Abstract—In weak GNSS signal environments, extending integration time is paramount to improving the GNSS receiver’s sensitivity. Furthermore, sufficient coherent integration can help to separate the line-of-sight (LOS) and non LOS (NLOS) signals—primarily in the Doppler domain—for multipath mitigation. However, extending integration time is limited by the presence of the navigation message data bits. The Maximum-Likelihood (ML) estimation method has been shown as the most effective way to estimate the navigation bit boundary locations (i.e., bit synchronization) and subsequently estimate the data bit values (i.e., bit decoding) in the presence of noise alone. This paper further analyzes the performance of ML estimation method as a function of various other parameters by using the successful synchronization rate (SSR) and successful decoding rate (SDR) as the criteria. The parameters considered include the number of data bits required (i.e., integration time) and the Doppler frequency error, and both are evaluated under different signal strength scenarios. The requirements for bit synchronization are analyzed under three SSRs, which are 85%, 90% and 95%. Results indicate that, under the SSR of 90% and the SDR of 90%, bit synchronization and bit decoding are valid for signal strengths of 15 dB-Hz and 20 dB-Hz respectively, and the Doppler frequency errors should not be larger than 24 Hz and 11 Hz respectively.

Keywords- GNSS (GPS) receiver; standalone; weak signal; high sensitivity; extended integration; bit synchronization; bit decoding

I. INTRODUCTION

Global Navigation Satellite Systems (GNSS) such as the Global Positioning System (GPS) are widely used in modern positioning and navigation fields. They are vital for applications such as aircraft auto-piloting, automobile en-route guidance, pedestrian positioning, etc. Recently, because of the increased demand for navigation in indoors, under dense foliage canopies and in urban canyons, processing weak GNSS signal has been receiving growing attention.

High-sensitivity GPS (HSGPS) receivers are capable of providing satellite measurements for signals attenuated by up to approximately 30 dB [1-3]. For HSGPS receivers, extending integration is paramount to obtaining higher sensitivity but current signals are limited by the presence of the navigation message data bits. For the GPS L1 C/A signal, the navigation data are binary phase-shift keying (BPSK) modulated onto the carrier phase with a bit duration of 20 ms. For coherent integration beyond 20 ms, the navigation data bit wipe-off is required to avoid energy loss that occurs due to bit transitions. Furthermore, complete bit wipe-off does require the knowledge about bit boundaries and bit values.

By using the navigation data bit aiding and frequency aiding from an external source, [4] showed that in acquisition stage signals with carrier to noise-density ratios (C/N0) of 32, 22, 17, and 12 dB-Hz can be detected requiring coherent integration time of at least 8, 200, 400, and 800 ms respectively. Similarly [5-7] used aiding information from wireless network broadcasting, and [8] mentioned that network assistance can be used for bit synchronization by providing time and position information. However, all of these methods need access to external aiding sources, which may not be applicable in all circumstances, and the system is more complex and expensive than a stand-alone GNSS receiver infrastructure.

This paper looks at the stand-alone receiver case only and uses Maximum-Likelihood (ML) estimation algorithms to first perform bit synchronization [9] (i.e., determine the location of the data bit boundaries within the data stream) and subsequently bit decoding [10] (i.e., estimate the value of the data bits themselves). With the bit synchronization obtained and the ability to estimate the bit values, the integration time can be extended coherently by data wipe-off.

The main objective of this paper is to investigate the performance of ML bit synchronization and bit decoding algorithms in terms of the number of data bits required (for bit synchronization) and the tolerance to Doppler/frequency error; both as a function of the received signal strength. Since these three factors are mutually coupled the analysis is divided into three stages. First, given a signal strength and Doppler frequency error level, what is the necessary number of data bits required? Second, given an available number of data bits and
Doppler frequency error level, what received signal power can be used? Finally, for a given signal strength and some number of available data bits, what is the maximum Doppler frequency error that can be tolerated?

