Comparison of Vector-Based Software Receiver Implementations with Application to Ultra-Tight GPS/INS Integration

M.G. Petovello and G. Lachapelle
Position, Location and Navigation (PLAN) Group
Department of Geomatics Engineering
Schulich School of Engineering
University of Calgary

BIOGRAPHY
Dr. Mark Petovello is a senior research engineer in the Position, Location and Navigation (PLAN) group where he executes and supervises various navigation-related projects. Since 1998, he has worked on several projects including satellite-based navigation, inertial navigation, reliability analysis, dead-reckoning sensor integration, and most recently, software-based GNSS receivers.

Dr. Gérard Lachapelle is a professor and CRC/iCORE Chair in Wireless Location in the above department, where he heads the PLAN group. He has been involved with GPS developments and application since 1980. For more information visit http://plan.geomatics.ucalgary.ca.

ABSTRACT
This paper investigates three different Kalman filter implementation options to estimate signal tracking errors in a GPS receiver with particular emphasis given to the carrier phase. The three options are outlined in terms of the system and measurement models. Initial assessment of the different options is performed in scalar-tracking mode with strong signals to assess their relative carrier phase tracking performance. Each option is then evaluated in vector-tracking mode both with strong signals collected in the field and with attenuated signals collected with a hardware GPS simulator. Results indicate that only two of the three options are able to track incoming signals in vector-tracking mode. Furthermore, both options are shown to be able to track the carrier phase up to an attenuation of about 15 dB and still maintain a velocity solution accurate to a few centimeters per second.

INTRODUCTION
Recent interest in software-based GNSS receivers has sparked many new and interesting application ideas. Included is the ultra-tight (or deep) integration of a receiver with an INS to improve overall tracking performance, primarily in degraded signal environments. The basic concept is to use the INS to measure user motion which is then fed into the receiver tracking loops to effectively limit the amount of dynamics “seen” by the receiver. In turn, the tracking performance can be improved since the tracking loop parameters can be optimized beyond the GPS-only case.

Several different ultra-tight GPS/INS approaches have been proposed, but most involve coupling the tracking of the different satellite signals via the antenna’s position and velocity. This approach is generally referred to as vector-based tracking (Spilker 1994). Typically, this is implemented using a Kalman filter to estimate the relevant system states (including position and velocity) and then to compute line-of-sight information for the individual tracking loops.

However, vector-based tracking techniques are extremely useful in their own right because of their ability to mitigate noise (Spilker 1994) which should improve signal tracking under weak-signal conditions, even without integration with an INS. The obvious benefit is the cost and power savings associated with the absence of the INS. As such, it is worthwhile to investigate GPS-only vector-based receiver strategies. Once the vector-based approach is well understood, the transition to an ultra-tight GPS/INS system is relatively simple since it mostly involves augmenting the navigation Kalman filter with appropriate INS-specific states; a well understood process (see e.g., Petovello et al 2003).

Given the above, this paper investigates various methods of implementing a vector-based receiver. With specific consideration for carrier phase tracking, three different algorithms are considered, each with a different system and/or measurement model. Each model is implemented in a software receiver developed at the University of Calgary.
The performance of the different algorithms is compared using field data as well as data collected using a GPS hardware simulator. Attention is given primarily to carrier phase tracking performance, since this has the most stringent tracking requirements, although position and velocity accuracy results are also presented. The primary objective of the paper is to determine if any of the algorithms yield sufficiently better carrier phase tracking performance than the others.

The paper begins with a review of vector-tracking methodology. Three different implementation options are then presented, including the relevant system and measurement models. Performance of the three algorithms is initially assessed using field data collected in an open-sky environment. Following this, data collected with a hardware GPS simulator is used to assess the algorithms.

**METHODOLOGY**

The fundamental objective of a GPS receiver during signal tracking is to generate a local signal that matches the incoming signal as closely as possible. Traditional GPS receivers track each satellite independently; an approach herein referred to as “scalar-tracking”. The objective of scalar-tracking is to estimate, on a satellite-by-satellite basis, the code phase and carrier frequency (and optionally carrier phase) of the incoming signals. These estimates, or more precisely, the errors thereof, are then used in feedback loop to drive the local signal generation.

