(.,.)$ is some recursive operator , linear or non-linear, which combines code and carrier observables. The success of any CSC algorithm in the presence of different impairments such as high receiver noise, cycle slips and divergence error

depends on the nature of $f(.,.)$ that this CSC algorithm is employing. We start our discussion with the basic Hatch algorithm.

Hatch Filter

The idea behind Hatch Filter is a time-varying weighted average of code and carrier measurements. In order to deal with N, two subsequent phase measurements are subtracted to get *delta phase measurements*. Then, at each epoch these delta measurements are added to previous smoothed pseudorange. Resulting quantity is combined with current code pseudorange by assigning proper weights to both. Initially, total weight is assigned to code measurements. This weight is updated at each time step with more and more weight being given to carrier measurements term subsequently, up to a certain averaging constant. Beyond this averaging constant the filter can diverge because of the opposite sign of ionospheric delays in code pseudorange and carrier phase observables. The Hatch filter is mathematically formulated as [10]

$$\rho_{sm}(n) = W(n)\rho_c(n) + (1 - W(n))[\rho_{sm}(n-1) + \rho_\phi(n) - \rho_\phi(n-1)] \quad (7)$$

where $0 < W(n) < 1$ is the weight factor which is updated as

$$W(n) = W(n-1) - \gamma \quad (8)$$

where γ is the averaging constant and is typically set as 0.01 or 0.02 which corresponds to smoothing interval of 100 s or 50 s respectively for the pseudorange data coming at 1 Hz rate. Typically, $W(1) = 1$ and $W(n) = \gamma$ in steady state. Increasing the length of smoothing interval will introduce more smoothing but at the same time divergence error is increased. So, there is a limit on maximum smoothing interval e.g. it is 100 s for LAAS users [18].

Modified Hatch Filter

There is a variety of CSC methods based on Hatch filter idea. The one we have chosen is quite recent proposed in [18]. This modified Hatch Filter formulation combines code and carrier phase observables in two stages: prediction and filtering. At each epoch the algorithm finds a new estimate of the code pseudorange at the first epoch using the current code pseudorange and current and first carrier phase observables i.e.

$$\hat{\rho}_c^{(n)}(1) = \rho_c(n) - [\rho_\phi(n) - \rho_\phi(1)] \quad (9)$$

where $\hat{\rho}_c^{(n)}(1)$ is the estimate of $\rho_c(1)$ using current code observable $\rho_c(n)$ and current phase observables $\rho_\phi(n)$ along with first phase observable $\rho_\phi(1)$. Different estimates of the first epoch code pseudorange, obtained at subsequent epochs are then averaged over a window of length *m* to obtain a refined estimate of the code pseudorange of the first epoch as

$$\bar{\rho}(1) = \frac{1}{m}\sum_{i=1}^{m} \hat{\rho}_c^{(i)}(1) \quad (10)$$

This is the prediction stage. The filtering stage now constructs the smoothed measurement at each epoch *n* by using the refined initial code pseudorange and the raw phase observables as:

$$\rho_{sm}(n) = \bar{\rho}(1) + [\rho_\phi(n) - \rho_\phi(1)] \quad (11)$$

This method seems promising as it depends only on code phase measurements, so, we can expect good noise rejection performance. However, as explained later, there are some serious drawbacks of this approach especially in the presence of cycle slips.

Kalman Filter

The Kalman filter formulation is the casting of the Hatch Filter equation into the Kalman filter format. In Hatch filter, the step size and final weights are fixed. It does not take into account the variance of noise on the code and carrier observables. Kalman filter setting allows the filter to take the noise statistics into account and calculate the optimal weight in this setting. We can decompose (7) as

Prediction Equation:

$$\rho_{sm}^{-}(n) = \Phi_s \rho_{sm}^{+}(n-1) + [\rho_\phi(n) - \rho_\phi(n-1)] \quad (12)$$

where $\rho_{sm}^{-}(n)$ is *a priori* estimate of smoothed pseudorange at epoch n, $\rho_{sm}^{+}(n-1)$ is *a posteriori* estimate of smoothed pseudorange at epoch $n-1$ and $\Phi_s = 1$ is state transition matrix in this formulation.

Update Equation:

$$\rho_{sm}^{+}(n) = (I - K(n)H)\rho_{sm}^{-}(n) + K(n)\rho_c(n) \quad (13)$$

where

$K(n) =$ Kalman gain at epoch n= $P^{-}(n)H^T \left(HP^{-}(n)H^T + R\right)^{-1}$

$H = 1$ is measurement matrix

$P^{-}(n)$ = Predicted state vector covariance at epoch n= $\Phi_s P^{+}(n)\Phi_s^T + Q$

$R =$ Measurement noise variance i.e. variance of $\rho_c(n)$

$Q =$ Process noise variance

$P^{+}(n)$ = Updated state vector covariance = $(I - K(n)H)P^{-}(n)$

Then at each time epoch, our smoothed pseudorange is equal to *a posteriori* estimate i.e. $\rho_{sm}(n) = \rho_{sm}^{+}(n)$. This simple formulation of Kalman filtering described in (12) and (13) gives more parameters to control the behavior of smoothing process. Proper tuning of Q, P_{ini} and R is much necessary to get the best out of Kalman filter. We have observed the following facts during different simulation runs:

- For better estimates and good convergence, $Q << P_{ini}$
- Q must be very small for low errors in final estimates. Increasing Q increases the convergence rate while error in final estimates is increased.
- Estimate of R should be accurate enough for better results.