In the context of this work, performance is assessed in terms of the achievable synchronization rate (SSR) of bit boundaries and the successful decoding rate (SDR) of bit values. For bit synchronization, the above three parameters are assessed in terms of attaining one of two SSR values, namely 85%, 90% and 95%. Receiver performance under different SSR values considers the potential accuracy of the signal Doppler. More specifically, since large bit synchronization errors can result in a split/double correlation peak in the frequency domain, the required SSR values needed to avoid this situation are assessed.

The contributions of this work are two-fold. First, it assesses the performance of ML bit synchronization and decoding as a function of the number of data bits, the effect of Doppler error and received signal power. This contrasts with previous work that focused primarily on effects of noise only. Second, the ML bit synchronization and decoding is assessed in light of the ability of the receiver to properly estimate the signal’s Doppler frequency, that is, its ability to avoid dual peaks in the Doppler correlation function.

The paper is organized as follows. The Section II Methodology and Section III Test Description summarize the ML estimation algorithms of bit boundaries and bit values and the relevant test-bench scheme. The Section IV Test Results gives the simulation results and requirements for extending integration. The Section V summarizes the test outcomes.

Before moving on, it is noted that the algorithm is presented in the context of the GPS L1 C/A-code signal; however, the basic algorithms are applicable to any binary phase-shift keying (BPSK) modulated GNSS signal.

II. METHODOLOGY

This section gives a brief overview of the methodology used in this paper.

A. ML Bit Synchronization

The ML bit synchronization algorithm has been shown as the best approach [9] to estimate bit boundaries for a stand-alone GNSS receiver in challenged environments. The conventional Histogram method [11] counts sign changes of the prompt correlator output at various locations within one data bit interval and compares these counts with predefined thresholds. However, the Histogram method suffers in weak signal conditions because the bit error rate is excessive.

The ML bit synchronization is a process to detect bit boundary locations by using a likelihood function. Since the period of the GPS L1 C/A code is 1 ms and are synchronous with the 20-ms long data bits, there are 20 possible bit locations. The likelihood function used is the energy sum of cross-correlation functions between the prompt correlator output sequence and a 20 ms window function. The concept behind ML bit synchronization is that one bit transit can be detected by a designed match filter in open-sky, and more than one bit transits at the same bit boundary can help to improve the SSR in weak signal environments.

For GPS L1 C/A signal, the k-th ms prompt correlator output of an N data bit sequence is given by [12]

\[ I_k(\Delta \tau, \Delta f, t_{ran}) = R(\Delta \tau)B_k(t_{ran}) \exp(j\pi \Delta f T_{co} + j\Delta \theta_i) \text{sinc}(\pi \Delta f T_{co}), \]

\[ (k = 1, 2, ..., 20 \times N) \]

where \( R(\Delta \tau) \) is the ranging code (i.e., C/A code) correlation function, \( \Delta \tau \) is the error in the locally generated ranging code, \( B_k(t_{ran}) \) is the navigation data bit value with transit at \( t_{ran} \) ms (\( t_{ran} \) is between 1 and 20 ms), \( \Delta f \) is the Doppler/frequency error, \( T_{co} \) is the coherent integration time equal to 1 ms in this paper, \( \Delta \theta_i \) is the initial phase difference between incoming signal and locally generated carrier, and the \( \text{sinc} \) function represents the carrier spectrum after code wiping-off.