A basic scalar-tracking architecture is shown in Figure 1. Down-converted and filtered samples are passed to each channel in parallel. The samples are then passed to a signal processing function where Doppler removal and correlation is performed. The correlator outputs are then passed to an error determination function consisting of discriminators (typically one for code, frequency and phase) and loop filters. Finally, the filtered error estimates are passed to the local signal generator, whose output is used during Doppler removal and correlation. As necessary, each channels’ measurements are incorporated into the navigation filter to estimate position, velocity and time parameters.

The benefits of scalar-tracking are the relative ease of implementation and a level of robustness that is gained by not having one tracking channel corrupt another tracking channel. However, on the downside, the fact that the signals are inherently related via the receiver’s position and velocity is completely ignored. Furthermore, the possibility for one tracking channel to aid another channels is impossible. For more information on scalar-tracking, please refer to Ward et al. (2006), Misra and Enge (2001), Van Dierendonck (1995) or Spilker (1994).

In contrast to scalar-tracking, vector-tracking estimates the position and velocity of the receiver directly. A basic vector-tracking architecture is shown in Figure 2. As can be seen, the individual tracking loops are eliminated and are effectively replaced by the navigation filter. With the position and velocity of the receiver known, the feedback to the local signal generators is obtained from the computed range and range rate to each satellite.

The primary advantages of vector-tracking are (Spilker 1994); noise is reduced in all channels making them less likely to enter the non-linear tracking regions; it can operate with momentary blockage of one or more satellites; and it can be better optimized than scalar-tracking approaches. Vector-tracking is also able to improve tracking in weak-signal or jamming environments, especially when integrated with inertial sensors (e.g., Ohlmeyer 2006; Pany and Eissfeller 2006; Gustafson et al. 2000). The primary drawback is that all satellites are intimately related, and any error in one channel can potentially adversely affect other channels.

Extension of vector-tracking to ultra-tight (or deep) integration with an inertial navigation system (INS) is possible by augmenting the architecture in Figure 2 with an inertial measurement unit (IMU) and replacing the navigation filter with an integrated GPS/INS filter. For this reason, vector-tracking is a natural first step to ultra-tight GPS/INS integration.
The concept of vector-tracking or ultra-tight GPS/INS integration has been investigated by several authors including Landis et al. (2006), Ohlmeyer (2006), Pany and Eissfeller (2006), Pany et al. (2005), Jovancevic et al. (2004), Soloviev et al. (2004a; 2004b), Abbott and Lillo (2003) and Kim et al. (2003), to name a few. For efficiency, many approaches (Ohlmeyer 2006; Kim et al. 2003; Jovancevic et al. 2004; Abbott and Lillo 2003) use a cascaded approach, as shown in Figure 3. In this case, each channel has an associated local filter that estimates the tracking errors for that channel. The advantage of this is two fold. First, depending on the implementation of the navigation filter, it can reduce the order of the navigation filter state vector. Second, the output from the local filter can be sent to the navigation filter at a lower rate, thus improving efficiency. It is noted that in Figure 3, feedback can optionally be made directly from the local filter to the local signal generator. The reason for this will be discussed later in the paper.

This paper investigates different system and measurement models for the local filter and their impact in terms of carrier phase tracking. To this end, several investigations into Kalman filter-based tracking loop estimation (for GPS-only or GPS/INS systems) has been conducted by Ziedan and Garrison (2004), Jee et al. (2003), Psiaki and Jung (2002) and Psiaki (2001), in addition to those references listed above. In these works, different system and measurement models have been proposed, but have never been compared or contrasted, nor have they been assessed in terms of carrier phase tracking capability, which is the main contribution of this paper.