Kalman Filter with measurement noise estimation

Since the estimation of measurement noise statistics R is crucial for final estimates, here, we keep everything same as that of Kalman filter formulation except a method is described to estimate R online. R can be estimated from Kalman filter residuals as described below. The measurement in Kalman filter is equal to code pseudorange $\rho_c(n)$, the residuals $r_z(n)$ can be constructed inside Kalman filter as

$$r_z(n) = \rho_c(n) - H\rho_{sm}^{-}(n) \quad (14)$$

where the sequence $r_z(n)$ is independent with variance $\sigma_r^2 = R + HP^{-}(n)H^T$. So, R can be estimated if σ_r^2 and variance

of $HP^{-}(n)H^{T}$ is known. Considering a sample window of length N_w, we can estimate σ_r^2 as

$$\sigma_r^2 = \frac{1}{N_w - 1}\sum_{k=1}^{N_w}(r_z(k) - m_z)(r_z(k) - m_z)^T \tag{15}$$

where m_z is the sample mean estimator of residuals given as

$$m_z = \frac{1}{N_w}\sum_{k=1}^{N_w} r_z(k) \tag{16}$$

The variance of $HP^{-}(n)H^{T}$ over the sample window of N_w can be computed as

$$\frac{1}{N_w}\sum_{k=1}^{N_w} HP^{-}(k)H^{T} \tag{17}$$

In this way, we get the estimate of R at each epoch. These formulae can also be formulated in recursive form as suggested in [19].

Gaussian Sum Nonlinear Filter

The idea behind this approach is to recursively estimate the pseudorange dynamics from measurement likelihood. For this purpose, classical Bayesian filtering technique is used where in the first step, a prediction PDF is obtained through pseudorange dynamics model. This prediction PDF is updated using measurement likelihood to get a filtering PDF in next step. We start by introducing state dynamics model. The approach followed here is similar to the one given in [16] with noticeable differences in state dynamics model and some other implementation aspects.

State Dynamics Model—Let $\rho(t)$ and $v(t)$ be the pseudorange (error free) and relative velocity between a single satellite and user terminal in a generic direction. Then, Singer velocity model assumes that $v(t)$ is a zero-mean first-order stationary Markov process which can be described by [20]

$$\dot{v}(t) = -\alpha v(t) + w(t) \tag{18}$$

where $w(t)$ is zero-mean White noise with constant power spectral density $2\alpha\sigma_n^2$, σ_n^2 is the instantaneous variance of the velocity and α, in units of s^{-1} determines the duration of velocity maneuver. Assuming that our state is a vector $\mathbf{x(t)} = [\rho(\mathbf{t})\ \ \mathbf{v(t)}]^{\mathbf{T}}$, we can write the state space formulation for (18) as

$$\dot{\mathbf{x}}(\mathbf{t}) = \begin{bmatrix}1 & 0\\ 0 & -\alpha\end{bmatrix}\mathbf{x}(\mathbf{t}) + \begin{bmatrix}0\\ 1\end{bmatrix}\mathbf{w}(\mathbf{t}) \tag{19}$$

The discrete time equivalent of (19) is given by

$$\mathbf{x(n)} = \mathbf{\Phi x(n-1)} + \mathbf{\Gamma w(n-1)} \tag{20}$$

with $\mathbf{\Gamma} = [\mathbf{0}\ \ \mathbf{1}]^{\mathbf{T}}$ and state transition matrix, $\mathbf{\Phi}$, can be calculated, for example using Caylay-Hamilton theorem, as

$$\mathbf{\Phi} = \begin{bmatrix}1 & \frac{1-e^{-\alpha T}}{\alpha}\\ 0 & e^{-\alpha T}\end{bmatrix} \tag{21}$$

with T is the time difference between two pseudorange epochs. The process noise in (20), $w(n)$, is again assumed to be zero-mean White noise process with variance σ_w^2. Exact value of σ_w^2 will be function of α and T. Here, we will refer to it as process noise variance.

Measurement Model—Measurement model can be obtained from (4) and (5) as

$$\mathbf{z(n)} \equiv \begin{bmatrix}\rho_c(n)\\ \rho_\phi(n)\end{bmatrix} = \begin{bmatrix}\rho(n)\\ \rho(n) - \lambda N\end{bmatrix} + \begin{bmatrix}\epsilon_c(n)\\ \epsilon_\phi(n)\end{bmatrix} \tag{22}$$

Two noise processes $\epsilon_c(n)$ and $\epsilon_\phi(n)$ are assumed to be independent as they arise from two different and independent measuring processes within carrier tracking loops.