Using equation (1) as a starting point, the ML bit synchronization algorithm in [9] is summarized below. A designed match filter, that here is realized by a 20 ms width window function, is defined as

\[ W_j = 1, (i = 1, 2, ..., 20) \]

The match filter output which is the cross-correlation between \( I_k(\Delta \tau, \Delta f, t_{ran}) \) and \( W_j \) is given by

\[ CR_j(\Delta \tau, \Delta f, t_{ran}) \]

\[ = \sum_{i=0}^{20} I_{i,j}(\Delta \tau, \Delta f, t_{ran}) W_i, \ (j = 1, 2, ..., 20/(N+1)) \]

Assuming a constant error (within one chip so as to contain the peak of the auto-correlation function) in the locally generated ranging code (i.e., \( \Delta \tau = \text{const} \)), the squared cross-correlation output is given by

\[ SCR_j(\Delta \tau = \text{const}, \Delta f, t_{ran}) \]

\[ = \sum_{n=0}^{N} CR_{2n+1,j}(\Delta \tau = \text{const}, \Delta f, t_{ran}), \ (i = 1, 2, ..., 20) \]

The ML estimate of bit boundaries (20 possibilities, indicated with \( i \) ranging from 1 to 20) can be found by selecting the value that maximizes the sum (over time) of the squared cross-correlation values from the previous step. The ML estimate of bit boundaries is obtained as

\[ t_{ran} = \arg \max_{t_{ran} \in [1:20]} SCR_j(I_k(t_{ran})), \]

\[ (i = 1, 2, ..., 20; k = 1, 2, ..., 20 \times N) \]

B. ML Bit Decoding Algorithm

ML bit decoding algorithm has also been shown as the most effective way for bit decoding [10]. There are several conventional methods to obtain bit values for a stand-alone GNSS receiver, including: take advantage of the periodic

The research presented in this paper was sponsored by General Motors of Canada, and Natural Sciences and Engineering Research Council of Canada.
property of the ephemeris and almanac data [13]; estimate bit values by using the Viterbi algorithm [14]; and estimate bit values by using the extended Kalman filter method [15]. The method of using message periodicity assumes that – for a standalone receiver – at least one complete set of navigation data has been received in open-sky before GNSS users entering the weak signal environment, which may not always be achieved. Reference [14] employed a sequential estimation routine that can decode navigation data on a sequential bit basis, but did not operate over the entire correlation interval. Reference [15] requires considerable calculation resources. In addition to those bit decoding algorithms, the integration time can be extended non-coherently after knowing bit boundaries, but this is not as efficient as coherent integration due to squaring loss. Reference [4] and [16] mention an algorithm using two consecutive 10 ms of data for integration can guarantee at least one of two sets data does not contain a bit transition. However, this method can only provide the coherent integration time up to 10 ms and will be invalid if the signal strength is lower than 32 dB-Hz [4, 10].

Bit decoding is a process to determine bit values after the bit synchronization has completed. The likelihood function used in the ML algorithm used herein is the inner product between prompt correlator output of 20 ms starting from bit boundaries and locally generated bit combinations. If trying to decode \( N \) bits at a time, the number of possible bit combinations is equal to \( 2^{N-1} \), and the correct bit combination is one element of a \( 2^{N-1} \) bit combinations.

\[ IP_{\Delta \tau, \Delta f_D}^n (\Delta \tau, \Delta f_D, \hat{B}) = \sum_{m=1}^{N} \left[ I_{20,\tau} (\Delta \tau, \Delta f_D) \cdot \hat{B}_{m,n} \right] (m = 1, 2^{N-1}) \]  

where \( \hat{B}_{m,n} \) is one element of a \( 2^{N-1} \times N \) matrix containing \( N \) bits for each combination and \( 2^{N-1} \) bit combinations.

Assuming a constant error (again, within one chip) in the locally generated ranging code (i.e., \( \Delta \tau = \text{const} \)), the energy of the inner product given by \( IP_{\Delta \tau}^n (\Delta \tau = \text{const}, \Delta f_D, \hat{B}) \), where \( \hat{B} \) is a bit batch (i.e., vector) contains \( N \) bits.