Herein, three different local filter implementations are considered. These are discussed in more detail in the sub-sections below. However, before discussing these in detail, first consider the form of the correlator outputs. Assuming a known correlator offset of $\Delta$ (e.g., for early or late correlators), the in-phase ($I$) and quadra-phase ($Q$) correlator outputs are given by

$$I = A \cdot N \cdot R (\delta \tau - \Delta) \cdot \frac{\sin(\pi \cdot \delta f \cdot T)}{\pi \cdot \delta f \cdot T} \cdot \cos(\overline{\delta \phi})$$

(1)

$$Q = A \cdot N \cdot R (\delta \tau - \Delta) \cdot \frac{\sin(\pi \cdot \delta f \cdot T)}{\pi \cdot \delta f \cdot T} \cdot \sin(\overline{\delta \phi})$$

(2)

where $A$ is the signal amplitude; $N$ is the number of samples accumulated in the correlator; $R$ is the auto-correlation function of the ranging code; $\delta \tau$ is the error in the local code phase; $\delta f$ is the error in the local carrier frequency; $T$ is the coherent integration time interval; and $\overline{\delta \phi}$ is the average local phase error over the integration interval. Finally, the average phase error can be further expanded as (similar to Psiaki 2001; Psiaki and Jung 2002)

$$\overline{\delta \phi} = \delta \phi_0 + \delta f_0 \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6}$$

(3)

where a subscript of zero indicates a value at the start of the integration interval; and $\delta a$ is the frequency rate error (phase acceleration error).
Now that the output of the correlators has been defined, a
description of the different filter implementations is
presented.

**Option #1**
The first option follows closely the filter implementation
proposed in Psiaki (2001) and Psiaki and Jung (2002).  In
this case the correlator outputs are used directly in a
Kalman filter to estimate the amplitude, code phase error,
initial carrier phase error, initial carrier frequency error
and initial frequency rate error.  (In this context, “initial”
refers to the beginning of the most recent integration
interval.)  In fact, the only difference from *ibid.* is that
here we estimate the frequency and frequency rate errors
instead of their “real” value.  The covariance matrix of the
observations is computed as a function of the carrier-to-
noise density ratio, $C / N_0$, as (Van Dierendonck 1995)

$$
\sigma^2_{n_i} = \sigma^2_{n_q} = \frac{1}{2 \cdot 10^{0.1 C / N_0} \cdot T}
$$

where $n_i$ and $n_q$ represent the noise on the $I$ and $Q$
channels respectively.  It is noted that in theory, any
number of correlators (early, prompt and/or late) could be
used for updating the filter, although two to three
correlators is common in most receivers.

The system model for this implementation is written as
follows

$$
\begin{bmatrix}
\frac{d}{dt} A \\
\frac{d}{dt} \delta \tau \\
\frac{d}{dt} \delta \phi_0 \\
\frac{d}{dt} \delta f_0 \\
\frac{d}{dt} \delta a_0
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \beta \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
A \\
\delta \tau \\
\delta \phi_0 \\
\delta f_0 \\
\delta a_0
\end{bmatrix}
+ \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 1 & \alpha & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
w_A \\
w_{\delta \tau} \\
w_{\delta \phi_0} \\
w_{\delta f_0} \\
w_{\delta a_0}
\end{bmatrix}
$$

where $\beta$ converts units of radians per second into units of
chips per second; $w_A$ is the process noise for the
amplitude; $w_{\delta \phi_0}$ is the process noise for the code phase
error to account for code multipath effects; $w_{\delta f_0}$ is
process noise for the clock bias; $w_{\delta a_0}$ is the process noise
for the clock drift; and $w_{\delta f_0}$ is the process noise for the
phase acceleration (which is related to the receiver
dynamics).  Essentially, the model uses the carrier
frequency and frequency rate errors to propagate the
carrier phase and code phase.  Clock errors are
considered using a model similar to *ibid.* and Brown and
Hwang (1992).  The code phase error is also given a
random walk component to account for multipath effects.
The process noise value for the amplitude will be
application-specific and must account for the expected
level of signal variation.

