

CODE AND CARRIER TRACKING AND NAVIGATION MESSAGE DECODING FOR WEAK GPS L2 CIVIL SIGNALS*

Nesreen I. Ziedan⁺

Computer & Systems Engineering Dept., Faculty of Engineering, Zagazig University, Egypt

ABSTRACT

GPS signals are severely attenuated when GPS receivers do not have a clear view of the sky, and conventional algorithms fail to detect such weak signals. This paper introduces novel code and carrier tracking and navigation message decoding algorithms to work with the new GPS L2 civil signals under weak signal conditions. The tracking algorithms are based on the Extended Kalman filter (EKF), where the L2-CM and L2-CL signals are combined to form the integration and the EKF equations. The message decoding algorithm is based on the Viterbi algorithm, where it is designed to decode the convolutionally encoded message, CNAV. The tracking sensitivity is increased by using long integrations. The effect of high bit error rate, that comes with weak signals, is minimized by using a likelihood method to weight the estimated data bits before incorporating them into the EKF equations. The algorithms are tested using simulated GPS L2C signals. The results show the ability of the algorithms to work efficiently with signals that have carrier to noise ratio as low as 10 dB-Hz.

KEYWORDS: GPS, Tracking, Kalman Filter, Viterbi Algorithm, Weak Signals.

CODE ET DE SUIVI DE SUPPORT ET MESSAGE DE NAVIGATION DCODAGE DE FAIBLE GPS L2 SIGNAUX CIVILS

RÉSUMÉ:

Les signaux GPS sont sévèrement atténués lorsque les récepteurs GPS n'ont pas une vue claire du ciel, et les algorithmes classiques ne parviennent pas à détecter de tels signaux faibles. Cet article présente le code roman et poursuite de la porteuse et le message de navigation des algorithmes de décodage de travailler avec le nouveau GPS L2 signaux civils dans des conditions de faible signal. Les algorithmes de suivi sont basés sur le filtre de Kalman étendu (EKF), où les signaux L2 et L2-CM-CL sont combinés pour former l'intégration et les équations EKF. Le message algorithmique de décodage est basé sur l'algorithme de Viterbi, où il est conçu pour décoder le message codé convolutionnellement, CNAV. La sensibilité de suivi est augmentée à l'aide de longues intégrations. L'effet du taux d'erreur binaire élevé, qui vient avec des signaux faibles, est minimisé en utilisant une méthode de vraisemblance pour pondérer les bits de données estimé avant de les intégrer dans les équations EKF. Les algorithmes sont testés en utilisant les signaux GPS L2C simulées. Les résultats montrent la capacité des algorithmes de travailler efficacement avec des signaux qui ont rapport porteuse / bruit aussi bas que 10 dB Hz.

MOTS-CLÉS: GPS, suivi, filtre de Kalman, l'algorithme de Viterbi, des signaux faibles.

* Received 5/6/2010, accepted: 5/9/2010

⁺ Author whom to contact (0113002842)

1 INTRODUCTION

The GPS L2 civil signal, known as L2C, is a new and more robust signal that started transmission on the Block IIR-M satellites, which were launched between September 2005 and August 2009. The L2C signal has new structure and enhanced properties over the civil L1 C/A signal [1] [2] [3]. It consists of two codes, CM and CL, that are multiplexed chip-by-chip, i.e. a chip of the CM code is transmitted followed by a chip of the CL code. The combined codes have a chipping rate of 1.023 MHz, with 511.5 kHz each. The CM code has a length of 10230 chips, repeats every 20 ms, and is modulated by a 50 Hz message. The CL code is 75 times longer than the CM code (76,7250 chips), and it is dataless. The L2C navigation message, CNAV, is convolutionally encoded with a rate of 1/2 and a constraint length 7 [4]. Unlike the structure of the current legacy navigation message, LNAV, which consists of subframes repeating in a fixed pattern, CNAV structure has individual messages that can be broadcast in a flexible order with variable repeat cycles [5]. The CNAV message type 0 started broadcasting from one IIR-M satellite in late 2009. The navigation message contains error checking bits to detect errors in the decoded data; the data can be decoded correctly even if they are detected with inverted signs [4].

Following signal acquisition, a tracking module works to provide continuous accurate estimate of the code delay and carrier parameters. The carrier parameters are used to wipe-off the carrier and consequently decode the navigation message. The code delay is used to calculate the pseudorange, which is used to calculate the navigation solution.