Measurement Likelihood—The likelihood function associated to (4) is conditional Gaussian PDF which is given by

$$p(\rho_c(n)|\rho(n)) = \mathcal{N}(\rho_c(n), \sigma_c^2) \tag{23}$$

The likelihood function associated to (5) is an infinite train of conditional Gaussian PDFs due to the presence of unknown ambiguity term. As mentioned in [16], we will keep only $2L+1$ modes of this infinite series, which give rise to the following likelihood function

$$p(\rho_\phi(n)|\rho(n)) = \frac{1}{2L+1}\sum_{i=-L}^{L}\mathcal{N}\big(\rho_\phi(n) + \lambda[i + D(n)], \sigma_\phi^2\big) \tag{24}$$

In (24), the center position of the likelihood is controlled by $\lambda D(n)$ and this likelihood has L modes on either side. The integer $D(n)$ represents the estimate of N at each time step and integer L represents the amount of uncertainty in that estimate which can be tracked or accommodated by nonlinear filter. Where L will be kept constant at each epoch and $D(n)$ will be adjusted to center the grid around last smoothed pseudorange i.e. $\rho_{sm}(n-1)$. Note that, this approach is different from the one adopted in [16], where the authors center the grid at last code measurement i.e. $\rho_c(n)$. But we have observed that since $\rho_c(n)$ is much more noisy so it causes a lot of fluctuations in estimate of $D(n)$.

Since $\epsilon_c(n)$ and $\epsilon_\phi(n)$ were assumed to be independent, the joint likelihood function $p(\rho_\phi(n), \rho_c(n)|\rho(n))$ is given by the product

$$H(n) \equiv p(\rho_\phi(n), \rho_c(n)|\rho(n)) = p(\rho_\phi(n)|\rho(n))p(\rho_c(n)|\rho(n)) \tag{25}$$

which reduces to the following expression [16]

$$H(n) = \sum_{i=-L}^{L}\gamma_H^{(i)}(n)\mathcal{N}\Big(\rho_H^{(i)}(n), \sigma_H^2(n)\Big) \tag{26}$$

with

$$\gamma_H^{(i)}(n) = \tfrac{1}{2L+1}G\left(\rho_c(n) - \rho_\phi(n) - \lambda[i + D(n)], \sigma_c^2 + \sigma_\phi^2\right)$$

$$G(m, v) = \tfrac{1}{\sqrt{2\pi v}}e^{-0.5\left(\frac{m^2}{v}\right)}$$

$$\rho_H^{(i)}(n) = \tfrac{\rho_c(n)\sigma_\phi^2 + \sigma_c^2(\rho_\phi(n) + \lambda[i + D(n)])}{\sigma_c^2 + \sigma_\phi^2}$$

$$\sigma_H^2(n) = \tfrac{\sigma_c^2\sigma_\phi^2}{\sigma_c^2 + \sigma_\phi^2}$$

Bayesian Filtering—We need to estimate $\mathbf{x(n)}$ recursively. For this purpose, we need to define some PDFs as

Observation Model: $p(\mathbf{z(n)}|\mathbf{x(n)}) = \mathbf{H(n)}$ in (26)

Dynamics Model:
$p(\mathbf{x(n)}|\mathbf{x(n-1)}) = \mathcal{N}(\mathbf{\Phi x(n-1)}, \sigma_{\mathbf{w}}^{\mathbf{2}}\mathbf{\Gamma\Gamma^T})$ from (20)

Prediction PDF: $P(n) \equiv p(\mathbf{x(n)}|\mathbf{z(n-1)})$

Filtering PDF: $F(n) \equiv p(\mathbf{x(n)}|\mathbf{z(n)})$

Since, our observation model is multimodal, hence the use of nonlinear filter for estimation of $\mathbf{x(n)}$ is justified. Any nonlinear filter will involve the following steps for estimation of $\mathbf{x(n)}$:

Prediction: The famous Chapman-Kolmogrove equation gives $P(n)$ as the convolution between $F(n-1)$ and $p(\mathbf{x(n)}|\mathbf{x(n-1)})$ as

$$P(n) = \int p(\mathbf{x(n)}|\mathbf{x(n-1)})\mathbf{F(n-1)}d\mathbf{x(n-1)} \quad (27)$$

Filtering: $F(n)$ is given by the multiplication between $P(n)$ and $H(n)$ as

$$F(n) = d(n)P(n)H(n) \quad (28)$$

where $d(n)$ is a normalizing constant. The estimate $\hat{\mathbf{x}}(\mathbf{n})$ is obtained from $F(n)$ by following some criterion e.g. MMSE, MAP etc.

Gaussian Sum Filter—Exact solution to the Bayesian filtering problem described above is not possible, so, some suboptimal methods must be used to approximate *a posterior* PDFs: $P(n)$ and $F(n)$. Gaussian sum filter approximates them by weighted sum of Gaussian PDFs. The algorithm can be described by the following steps.

Step-1: Initialization: Assuming that $F(n-1)$ i.e. filtering PDF at step $n-1$ is available. Then, we can approximate $F(n)$ and $P(n)$ by $2L+1$ bi-variate Gaussian functions as

$$F(n-1) = \sum_{k=-L}^{L} \gamma_F^{(k)}(n-1)\mathcal{N}\Big(\eta_{\mathbf{F}}^{(\mathbf{k})}(\mathbf{n-1}), \mathbf{V_F(n-1)}\Big) \quad (29)$$

$$P(n) = \sum_{k=-L}^{L} \gamma_P^{(k)}(n)\mathcal{N}\Big(\eta_{\mathbf{P}}^{(\mathbf{k})}(\mathbf{n}), \mathbf{V_P(n)}\Big) \quad (30)$$