It is noted that the $\sin(x) / x$ term in equations (1) and (2)
is effectively combined with the amplitude term.  The
reason for this is that the attenuation due to frequency
error is very difficult to separate from variations in the
amplitude.  Therefore, the frequency error is only
estimated from the average phase error “component” in
equations (1) and (2).

**Option #2**
The second option has the same system model as Option
#1 (see equation (5)), but differs in terms of the
measurement model.  This case, the $I$ and $Q$ values are
combined in two ways.  First, their root sum square is
computed

$$
Z_1 = \sqrt{I^2 + Q^2}
$$

$$
= A \cdot N \cdot R(\delta \tau - \Delta) \cdot \sin(\pi \cdot \delta f \cdot T) / \pi \cdot \delta f \cdot T
\quad (6)
$$

where $B$ is the combination of the amplitude and $\sin(x) / x$
terms, as discussed at the end of the last
sub-section.  This observation is used to observe the
amplitude and code errors.

Second, the average phase error is extracted using an
inverse tangent operation

$$
Z_2 = \overline{\delta \phi} = \tan^{-1}\left(\frac{Q}{I}\right)
\quad (7)
$$

This observation is used to estimate the carrier phase error
and its derivatives.

It is noted that the variance of the above two observations
must be computed from the variance of the $I$ and $Q$
values given in equation (4).

Effectively, equations (6) and (7) approximate what
typically happens in a scalar-tracking receiver.
Specifically, the code and phase errors are separated in
their respective discriminators.  The difference in this
case, is that instead of passing these values into a digital
loop filter, they are used as observations to a Kalman
filter.

**Option #3**
The third option attempts to link the code and phase error
via a range error and an ionospheric effect, as proposed
by Abbott and Lillo (2003).  In particular
\[
\delta \tau = -(\delta \rho + 1) \frac{f_{\text{chip}}}{c} \tag{8}
\]
\[
\delta \phi = (\delta \rho - 1) \frac{2\pi}{\lambda} \tag{9}
\]

where \(\delta \rho\) is the range error; \(I\) is the ionospheric effect; \(f_{\text{chip}}\) is the ranging code chipping rate (1.023 MHz for GPS C/A code); \(c\) is the speed of light; and \(\lambda\) is the carrier wavelength.

For the measurement model, equations (8) and (9) are substituted into equations (1) and (2). At the same time, the initial frequency error and initial frequency rate error are replaced by initial range rate and range acceleration error terms (scaled to units of radians). In this way, the measurement model is similar to that of Option #1 since the outputs from the correlators are used to directly update the filter. The difference from Option #1 lies in the actual states that are being estimated (see below).

Since the state vector differs from the previous two options, the system model must also differ. In this case, the system model can be written as

\[
\begin{bmatrix}
    \dot{\delta \rho}_o \\
    \dot{\delta \phi}_o \\
    \dot{b}
\end{bmatrix} =
\begin{bmatrix}
    A & 0 & 0 & 0 & 0 \\
    0 & 0 & 1 & 0 & 0 \\
    0 & 0 & 0 & 1 & 0 \\
    0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
    \delta \rho_o \\
    \delta \phi_o \\
    b
\end{bmatrix} +
\begin{bmatrix}
    1 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0 \\
    0 & 0 & 1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
    w_{\text{d}} \\
    w_{\text{accel}} \\
    w_{\text{f}}
\end{bmatrix} \tag{10}
\]

where a “dot” above a parameter indicates its time derivative; and \(w_{f}\) is the process noise for the ionosphere error. It is noted that the process noise that accounts for the acceleration (i.e., \(w_{\text{accel}}\)) is the same as in equation (5) but is scaled to the appropriate units. Essentially, this model is driven by the range dynamics alone, with no consideration for clock errors. Both the range acceleration error and ionosphere error are assumed to be random walk.