The ML estimate of bit values can be found by maximizing the energy of the inner product. The ML estimate of bit values is obtained as

\[ \hat{B} = \arg \max_{\hat{B} \in \{\pm 1, \ldots, \pm 1\}} IP_{\Delta \tau}^n (\Delta \tau, \Delta f_D, \hat{B}), (n = 1, 2, \ldots, 20) \]  

\[ (9) \]

C. The SSR and Its Validity

Coherent integration can be extended if the bit boundaries and bit values are known. However, the quality of ML estimate for bit synchronization and bit decoding is highly related to the GPS signal strength and the Doppler frequency errors. In weak signal environments, the validity of ML estimation is therefore hard to approximate theoretically. In this paper, the concept of SSR is used, which is the percentage (i.e., rate) of the time the bit synchronization process yields the correct bit boundary location for a given number of data bits, Doppler error and C/N0. However, since SSR will not be 100% in weak signal conditions, the required number of data bits, maximum Doppler error and minimum C/N0 are computed herein for SSR values of 85%, 90% and 95%.

In the case that there is a bit synchronization error, the “sinc function” of Doppler frequency spectrum (which will not be a real sinc function if there is a bit synchronization error) will look as shown in Figure 1. The figure has several plots according to different bit synchronization errors (BSE). For the purpose of extending integration time, the distortion from the bit synchronization error of \( n \) ms is actually same as \( 20 - n \) ms, so the figure only shows BSE up to 10 ms.

![Figure 1. The attenuation of bit synchronization errors as a function of frequency error](image)

It can be shown that the attenuation at central Doppler frequency is given by [17]
\[ \alpha_{b_t} = |1 - 2\delta h_t| \] (10)

where \( \delta h_t \) is the normalized bit transit error, ranging from 0 to 1, which represents the bit transit error from 0 to 10 ms. Furthermore, as shown in Figure 1, the peak of “sinc function” becomes split if the bit synchronization error exceeds 3 ms. This is an important conclusion, which can be used as the criterion to evaluate the validity of different SSRs. Specifically, since the goal of any GNSS tracking loop is ultimately to compute the code phase and Doppler as accurately as possible, it follows from the above that in order to do this any bit synchronization errors should be 3 ms or less. Later on, the effect of different SSR values on the ability to limit the synchronization errors to less than this amount will be assessed.

To do this, we introduce the concept of the ratio of non-split peaks in false results (RNPFR) and the ratio of non-split peaks in total results (RNPTR). The RNPFR means the ratio of the bit synchronization errors between 1 to 3 ms (i.e., non-split peak results) within all of the false results (i.e., results with incorrect bit synchronizations). In contrast, the RNPTR means the ratio of the BSE from 1 to 3 ms in all of the results (including both correct and false results).

A sample histogram of synchronization errors amongst all false results is shown in Figure 2. These results are from the simulation of a signal with \( C/N_0 = 30 \) dB-Hz and using 9 data bits for synchronization (details of the simulation are given below). The result show that the number of BSE between 1 to 3 ms is far more than all other errors combined, meaning non-split peaks are the most likely situation. Details about different RNPFRs and RNPTRs according to different SSRs can be accessed in Section IV Test Results.

\[ r_i = Ac_i + n_i \] (11)

where \( r_i \) is the received signal after correlation (assuming perfect phase tracking) with sampling rate 1 kHz and \( n_i \) is the additive white Gaussian noise. For ML bit synchronization, the correlators are generated at a rate of 1 kHz and for bit decoding they are generated at 50 Hz.

To include the effect of Doppler errors, the model is updated as follows

\[ r_i = Ac_i \exp(j\pi f_i T_\text{co}) + n_{i,\text{Q}} \] (12)

where \( T_\text{co} \) is 1 ms for bit synchronization and 20 ms for bit decoding, and \( n_{i,\text{Q}} \) is the complex form of additive white Gaussian noise.

The second data set used was from a live field test conducted in an open-sky environment and was used to verify whether the simulation results are reasonable. A weak signal environment (i.e., different \( C/N_0 \)) is then generated by adding white Gaussian noise to contaminate the open-sky data. The field test data was collected on the roof of the CCIT building at the University of Calgary on November 29, 2011. Approximately one minute of data was collected using a National Instruments PXI-5600 front-end with an oven controlled crystal oscillator (OCXO). Seven satellites were visible during the test.