Notice that each mode is bi-variate since our state vector $\mathbf{x(n)}$ is two-dimensional. Here

$$\eta_{\mathbf{P}}^{(\mathbf{k})}(\mathbf{n}) = \Phi\gamma_F^{(k)}(n-1) \quad (31)$$

$$\mathbf{V_P(n)} = \mathbf{\Phi V_F(n-1)\Phi^T} + \sigma_{\mathbf{w}}^{\mathbf{2}}\mathbf{\Gamma\Gamma^T} \quad (32)$$

$$\gamma_P^{(k)}(n) = \gamma_F^{(k)}(n-1) \quad (33)$$

In order to start the algorithm at epoch $n=1$, we need to assume the parameters of $P(n)$ so that

$$P(1) = \sum_{k=-L}^{L} \gamma_P^{(k)}(1)\mathcal{N}\Big(\eta_{\mathbf{P}}^{(\mathbf{k})}(\mathbf{1}), \mathbf{V_P(1)}\Big) \quad (34)$$

with

$\gamma_P^{(k)}(1) = \frac{1}{2L+1}$

$\eta_{\mathbf{P}}^{(\mathbf{k})}(\mathbf{1}) = \Big[\rho_{\mathbf{H}}^{(\mathbf{k})}(\mathbf{1}) \;\; \mathbf{v_o}\Big]^{\mathbf{T}}$

$\mathbf{V_P(1)} = \text{diag}\{\sigma_{\mathbf{H}}^{\mathbf{2}}(\mathbf{1}) \;\; \sigma_{\mathbf{v}}^{\mathbf{2}}\}$

where $\rho_H^{(k)}(1)$ is the initial raw estimate of means of all modes, v_o is the initial raw estimate of velocity, $\sigma_H^2(1)$ is the initial uncertainty in $\rho_H^{(k)}(1)$ and σ_v^2 is the initial uncertainty in v_o.

Step-2: Filtering: Now as soon as, $\mathbf{z(n)}$ becomes available, we can form likelihood function $H(n)$ according to (26). Then $F(n) = P(n)H(n)$ is given by [16]

$$F(n) = \sum_{i=-L}^{L}\sum_{l=-L}^{L} \gamma_F^{(i,l)}(n)\mathcal{N}\Big(\eta_{\mathbf{F}}^{(\mathbf{i,l})}(\mathbf{n}), \mathbf{V_F(n)}\Big) \quad (35)$$

where the values of $\gamma_F^{(i,l)}(n)$, $\eta_{\mathbf{F}}^{(\mathbf{i,l})}(\mathbf{n})$ and $\mathbf{V_F(n)}$ are as given in [16]. Note that the number of modes in (35) has been doubled i.e. we have $(2L+1)^2$ modes. At this point we need to discard $2L+1$ modes having very low weight. in this way, only $2L+1$ modes are retained in (35). So, for each l-th mode of $P(n)$, only one mode of $H(n)$ is retained which minimizes the distance $\Big|\eta_{P,1}^{(l)}(n) - \rho_H^{(i)}(n)\Big|$, where $\eta_{P,1}^{(l)}(n)$ is the first element of $\eta_{\mathbf{P}}^{(\mathbf{l})}(\mathbf{n})$ [16]. Surviving modes are re-indexed and re-weighted to get the form of $F(n)$ as given in (29).

Step-3: Prediction: In next step, $P(n)$ is obtained according to (30)-(33). In this way, we propagate and update a posterior PDF at each epoch.

Step-4: MMSE Estimate: At epoch n, MMSE estimate of $\rho(n)$ is given by

$$\rho_{sm}(n) = \int \rho(n)p(\mathbf{x(n)}|\mathbf{z(n)})d\mathbf{x(n)} \quad (36)$$

$$= \sum_{k=-L}^{L} \gamma_F^{(k)}(n)\eta_{F,1}^{(k)}(n) \quad (37)$$

4. SIMULATION RESULTS AND DISCUSSION

In order to set a simulation environment, $\rho(n)$ in (4) and (5) is considered to be a sinusoid as $\rho(n) = 40\sin(2\pi\frac{nT}{5})$ where $T = 1$ ms. Code and carrier observables are generated according to (4) and (5) respectively where σ_ϕ^2 is fixed to $0.0001m^2$ while σ_c^2 can be varied. We present two different type of results. First, we test the convergence speed of each method and robustness of each method against cycle slip phenomenon. These results will provide us a reasonable estimate for the values of parameters of each method to be used subsequently. Then, using these parameters, we present the results indicating the smoothing power of each method.

Convergence Speed and Cycle Slips—In order to test the robustness against cycle slips, we introduced an artificial cycle slip of 20 cycles at epoch 1500 which lasts for the remaining time while $\sigma_c^2 = 20m^2$. Results for different algorithms are presented below.