**Navigation Filter**

Finally, the navigation filter, which is common for all three options, is assumed to have the following system model

\[
\begin{bmatrix}
    \dot{\bar{p}}_x \\
    \dot{\bar{v}}_x \\
    \dot{d}_x
\end{bmatrix} =
\begin{bmatrix}
    0 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
    \bar{p}_x \\
    \bar{v}_x \\
    d_x
\end{bmatrix} +
\begin{bmatrix}
    0 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
    \bar{w}_{\text{clock}} \\
    \bar{w}_{\text{dop}}
\end{bmatrix} \tag{11}
\]

where subscripts indicate the dimension of the quantity; \(\bar{p}\) is the position vector; \(\bar{v}\) is the velocity vector; \(b\) is the clock bias; \(d\) is the clock drift; \(\theta\) is a zero matrix; \(I\) is the identity matrix; and \(\bar{w}_{\text{dop}}\) is the process noise vector for the velocity states. The model assumes the velocity is random walk, an assumption that is valid for the data sets analyzed herein. However, it is noted that other applications may need to select a more appropriate system model.

The measurements used to update the navigation filter are the pseudorange and carrier Doppler for each satellite. This approach was selected so that it is common amongst all three filter implementation options. In a final implementation, the error estimates output from the local filters could be used to directly update the navigation filter. The covariance of the observations is obtained as a natural by-product of the channel filters.

It is noted that the carrier phase data is not used to update the navigation filter. The reason for that is because in order to track the carrier phase using the position, the position would have to be known to the centimeter level. However, in single-point operation, this level of accuracy is not attainable. Instead, the estimated carrier phase errors are passed directly to the carrier generator to correct the current phase value. The navigation filter is then used to estimate the carrier Doppler that is needed to propagate the corrected phase value forward in time.

**TEST DESCRIPTION**

To evaluate the various filter implementations, two sets of data were used. First, field data collected in an open-sky environment was used to assess the performance of the algorithms under ideal, but real, situations. Second, data was collected using a GPS hardware simulator in order to control the received signal power and assess performance of the algorithms under weaker signal conditions.

**Field Test Description**

The field test data was collected on the roof of the CCIT building at the University of Calgary on July 25, 2005. Approximately one minute of data was collected, and seven satellites were visible during the test. This data set has been used extensively for evaluating other software
receiver implementations at the University of Calgary. It therefore provides an opportunity to evaluate the performance of the different filter implementations using real field data under ideal situations.

Simulator Test Description
The second data set was collected using a Spirent 7700 GPS hardware simulator. Although the simulator is capable of outputting L1, L2C and L5 signals, only the L1 data was simulated in this case. The location of the test was simulated to be in Calgary, Canada with the vehicle moving in an easterly direction with a constant speed of 5 m/s. A total of nine satellites were simulated. The ionosphere was simulated using the Klobuchar model (Klobuchar 1996).

The power of the simulated signals was varied as shown in Figure 4, relative to -157 dBW. After an initial period with full signal power, the power of all satellite signals was dropped at a rate of 0.5 dB/s for one minute. By decreasing the signal power, any differences in terms of the tracking sensitivity of the three filter algorithms can be identified.

**Figure 4 - Simulated Signal Power on All Satellites Relative to -157 dBW**

Data Collection System
The data collection setup is shown in Figure 5. The signal from the antenna is passed to a NovAtel Euro-3M card that acts as the front-end to the software receiver (the NovAtel card can also track the signals and output measurements, but this information was not used in this case). Samples (3-bits for L1 and 3-bits for L2) are output at a rate of 40 MHz. These samples are then repackaged into a more compact format using an FPGA card before being passed to a data acquisition card that resides on a PC. The data acquisition card then writes the samples to file for later processing. For the testing performed here, only the L1 signals are logged.

**Figure 5 - Data Collection System**

FIELD TEST RESULTS
As a first step to assessing the performance of the different local filter implementations, the data was processed in scalar-tracking mode, but using the local filters to close the loops instead of the traditional discriminator and loop filter approach. This approach allows for an assessment of the local filter operation without any influence due to the navigation filter.