There are two main tracking approaches, which are hardware-based and software-based. Conventional hardware tracking uses tracking loops [6] [7], while software tracking implements all the functionalities of the tracking loops in software [8]. This makes software tracking more flexible to use advanced techniques that give better performance. Extended Kalman Filter-based (EKF) techniques were introduced in [9] to enable the tracking of C/A weak sidelobe signals in high altitude satellites. EKF-based C/A tracking algorithms and LNAV message decoding were introduced in [10] to work with weak signals under various dynamic conditions. Techniques to increase the robustness of L2C tracking through ionospheric scintillations were demonstrated in [11]. Performance evaluation of L2C tracking using different combinations of discriminators and channels were presented in [12]. A software receiver capable of processing L2C signals was developed in [13].

Conventional tracking approaches

can work with signals that have carrier to noise ratio (C/N_0) higher than 35 dB-Hz, but they fail to work with weaker signals. Tracking of weak signals requires long coherent and incoherent integrations to increase sensitivity. The coherent integration is more robust because it averages the noise out, but it is limited because of the unknown values of data bits that modulate the code. For the L2 CM signal, this limit is 20 ms. Increasing the coherent integration beyond this limit will incur the risk of integrating across data bits with different signs, which will be equivalent to a shorter integration time. The incoherent integration, which squares the signals before integration, does not have this limit on its length, but it is less sensitive. In addition, incoherent integration is not suitable for carrier tracking.

This paper introduces novel L2C algorithms for the code and carrier tracking and the CNAV message decoding to work under weak signal conditions. The algorithms development is motivated by applications like wireless positioning and indoor navigation, which suffer from signal attenuation. The algorithms utilize the signal structure and properties to achieve high performance. The tracking algorithms are based on EKF, and they combine the CM and CL signals to form the integration and the EKF equations. The CNAV decoding is based on the Viterbi

Algorithm (VA), which is an optimal dynamic programming technique [14]. The navigation message structure is utilized to aid tracking. Following the decoding of each part of the navigation message, the error checking bits are used to check if the data are decoded correctly. The correctly decoded data are re-encoded and passed on to the tracking module to be used as an aid when the same data are repeated in the navigation message.

2 SIGNAL MODELS

The effect of the Doppler shift on the code length is considered in the signal models- the Doppler shift either expands or shrinks the code length based on its polarity. Define T_i as the length of one CM code, taking into account the Doppler effect on the code length. T_i can be calculated as

$$T_i = T_c \frac{f_{L2}}{f_{L2} + \hat{f}_{d_i} + \hat{\alpha}_i T_c / 2}. \quad (1)$$

Where, T_c is the code length without Doppler effect. f_{L2} is the L2 carrier frequency. \hat{f}_{d_i} is the estimated Doppler shift over the interval T_i . $\hat{\alpha}_i$ is the estimated Doppler rate. The reception time of the samples of the received signal is calculated relative to the start of tracking; it is defined as t_i , $i = 0, \dots, L_i - 1$, where L_i is the number of samples in the i^{th} interval. The received IF sampled signal is modeled as

$$r_{L2C}(t_{ij}) = A \{ d(t_{ij}, f_d) C_{M0}(t_{ij}, f_d) + C_{0L}(t_{ij}, f_d) \} \cos \left(\theta_0 + \theta_{n_{ij}} + 2\pi (f_{IF} + f_{d0}) t_{ij} + \pi \alpha t_{ij}^2 \right) + n_{ij}. \quad (2)$$

Where, $t_{ij}, f_d = (t_{ij} - \tau_{e_i}) \left(1 + \frac{f_{d0} + \alpha t_{ij}/2}{f_{L2}} \right)$ is the time of the samples considering the Doppler effect on the code length. τ_{e_i} is the code delay error. θ_0 and f_{d0} are the phase and Doppler shift at the start of the tracking. α is the Doppler rate. A is the signal amplitude. d is the navigation data. f_{IF} is the IF carrier frequency. $\theta_{n_{ij}}$ is the accumulated clock noise at time t_{ij} ; it is composed of the total phase and frequency clock disturbances. n is a white Gaussian noise (WGN). The CM and CL codes are modeled such that C_{M0} is a chip-by-chip combination of the CM code and zeros, and C_{0L} is a chip-by-chip combination of zeros and the CL code.

A signal generator is used to obtain the local CM and CL replica signals. The times of the samples of the local signals are calculated relative to the start of an interval. The first sample in an interval is not necessarily located exactly on the start of the interval. Define λ_i as the difference between the time of the first sample in the i^{th} interval and the start of that interval. If T_s is the sampling time, then $0 \leq \lambda_i < T_s$.