The input to the ML bit synchronization algorithm was 1 ms prompt correlator outputs. The input to the ML bit decoding was 20 ms prompt correlator outputs. All the signal processing was initially performed in Matlab™ and has since been ported to C++.

IV. TEST RESULTS

In this section, the results of bit synchronization are presented first followed by the results of bit decoding.

A. ML Bit Synchronization Algorithm

1) Simulation Results

For bit synchronization, in order to investigate the relationship between the numbers of data bits and SSR, a weak signal scenario was considered in which \( C/N_0 = 20 \) dB-Hz. A Monte Carlo test consisting of 10,000 trials was implemented to obtain mean and standard deviation (SD) values of SSR when using different numbers of data bits. Specifically, the number of bits used included 2, 4, 8, 16, 32, 64, and 128. The data bit sequence used here is randomly generated, that is, each bit has an equally likely chance of being ±1. The impact of other simulated bit sequences is assessed later. The result in Figure 3 shows that the SSR increases with more navigation data bits. For this particular situation, more than 30 data bits are required to obtain a 90% SSR (at 20 dB-Hz).
In order to investigate the impact of different $C/N_0$ on the SSR, a simulation with different signal strengths and different numbers of data bits was implemented and the results are shown in Figure 4. The results indicate that in order to achieve an $SSR \geq 95\%$, the ML bit synchronization algorithm requires approximate 200 bits at $C/N_0 = 15 \text{ dB-Hz}$, and 50 bits at $C/N_0 = 20 \text{ dB-Hz}$, but only 15 bits at $C/N_0 = 25 \text{ dB-Hz}$.

It should be noted that more data bits cannot help to improve SSR if all the bits have the same value (i.e., lack of bit transitions). However, because the navigation message content can be considered as random values, more data bits can contribute by assuming bit transitions happen with the probability of $50\%$ for each incoming data bit. Two sets of bit streams are generated to assess this problem. One is composed by alternating $\pm 1$ and the other is a random sequence with the probability of $+1 = 50\%$. For a $90\%$ SSR, the result is shown in Figure 5. As expected, the alternating bit sequence gives better results because there are more bit transitions.

The reason to consider these two bit sequences is also shown in Figure 5. Specifically, for the alternating $\pm 1$ bit stream, a single bit is sufficient to obtain an SSR of $90\%$ for signals stronger than $35 \text{ dB-Hz}$, but in contrast, at least six bits are required for signals even as high as $40 \text{ dB-Hz}$ when a random bit sequence is used. To further the analysis, we now consider the number of data bits needed in order to obtain SSR values of $85\%$, $90\%$ and $95\%$. As before, Monte Carlo simulations are used with different assumed $C/N_0$ values. The number of required data bits as a function of $C/N_0$ is shown in Figure 6.
The RNPFr and RNPR results as a function of C/N₀ for different SSR values are shown in Figure 7. Results indicate that there is no significant difference among SSR values of 85%, 90% and 95%. In particular, the RNPR higher than 99.9% for all the cases meaning that the ML bit synchronization algorithm is quite reliable. In addition, in the rare instances that the bit synchronization is incorrect, the frequency spectrum only exhibits a split peak 10% of the time or less (worst case). This shows that if an SSR of 85% is obtainable the Doppler can be estimated correctly (i.e., without having to worry about a split in the spectrum) with a likelihood of at least 99.9%.

From a practical perspective, using a lower SSR offers the ability to use fewer navigation bits and also helps to reduce the computational load, but still provides satisfactory results in nearly all cases. By extension, using fewer bits helps to reduce the receiver’s sensitivity to frequency error introduced by user dynamics.

Of course, an SSR of 85% is not the same as an SSR of 95%. If the coherent integration gain is paramount in some applications, the synchronization error of 3 ms, which causes a 3 dB loss in energy, will not be acceptable. As such, different SSR values can be implemented in GNSS receivers for different applications/situations.