The figure of merit used to assess the algorithm performance is termed the “phase estimation error” given by

\[
\text{Phase estimation error} = \widehat{\delta \varphi} - \tan^{-1}\left(\frac{Q}{I}\right)
\] (12)

where \(\widehat{\delta \varphi}\) is the estimated average phase error (i.e., the initial estimated phase error propagated to the middle of the integration interval). The second term is the true
average phase error as computed from the correlator output. Although this value is corrupted by noise, it is ultimately this value that the local filters are trying to estimate.

Figure 6 shows the carrier phase estimation error for Option #1 scaled to units of length (for L1). The different colors represent different satellites. As can be seen, the errors are at the sub-millimeter level (1σ), which is in line with traditional scalar-tracking methods. The root mean square (RMS) error across all satellites is 0.4 mm. This suggests the algorithm is accurately tracking the carrier phase of the incoming signal.

Similar results are obtained for Option #2 but are not shown here in the interest of space. The RMS errors for Option #2 is 0.5 mm.

For Option #3, carrier phase estimation errors were found to be highly sensitive to the ionosphere estimate. To illustrate this, Figure 7 shows the carrier phase estimation errors for the case when the ionosphere state is constrained to zero (this is equivalent to removing the state from the system altogether). As can be seen, the tracking performance is comparable to Options #1 and #2, and the RMS error is 0.6 mm.

However, if the ionosphere state is allowed to vary from zero, the results degrade rapidly. For example, if the initial standard deviation of the ionosphere state is set to 1 cm and the process noise is set to 1 cm²/s³, the phase estimation errors degrade to those shown in Figure 8. This is a marked degradation relative to the first two approaches. The likely reason for this behavior is that the ionosphere error is not easily observable using the data available. Perhaps such a model would be more useful if tracking signals on two frequencies, instead of just on L1 as is the case here. Regardless, using a constrained ionosphere has shown good results and as such, this model will be used in further testing.

For Option #3, carrier phase estimation errors were found to be highly sensitive to the ionosphere estimate. To illustrate this, Figure 7 shows the carrier phase estimation errors for the case when the ionosphere state is constrained to zero (this is equivalent to removing the state from the system altogether). As can be seen, the tracking performance is comparable to Options #1 and #2, and the RMS error is 0.6 mm.

However, if the ionosphere state is allowed to vary from zero, the results degrade rapidly. For example, if the initial standard deviation of the ionosphere state is set to 1 cm and the process noise is set to 1 cm²/s³, the phase estimation errors degrade to those shown in Figure 8. This is a marked degradation relative to the first two approaches. The likely reason for this behavior is that the ionosphere error is not easily observable using the data available. Perhaps such a model would be more useful if tracking signals on two frequencies, instead of just on L1 as is the case here. Regardless, using a constrained ionosphere has shown good results and as such, this model will be used in further testing.

Having confirmed that the three filter options can (in certain configurations) accurately track the carrier phase of the incoming signal in scalar-tracking mode, testing proceeded to full vector-tracking mode. Unfortunately, Option #3 was unable to track any satellite after only a few milliseconds. As such, Option #3 is deemed inappropriate to track L1-only carrier phase data in vector-tracking mode, and will not be considered further in this paper. However, Option #1 and #2 both performed well and were able to track the carrier phase of all seven satellites for the duration of the data set. The RMS carrier phase estimation error, position error and velocity using these two options are summarized in Table 1. The position and velocity errors are computed relative to the known coordinates and velocity (static) of the antenna.
Table 1 - RMS Phase Estimation Errors, Position Errors and Velocity Errors for Option #1 and #2 in Full Vector-Tracking Mode During Field Test

<table>
<thead>
<tr>
<th>Option</th>
<th>Phase Estimation Error</th>
<th>Position Error</th>
<th>Velocity Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3 mm</td>
<td>4.6 m</td>
<td>3.2 cm/s</td>
</tr>
<tr>
<td>2</td>
<td>0.5 mm</td>
<td>5.5 m</td>
<td>3.5 cm/s</td>
</tr>
</tbody>
</table>

As can be seen, the positioning and velocity results are commensurate with those obtained using standard receivers operating in single-point mode. As such, initial results suggest proper operation of the receiver under ideal signal conditions.