The local signals are modeled as

$$Y_{L2C}(t_i + \delta) = C_{L2}(t_i, \delta, \hat{f}_{d_i}) e^{j\hat{\theta}_i}, \quad (3)$$

$$\hat{\theta}_i = \hat{\theta}_i + 2\pi (f_{IF} + \hat{f}_{d_i}) (\lambda_i + IT_s + \delta) + \pi \hat{\alpha}_i (\lambda_i + IT_s + \delta)^2.$$

δ is a delay induced in the local signal to produce either a prompt signal ($\delta = 0$), an early signal ($\delta = -\Delta$), or a late signal ($\delta = \Delta$). C_{L2} is the local code, which is C_{M0} for the CM code, and C_{0L} for the CL code. $\hat{\theta}_i$ and \hat{f}_{d_i} are, respectively, the estimated phase and Doppler shift at the start of the i^{th} interval. $\hat{\alpha}_i$ is the estimated Doppler rate. $t_i, \delta, \hat{f}_{d_i} = (\lambda_i + IT_s + \delta) \left(1 + \frac{\hat{f}_{d_i} + \hat{\alpha}_i (\lambda_i + IT_s + \delta)/2}{f_{L2}} \right)$.

The integrated signals are obtained by correlating the received signal with the local signals. The models of the in-phase and quad-phase of the CM integrated signal, at the i^{th} interval, are

$$I_{CM_i}(\delta) = A d_i R(\tau_{e_i} + \delta) \text{sinc}(\tilde{f}_{e_i} T_i) \cos(\tilde{\theta}_{e_i}) + n_{ICM_i},$$

$$Q_{CM_i}(\delta) = A d_i R(\tau_{e_i} + \delta) \text{sinc}(\tilde{f}_{e_i} T_i) \sin(\tilde{\theta}_{e_i}) + n_{QCM_i}.$$

Where, τ_{e_i} is the code delay estimation error over the interval. $R(\cdot)$ is the auto-correlation function. $\tilde{\theta}_{e_i}$ and \tilde{f}_{e_i} are the average phase error and Doppler shift error, respectively, over the interval. $\tilde{f}_{e_i} = f_{e_i} + \alpha_{e_i} \frac{T_i}{2}$. f_{e_i} is the Doppler shift estimation error at the start of the interval. α_{e_i} is the Doppler rate estimation error over the interval. A is the

signal level; it is normalized to drive the noise variance to 1 as in [6]. d_i is the data bit. n_{ICM_i} and n_{QCM_i} are independent WGN with zero mean and unit variance. The models of the CL integrated signal, I_{CL_i} and Q_{CL_i} , are similar to those of the CM signal, except that there are no data bits. The noise terms for the CL integrated signals, defined as n_{ICL_i} and n_{QCL_i} , are WGN and independent from those of the CM integrated signals.

3 MODULES INTERACTION

Tracking consists of four modules: signal generator; carrier tracking; code tracking; and CNAV message decoding. Figure 1 illustrates the modules' interaction. The code and carrier tracking algorithms are based on EKF, but they do not use the same integration length. The u^{th} carrier tracking interval is defined as T_{car_u} , and the v^{th} code tracking interval is defined as T_{code_v} . To synchronize the operation of different modules, each module generates updated outputs every T_i ms for the other modules to use.

The signal generator outputs are the early, prompt and late signals. The code tracking outputs are the estimated code delay $\hat{\tau}_i$, delay error $\hat{\tau}_{e_i}$, code length taking into account the Doppler effect, and start time of the next code. The carrier tracking outputs are the estimated phase $\hat{\theta}_i$, phase error $\hat{\theta}_{e_i}$,

Doppler shift $\hat{\omega}_{d_i}$, Doppler shift error $\hat{\omega}_{e_i}$, Doppler rate $\hat{\alpha}_{\omega_i}$, rate error $\hat{\alpha}_{\omega_{e_i}}$, signal level \hat{A}_i , and data bit \hat{d}_i if the signal is modulated by a data message.

Since the data are convolutionally encoded, the detected and decoded data are not the same. The detected data are the transmitted data, while the decoded data are the actual message. Since the navigation message is structured such that its contents repeat with predetermined cycles, this structure is utilized to aid tracking; the correctly decoded data bits are re-encoded and passed on to the tracking modules to wipe off the data signs when the same data bits are repeated. The CNAV decoding generates two outputs, which are a flag, F_{d_i} , and a data bit, d_{L_i} . F_{d_i} is set to 1 to indicate that d_{L_i} is a re-encoded data bit that is coming from a correctly decoded data, and is set to 0 otherwise.