Hatch Filter: Hatch filter described in Section 3 has only one parameter to control its behavior i.e. averaging constant

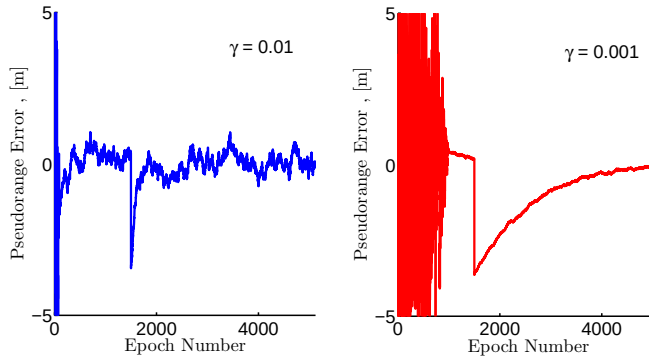

Figure 1. Pseudorange error of traditional Hatch Filter in the presence of cycle slip

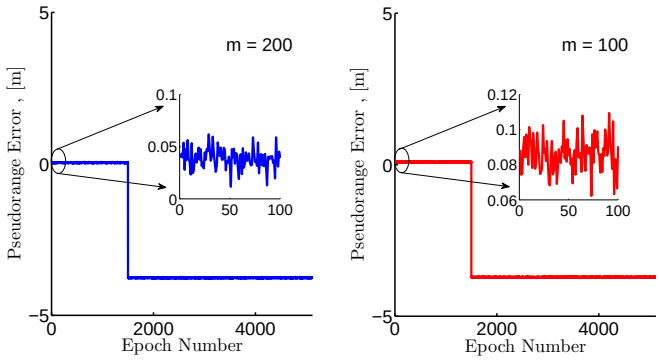

Figure 2. Pseudorange error of modified Hatch Filter in the presence of cycle slip

γ which is also the steady state value of the weight factor assigned to code pseudorange measurements. Results are shown in Fig. 1. The effect of parameter γ is clear from the figure. As we decrease the value of γ, the filter converges slowly, not only initially but also whenever a cycle slip is introduced. However, for smaller values of γ, more smoothing is achieved as the steady state weight on carrier phase measurement is increased. So, the value of γ cannot be decreased to very small number. Keeping in mind the trade-off, $\gamma = 0.01$ is a reasonable number.

Modified Hatch Filter: Modified Hatch filter also has only one parameter for tuning i.e. the window size m over which the estimate of code pseudorange of first epoch $\rho_c(1)$ is smoothed to be used for subsequent processing. In steady state, this filter only uses carrier phase measurements for smoothing. So, the final estimate is supposed to be very precise. However, the introduction of cycle slip will permanently introduce an error which cannot be recovered as we are not using current code phase measurements. This is exactly the behavior observed from the results shown in Fig. 2. Note that although final estimate is very precise, where precision is almost the same as that of only carrier phase observables, but the estimate has small bias which is clear from the zoomed image. This bias is due the reason that $\rho_c(1)$ being noisy will not be close to actual $\rho(1)$. Hence, increasing the value of m decreases this bias which is also evident from the results. However, we cannot increase m beyond a maximum value due to two reasons: first it increases the computational complexity and second in the presence of ionosphere bias, divergence error starts increasing. So, as suggested in [18], a reasonable value for m is 100.

Kalman Filter: Kalman filter formulation gives us much more flexibility to tune the overall smoothing process. We can control the behavior by tuning the values of Q, P_{ini} and R. If we follow the criterion $Q << P_{ini}$, we have observed that it is only Q which controls the convergence speed and amount of final steady state error if the estimate of R is correct. As an example, Fig. 3 shows the behavior in the presence of cycle slips for $P_{ini} = 10$ and $R = 20$ which is the true value of measurement noise variance. In this situation, decreasing the value of Q improves the accuracy of final estimate while convergence speed reduces. Larger inaccuracies in the estimate of R affect the behavior of Kalman filter. In order to see this affect, Fig. 4 shows the behavior of Kalman filter when true value of measurement noise is 20 but the value of R used is different from the true value. Clearly, under-estimation in value of R increases the

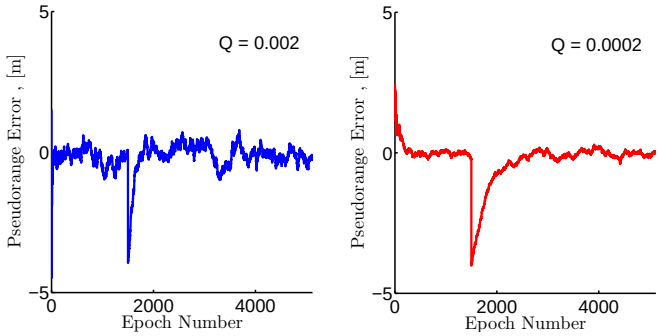

Figure 3. Pseudorange error of Kalman Filter in the presence of cycle slip when estimate of R is perfect

convergence speed while final estimate is more noisy. This type of behavior is comprehendable as when R is small, the value of $K(n)$ will increase, that means, we are placing more weight to $\rho_c(n)$. This will increase the convergence speed but the final estimate will be more noisy. An exactly opposite behavior is observed from Fig. 4 for over-estimation of value of R.