SIMULATOR TEST RESULTS

Given the results of the previous section, the objective of the simulator test is to assess the performance of Option #1 and #2 in the presence of weaker signals. To this end, Figure 9 shows the carrier phase estimation errors for Option #1 along with the estimated clock drift. As can be seen, the carrier phase estimation errors show a very periodic behavior that is highly correlated with the estimated clock drift.

Figure 9 - Vector-Tracking L1 Carrier Phase Estimation Errors for All Satellites Using Option #1 During Simulator Test (each satellite is a different color), and Estimated Receiver Clock Drift

The exact reason for this periodic behavior is unknown, but it is likely due to the test setup. In particular, during the simulator test, the NovAtel Euro-3M card was using its internal oscillator with clock steering enabled. However, for the field data, it was using an external Rubidium oscillator with clock steering disabled. It is suspected that the periodic behavior of the clock drift in Figure 9 is due to the steering of the receiver’s internal oscillator. Regardless of its source, the actual clock behavior is not well approximated by a random walk model, as is used in the local filters and the navigation filter. The result is that the filters are not well suited to track this type of clock drift and is likely negatively impacting the tracking performance because the un-modeled effects will be “seen” as extra dynamics by the receiver, thus complicating the tracking process.

Nevertheless, the carrier phase estimation errors are still on the order of a few millimeters throughout the test. More interesting, however, is that the estimation error does not appear to vary as the signal power is decreased, at least until the signal is lost altogether. These results indicate Option #1 is able to track the carrier phase of the signal even in the presence of decreasing signal power.

Results for Option #2 are similar to those of Option #1 and are therefore not shown explicitly. The only difference is that only six of the nine satellites could be tracked with Option #2. Regardless, the tracking performance of those six satellites is deemed satisfactory. It is noted that tuning of the local filter for Option #2 may allow all nine satellites to be tracked, although this was beyond the scope of this paper.

A summary of the results for Option #1 and #2 is given in Table 2. The maximum attenuation is the level of attenuation at which the last satellite lost lock. Overall, both options were able to track the signal down to approximately 15 dB worth of attenuation, which is a sensitivity improvement of about 7 dB over scalar-tracking methods obtained the same data set (not shown). Although Option #1 appears to yield slightly better performance, the differences are considered minimal, especially since only one test was conducted.

Table 2 - Summary of Simulator Test Results for Option #1 and #2

<table>
<thead>
<tr>
<th>Option</th>
<th>Maximum Attenuation</th>
<th>RMS Velocity Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Horizontal</td>
</tr>
<tr>
<td>1</td>
<td>15 dB</td>
<td>1.4 cm/s</td>
</tr>
<tr>
<td>2</td>
<td>16 dB</td>
<td>3.2 cm/s</td>
</tr>
</tbody>
</table>

CONCLUSIONS AND FUTURE WORK

This paper investigated three local filter implementation options for estimating individual channel errors. All three algorithms were shown to be capable (using certain configurations) of tracking the carrier phase in scalar-tracking mode. However, once vector-tracking was enabled, only two of the three methods were able to track signals for more than a few milliseconds.

Both of the options that worked in vector-tracking mode were able to estimate the phase error with millimeter level accuracy (at L1) suggesting they could be used for RTK applications. Both methods were also able to track the carrier phase up to a maximum signal attenuation of about...
15 dB, which is a significant improvement over scalar-tracking approaches.

Future work will proceed on three fronts. First, more testing will be done to determine the cause of the periodic clock drift behavior observed in the simulator test since this will likely have a beneficial impact on tracking ability. Second, more testing will be performed on the two filter options that were able to work in vector-tracking mode. This testing will allow for a more thorough evaluation of the algorithms. Finally, once a local filter implementation is chosen, an IMU will be integrated into the receiver and the navigation filter will be replaced with an equivalent GPS/INS filter. Integration with the IMU is expected to yield improved tracking capability.

ACKNOWLEDGMENTS
The authors would like to acknowledge the Informatics Circle of Research Excellence (iCORE) for funding part of this research.

REFERENCES