The signal generator outputs are used by the code and carrier tracking to form the EKF measurements. The code tracking module uses the outputs of the carrier tracking module in the EKF equations to calculate the expected measurements, the linearization function and the residual. It also uses the Doppler shift and rate estimates to calculate the code length taking into account the Doppler effect. The carrier tracking uses the code length estimate provided by the code tracking. The signal generator uses the outputs of the code and carrier tracking to gen-

erate the local signals. The CNAV decoding uses the estimates of \hat{A}_i and \hat{d}_i . If $F_{d_i} = 1$, the tracking modules use the correct data values, d_{L_i} , otherwise they use the estimated values, \hat{d}_i .

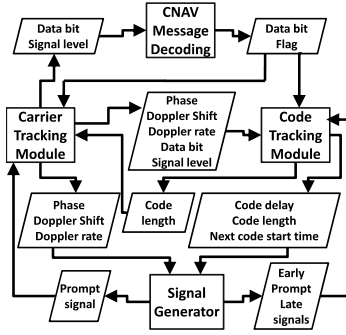


Figure 1. Interaction between the four modules. The modules are represented by rectangles, and the inputs/outputs are represented by parallelograms.

4 SIGNAL GENERATOR

The signal generator aligns the replica code with the received code, and generates the early, prompt and late integrated signals. The time of the start of an interval is estimated as $t_{start_i} = t_{start_{i-1}} + T_i$, where $t_{start_1} = 0$. As mentioned earlier, the time of the first sample in an interval, t_{i_1} , is not necessarily equal to t_{start_i} . If t_{i_1} is the time of the first sample in the first interval, then the residual of the division of $(t_{i_1} - t_{i_1})$ by the sampling time, T_s , should be zero,

i.e. $\{(t_{i_1} - t_{i_1}) \bmod T_s\} = 0$. Thus, if $t_{Res_i} = \{(t_{start_i} - t_{i_1}) \bmod T_s\}$, then t_{i_1} is estimated from

$$t_{i_1} = \begin{cases} t_{start_i} & \text{if } t_{Res_i} = 0 \\ t_{start_i} + T_s - t_{Res_i} & \text{if } t_{Res_i} \neq 0 \end{cases} \quad (4)$$

The local signals and the integrated signals are generated as in section 2.

5 TRACKING

The integrated signals are formed using a combination of the CL and CM codes. The integration lengths, T_{car_u} and T_{code_v} , can be multiple of one CM code length. The carrier tracking uses coherent integration only. The code tracking uses a combination of coherent and incoherent integrations. The integration lengths change adaptively depending on the tracking condition. Small integration is used to achieve rapid EKF convergence. Long integration is used to achieve high tracking accuracy. Small coherent integration is used if large tracking error is detected.

In order to use long coherent integration, data signs must be removed. If some of the data are correctly decoded, the re-encoded data are provided as an aid by the CNAV decoding module to remove the signs, otherwise the estimates of \hat{d}_i are used. The actual length, T_{dm} , of each data bit is 10 ms since the two codes are multiplexed. The theoretical minimum BER- which

is $\tilde{\beta} = 1/2 \operatorname{erfc} \sqrt{C/N_0 T_{dm}}$, where erfc is the complementary error function- is high for very weak signals, and hence the incorrect estimates of \hat{d}_i could degrade the integration. To minimize the degradation, if $\tilde{\beta}$ is above a predefined threshold γ_{BER} , \hat{d}_i is weighted by an estimate of its likelihood. Another considered issue is the possibility of a 180° phase error that will cause the data to be estimated with inverted signs. This can degrade the integration when using the re-encoded data, d_{L_i} , since they have correct signs. This problem is handled by multiplying d_{L_i} by -1 if such phase error is detected. It can be detected directly if $\tilde{\beta} \leq \gamma_{BER}$ by comparing \hat{d}_i and d_{L_i} over a number of bits. If \hat{d}_i and d_{L_i} have the same signs over more than half the compared bits, a flag F_{slip_u} is set to 1 to indicate that there is no 180° phase error, otherwise F_{slip_u} is set to -1 . If $\tilde{\beta} > \gamma_{BER}$, the 180° phase error detection is combined in the calculation of the likelihood of \hat{d}_i .