Kalman Filter with online estimation of R: Here, we estimate R using the algorithm described in Section 3. Increasing the window size N_w will increase the initial transient but the estimate of R becomes more smooth. So, keeping this trade-off in mind, we have selected $N_w = 1000$ and results are

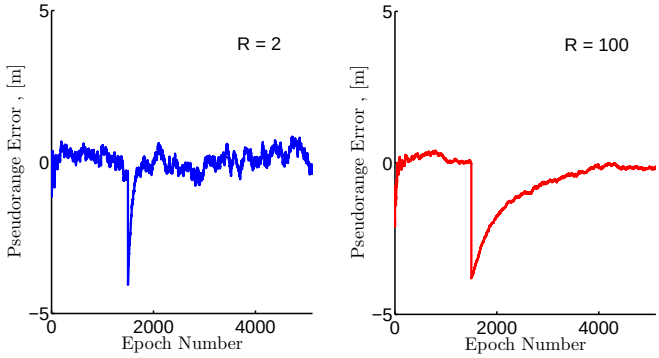

Figure 4. Pseudorange error of Kalman Filter when estimate of R is incorrect, $Q = 0.0002$

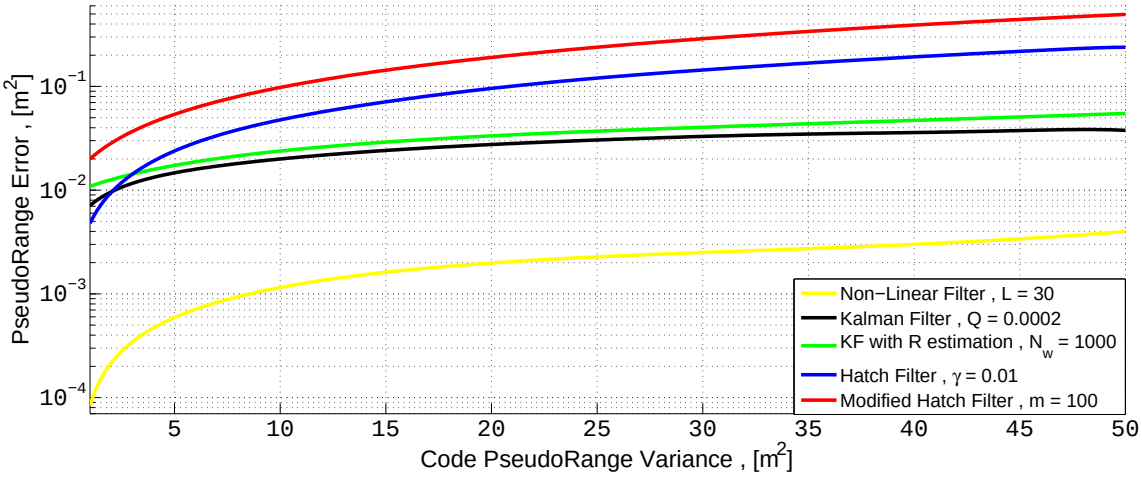

Figure 7. Pseudorange MSE of of all algorithms

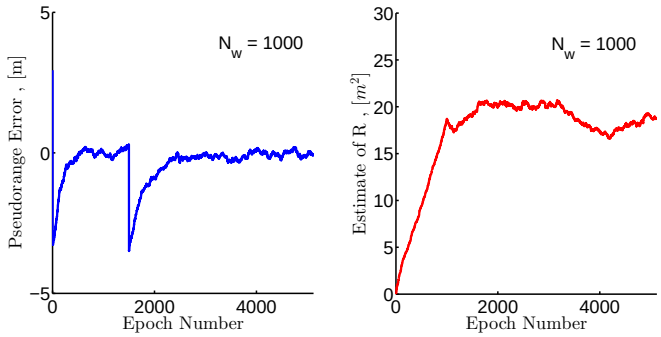

Figure 5. Pseudorange error of Kalman Filter with online estimation of R, true value of $R = 20$, $Q = 0.0002$

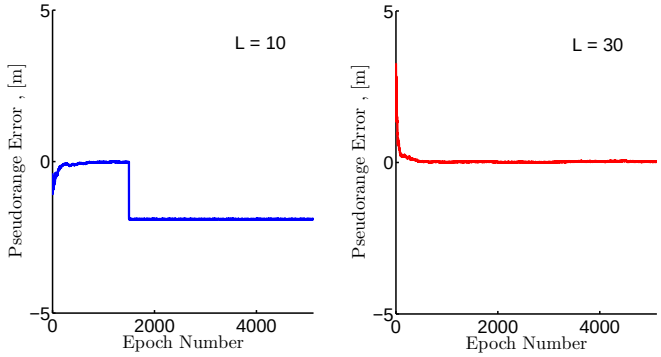

Figure 6. Pseudorange error of nonlinear filter

shown in Fig. 5.

Nonlinear Filter: Gaussian sum filter described in Section 3 has been implemented for smoothing the pseudorange using code and carrier observables. This is very powerful algorithm and here we discuss some aspects. Most important parameter of the filter is L which controls the total number of modes of the likelihood function. L must be wide enough to cover the uncertainty due to initial ambiguity or maximum amount of cycle slips that can be introduced during the receiver operation. As an example, we have selected two different values for L, 10 and 30, where first value is not enough to cover the introduced amount of cycle slipping i.e. 20 cycles while second value is enough. For both scenarios, pseudorange error results are shown in Fig. 6.

It is clear that for $L = 30$, there is no effect of cycle slips on the estimates. On the other hand, for values of L not covering the complete range of maximum amount of cycle slips, the filter has a random bias in its output estimate. The reason behind this estimate lies in the fact that when L is small, $P(n)$ and $H(n)$ gets displaced by large amount resulting in either very small overlap or no overlap at all. Hence, the resulting $F(n)$ can have any mode with maximum weight.

Another important parameter of this filter structure is $D(n)$ which indicates the value of ambiguity at each epoch. This parameter can be estimated either from last code measurement $\rho_c(n)$ or from last $\rho_{sm}(n-1)$. We have selected to use $\rho_{sm}(n-1)$ which produces much smoother estimates.