Previous analysis – both here and in the literature – has not considered the effect of Doppler error on bit synchronization. These errors can arise from biased carrier estimation, unstable local oscillators and/or user dynamics during integration. In the interest of space, only two representative tests are considered here; one for open-sky signals of C/N₀ = 40 dB-Hz and the other for weak signals of C/N₀ = 20 dB-Hz. The results are shown in Figure 8 and Figure 9 respectively.

The result shows that in order to maintain a certain SSR, as the Doppler error increases so should the number of data bit used. It is noted that the Doppler frequency errors are only generated to a maximum of 24 Hz because the ML synchronization algorithm will fail for a 25 Hz frequency error no matter how many data bits are used. In that case, the bit transitions cannot be distinguished from Doppler frequency errors, so the Doppler frequency error of 24 Hz is the bound of this algorithm. It is noted however, that if the Doppler frequency uncertainty is mainly from user’s dynamics, the inertial sensors can be included in the system to reduce the Doppler error using, for example, an ultra-tight receiver architecture.

2) Field Test Results
As mentioned above, a field test was also conducted to confirm the simulation results. The field test results for the required number of navigation data bits as a function of C/N₀ are shown in Figure 10, along with the simulation results taken
from Figure 5. Please note that the maximum L1 Doppler rate due to satellite motion is about 1 Hz/s [16] and correspondingly an integration time of 1 s (i.e., 50 bits) may have a Doppler uncertainty of 1 Hz. Practically, integrating for 1 s with a 1 Hz/s Doppler rate in a standalone receiver (as considered here) will result in such a large attenuation of power that the correlator outputs would be effectively useless. As such, requiring more than 50 data bits is not considered for the field test data. However, this will not weaken the purpose of the test, which is to confirm the correctness of simulation models.

The field test result is quite comparable to the simulation results. For 40 dB-Hz signal, just one bit is sufficient for bit synchronization because the first data bit contains a transit. Also, in this particular case the navigation data contained more bit transitions than a pure random sequence as simulated above, because the requirement of the number of data bits is lower. This is a favorable phenomenon but needs to be confirmed using longer data sets.

Figure 11 shows the field test result as a function of Doppler error for an SSR of 90%. Compared with simulation results (also included in Figure 11), the field test results are quite consistent. In particular, when the frequency error reaches to 25 Hz (not shown in Figure 11), the SSR almost falls to 0% no matter how many bits are used.

**B. ML Bit Decoding Algorithm**

The requirements of ML bit decoding algorithm are a bit different from those of bit synchronization. The major difference is that the bit decoding error rate will not decrease with more navigation data bits. On the contrary, as more data bits are decoded at a time, the tolerance to Doppler error is lower. However, at least 2 data bits should be implemented in one batch, due to the energy based ML bit decoding algorithm actually estimates bit transits instead of real bit values.

Using Monte Carlo tests from 1,000 to 10,000 trials (as a function of number of bits being decoded at one time), SDR values are generated as a function of different C/N0 and number of bits being decoded and the results are shown in TABLE I. As shown, the ML bit decoding algorithm shows more consistent SDR values for a given C/N0. As such, only a SDR value of 90% is used to generate the requirements for ML bit decoding.

**TABLE I. DIFFERENT SUCCESSFUL DETECT RATES (SDR) AS A FUNCTION OF C/N0 AND NUMBERS OF BITS BEING DECODED (VALUES IN RED ARE BELOW 90% SDR)**