Pseudorange MSE Performance— In next results, we aim to compare MSE performance of all the algorithms. For this purpose, results are generated for each algorithms with the values of different parameters chosen from discussion in the previous section. The value of variance on code observables i.e. σ_c^2 is varied while keeping $\sigma_\phi^2 = 0.0001m^2$. For each value of σ_c^2, MSE of pseudorange is evaluated in steady state for each run and then it is averaged over all number of iterations. This procedure is repeated for each algorithm. Results are shown in Fig. 7 in the absence of any cycle slip. MSE is measured in steady state for all the algorithms. Nonlinear filter clearly outperforms all other methods followed by Kalman filter with perfect estimate of R. Note that in the case of modified Hatch filter, window size is chosen to be equal to 100 which is not sufficient to remove all the bias due to imperfect estimate of $\rho_c(1)$. This bias lifts the curve to larger values. If this bias is removed from the final estimate, its performance will be equivalent to pure carrier phase observables.

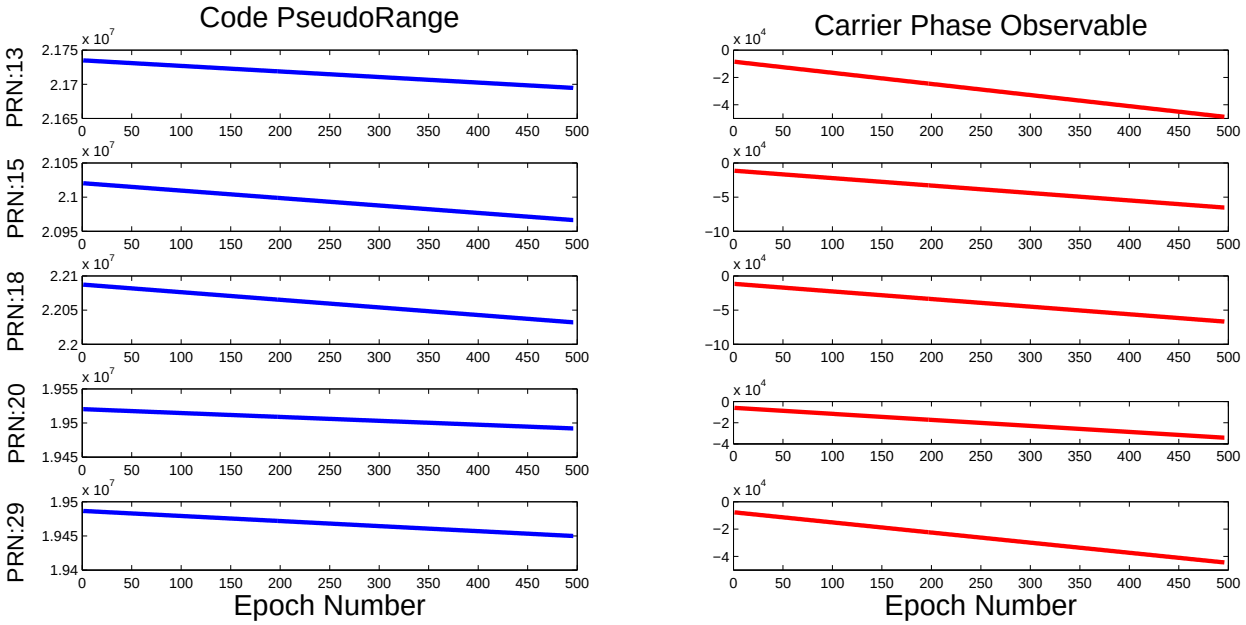

Figure 8. Pseudorange and carrier phase observables from collected data

5. Real GPS Data Results

All the smoothing algorithms, described in Section 3, were tested on real GPS collected using GNSS RF signal recorder from iP-Solutions. The Intermediate Frequency (IF) data was collected outside School of Science and Engineering (SSE) building in Lahore University of Management Sciences (LUMS). IF data was processed using Kalman filter based tracking loop to get code and carrier phase observables. Resulting code and carrier phase observables from five visible satellites are shown in Fig. where the variance of the noise on $\rho_c(n)$ is in the range 1-5 m^2 while on $\rho_\phi(n)$ is of the order of 10^{-4} m^2. These observables are processed by CSC methods considered in this paper and the computed position using least-square navigation solver is shown in Fig. 9. Results are very much in agreement with what we have already observed in simulations. Although it may not be directly observable from the figure due to very close proximity of all the points, the minimum variance is obtained in the case of nonlinear filter followed by Kalman filter and then Hatch filter. For nonlinear filter to work, we need to provide a rough estimate of integer ambiguity which goes directly as starting value of the integer $D(n)$. We have calculated this value as

$$D(n) = \frac{1}{n}\sum_{i=1}^{n} \rho_c(i) - \rho_\phi(i) \tag{38}$$

This value of $D(n)$ along with sufficient larger value for L was sufficient for nonlinear filter to give the smoothed estimate which were used to compute the position displayed in Fig. 9.

Figure 9. Position computation using real GPS data after smoothing pseudoranges from each algorithm

6. Conclusions

The problem of smoothing GNSS satellite pseudorange using both code and carrier observables in a filter structure has been considered. Some representative algorithms from linear and nonlinear filtering domains have been chosen for comparison. Classical Hatch filter was formulated in a simple Kalman filter based structure which gives more flexibility and control to tune the behavior of smoothing process. A nonlinear filter with simplified state dynamics model was devised to work with simulated as well as real GPS data. Simulation results presented in the paper, along with real GPS data results, indicate the superiority of nonlinear filter over all the linear domain filters. However, the computational complexity involved in estimating pseudorange from nonlinear filter is much higher as compared to simple Kalman or Hatch filter. The choice of any single algorithm depends highly on the type of application. For instance, nonlinear filter has been observed to be highly robust towards cycle slips given the number of modes have been chosen to be sufficiently large. On the other hand, large number of modes also increases the computational complexity. So, if number of modes are not enough to span the whole uncertainty of cycle slips, nonlinear filter has a random bias in its output and the filter cannot recover this bias by its own. In such situations, Kalman filter would be the better choice which will always be affected by cycle slip but will also recover from it after some time. As a future work, it will be interesting to completely characterize the performance of nonlinear filter in different environments especially in the presence of ionosphere storms.

REFERENCES

[1] E. Kaplan and C. Hegarty, Eds., *Understanding GPS Principles and Applications*. Norwood, MA: Artech House, 2006.

[2] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*. Lincoln, MA: Ganga-Jamuna Press, 2006.

[3] HANS VAN DER MAREL and PETER F. DE BAKKER, "Single versus Dual-Frequency Precise Point Positioning," *InsideGNSS*, July/August 2012.

[4] Marcus Obst, "Bayesian Approach for Reliable GNSS-based Vehicle Localization in Urban Areas," Ph.D. dissertation, Technischen Universitat Chemnitz, 2014.

[5] Elena Alonso Yebra, "ALGORITHMS TO CALCULATE GPS POSITION USING THE PHASE OF CARRIER SIGNALS," Master's thesis, Universidad San Pablo, 2008.

[6] H. Lee and C. Rizos, "Position-domain hatch filter for kinematic differential gps/gnss," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 44, no. 1, pp. 30–40, January 2008.

[7] J. Lee, H. Kim, K. Choi, J. Cho, and H. Lee, "Adaptive position-domain carrier-smoothed code filter based on innovation sequences," *Radar, Sonar Navigation, IET*, vol. 8, no. 4, pp. 336–343, April 2014.

[8] H. Lee, C. Rizos, and G.-I. Jee, "Position domain filtering and range domain filtering for carrier-smoothed-code dgnss: an analytical comparison," *Radar, Sonar and Navigation, IEE Proceedings* -, vol. 152, no. 4, pp. 271–276, Aug 2005.

[9] S. Sen and J. Rife, "Nonlinear filter for ionosphere divergence error reduction in laas," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 48, no. 2, pp. 981–990, APRIL 2012.

[10] R. Hatch, "The Synergism of GPS Code and Carrier Measurements," in *Proceedings of 3rd Int. Geodetic Symposium on Satellite Doppler Positioning*, vol. vol 2, New Mexico State University, 8-12 February 1982, pp. 1213–1231.

[11] R. Hatch, "Dynamic Differential GPS at the centimetre level," in *Proceedings of 4th Int. Geodetic Symposium on Satellite Positioning*, vol. vol 2, Austing, TX, April-May 1986, pp. 1287–1298.

[12] G. Lachapelle, J. Hagglund, W. Falkenberg, P. Bellemare, M. Casey, and M. Eaton, "GPS land kinematic positioning experiments," in *Proceedings of 4th Int. Geodetic Symposium Satellite Positioning*, vol. 2, Austin, TX, April-May 1986, pp. 1327–1344.

[13] S.L. Meyerhoff and A.G. Evans, "Demonstration of the combined use of GPS pseudorange and Doppler measurements for improved dynamic positioning," in *Proceedings of 4th Int. Geodetic Symposium Satellite Positioning*, vol. 2, Austin, TX, April-May 1986, pp. 1397–1409.

[14] P.Y.C. Hwang and R.G. Brown, "GPS navigation: combining pseudorange with continuous carrier phase using a Kalman filter," *Journal of The Institute of Navigation*, vol. 37, no. 2, pp. 181–196, 1990.

[15] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System Theory and practice*. Viena: Springer, 2001.

[16] F. Nunes, J. Leitao, and F. Sousa, "Nonlinear filtering in gnss pseudorange dynamics estimation combining code delay and carrier phase," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 3, no. 4, pp. 639–650, Aug 2009.

[17] B. Parkinson and J. Spilker, Eds., *Global Positioning System: Theory and Applications, Vol. I*. Washington, D.C.: American Institute of Aeronautics and Astronautics, 1996.

[18] M. Bahrami, "Getting Back on the Sidewalk: Doppler-Aided Autonomous Positioning with Single-Frequency Mass Market Receivers in Urban Areas," in *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2009)*, Savannah, GA, September 2009, pp. 1716–1725.

[19] D. G. HULL, J. L. SPEYER, and M. GREENWELL, "Adaptive noise estimation for homing missiles," *Journal of Guidance, Control, and Dynamics*, vol. 7, no. 3, pp. 322–328, 1984.

[20] X. Li and V. Jilkov, "Survey of maneuvering target tracking. part i. dynamic models," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, no. 4, pp. 1333–1364, Oct 2003.