<table>
<thead>
<tr>
<th>Number of Data Bits</th>
<th>C/N0 (dB-Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>76</td>
</tr>
<tr>
<td>3</td>
<td>66</td>
</tr>
<tr>
<td>4</td>
<td>57</td>
</tr>
<tr>
<td>5</td>
<td>N/A</td>
</tr>
<tr>
<td>6</td>
<td>N/A</td>
</tr>
<tr>
<td>7</td>
<td>N/A</td>
</tr>
<tr>
<td>8</td>
<td>N/A</td>
</tr>
<tr>
<td>9</td>
<td>N/A</td>
</tr>
<tr>
<td>10</td>
<td>N/A</td>
</tr>
<tr>
<td>20</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Results show that as more data bits are being decoded the SDR decreases, as expected. For an SDR of 90%, the result indicates that signals should have C/N0 of 20 dB-Hz or higher. Furthermore, for C/N0 = 20 dB-Hz, the number of bits being decoded should not exceed four. For C/N0 = 25 dB-Hz or higher, the number of bits being decoded can be as large as 20 bits (the combination of more than 20 bits will not be considered due to the increased computational load). It is noted that one would not likely try to decode 20 bits due to the associated processing requirements; rather the bits would likely be decoded pairwise. However, from a practical perspective, the fact that 20 bits could be decoded correctly – either all at
once or in some combination of small groups – means that coherent integration could also be extended to 20 ms as well, thus providing more processing gain.

Next, Figure 12 shows the maximum number of bits to decode as a function of Doppler error for different C/N\textsubscript{0} values in order to obtain an SDR of 90%. The corresponding field test results are shown in Figure 13.

![Figure 12. The simulation result of bit tolerance as a function of Doppler error for different signal strengths and for a 90% SDR](image)

As can be seen, both results show that the largest tolerance of frequency error is 11 Hz (when C/N\textsubscript{0} is equal to 40 dB-Hz), and the highest tolerance of signal strength is 20 dB-Hz. Compared to ML bit synchronization, the ML bit decoding algorithm has a relatively lower tolerance of Doppler/frequency error and signal strength. This is mainly because ML bit synchronization can obtain higher sensitivity by using longer bits sequence, but the bit error rate (BER) in ML bit decoding is a function of C/N\textsubscript{0}, and is insensitive to the length of bit sequence. However, working with signal strengths of 20 dB-Hz can still be beneficial for many applications.

V. CONCLUSION

This paper has analyzed the requirements of extending coherent integration by using the ML estimate algorithm in the context of stand-alone GNSS receivers. Both the simulated GPS data and the real field data are used for investigating the performance of ML bit synchronization and bit decoding algorithms in terms of the number of data bits required (for bit synchronization) and the tolerance to Doppler/frequency error as a function of the received signal strength. The criteria of SSR and SDR have been used to generate the requirements of the ML bit synchronization algorithm and ML bit decoding algorithm respectively. The results can be beneficial for stand-alone high sensitivity GNSS receiver design.

For ML bit synchronization, a higher SSR, a lower C/N\textsubscript{0} and a higher Doppler frequency error all require more data bits. By using 200 data bits, a 90% SSR can be achieved with C/N\textsubscript{0} as low as 15 dB-Hz with no Doppler error. The maximum tolerance of Doppler error is 24 Hz.

In contrast to the ML bit synchronization, ML bit decoding requires fewer bits to be decoded at one time to achieve a better tolerance of weaker signal strength and higher Doppler frequency error, e.g., for a C/N\textsubscript{0} of 20 dB-Hz, two data bits combination can tolerate Doppler error of 6 Hz but ten bits combination can only tolerate Doppler error of 1 Hz. In order to obtain an SDR of 90% using only two data bits (i.e., the best case scenario), the Doppler error should be less than 11 Hz for signals with C/N\textsubscript{0} of 40 dB-Hz and should be less than 7 Hz for a C/N\textsubscript{0} of 20 dB-Hz.

Future work will use more field tests under different environments to confirm the results presented here. Also, the information will be used to define parameters within a software receiver in order to improve navigation performance.

ACKNOWLEDGMENTS

The research presented in this paper was conducted as part of a collaborative research and development grant between the University of Calgary, General Motors of Canada, and Natural Sciences and Engineering Research Council of Canada.

REFERENCES