$\tilde{\beta}$ can be calculated from the estimated C/N_0 . The actual BER, $\hat{\beta}$, can be estimated over the repeated data that have already been correctly decoded. This is done by comparing the estimates, \hat{d}_i , with the actual values, d_{L_i} , over $N_{\tilde{\beta}}$ data bits and counting the number of bits with incorrect signs, N_{β_0} . Since there could be undetected 180° phase error, $\hat{\beta} = \min \{ N_{\beta_0}/N_{\tilde{\beta}}, (N_{\tilde{\beta}} - N_{\beta_0})/N_{\tilde{\beta}} \}$.

The tracking performance is moni-

tored to detect large frequency or code delay errors. The C/N_0 can be estimated, following an acquisition process, and used to estimate the expected signal level, \hat{A} . A large frequency error is concluded either if the signed signal level of the data-less CL integration changes sign randomly and is degraded in value, or if $\hat{\beta} > \zeta \tilde{\beta}$ (given that $\tilde{\beta}$ is available), $\zeta > 1$. A carrier re-initialization module is activated to refine the frequency and the other carrier parameters. A large code delay error is concluded if the signal levels of the CM and CL integrations are degraded. An acquisition-like module is activated to refine the delay error. It tests only a small number of possible code delays, which are located around the current delay estimate. The EKF for carrier and code tracking are de-activated during the carrier re-initialization and code re-acquisition, respectively, and re-activated afterwards.

5.1 Data Likelihood Estimation

If $\tilde{\beta} \leq \gamma_{BER}$, the in-phase and quad-phase prompt signals, in the u^{th} interval, are

$$I_{L2C_u}(0) = \frac{1}{\sqrt{N_u}} \sum_{i=g_u}^{g_u+N_u-1} \left\{ [(1 - F_{d_i}) \hat{d}_i + F_{d_i} F_{slip_u} d_{L_i}] I_{CM_i}(0) + I_{CL_i}(0) \right\},$$

$$Q_{L2C_u}(0) = \frac{1}{\sqrt{N_u}} \sum_{i=g_u}^{g_u+N_u-1} \left\{ \left[(1-F_{d_i}) \hat{d}_i + F_{d_i} F_{slip_u} d_{L_i} \right] Q_{CM_i}(0) + Q_{CL_i}(0) \right\}.$$

Where, N_u is the number of the CM codes in the carrier integration interval, T_{car_u} , and g_u is the index of the first CM code in the interval.

If $\bar{\beta} > \gamma_{BER}$, then each data bit is estimated and weighted by its likelihood. Define N_{bit_u} as the number of data bits in the T_{car_u} interval that have been correctly decoded at a previous cycle of repeated data. Since there are a total of N_u data bits in the T_{car_u} interval, there are $N_{dc_u} = 2^{N_u - N_{bit_u}}$ possible data bit combinations. This is because no data bit combinations are considered at the positions of the correctly decoded data- these positions are set based on their correct values. To account for the possible 180° phase error, two groups of possible data bit combinations are formed. One group sets the correctly decoded data, in each combination, to d_{L_i} , while the other group sets the correctly decoded data to $-d_{L_i}$. The data bits are estimated from the group that has the larger likelihood. Let E_{u1} define a data combination in the first group, while E_{u2} define a data combination in the second group, where $E = 1, \dots, N_{dc_u}$. Let $\bar{d}_{E_{u1_i}}$ and $\bar{d}_{E_{u2_i}}$ define the i^{th} data bit value

within, respectively, E_{u1} and E_{u2} possible data bit combinations. Let

$$s_i = \text{sign}(I_{CM_i}(0)) \sqrt{I_{CM_i}^2(0) + Q_{CM_i}^2(0)}. \quad (5)$$

Consider \mathbf{s} as a random variable with a normal distribution density function. Let s_j be the observed values of \mathbf{s} . The likelihood function of \mathbf{s} is the product of the likelihood functions of s_j [15], chapter 9. Thus, the likelihood of each data bit combination, in each group, is calculated as

$$L_{E_{uJ}} = \frac{1}{(2\pi)^{\frac{N_u}{2}}} \exp\left(-\frac{1}{2} \sum_{i=g_u}^{g_u+N_u-1} |s_i - \bar{A} \bar{d}_{E_{uJ_i}}|^2\right).$$

Where, J refers to the index of the group, i.e. $J = 1$ or 2 . E_u is a possible data bit combination. If $\sum_{E=1}^{N_{dc_u}} L_{E_{u1}} > \sum_{E=1}^{N_{dc_u}} L_{E_{u2}}$, the first group of data combinations is chosen, i.e. $L_{E_u} = L_{E_{u1}}$, $E_u = E_{u1}$, $\bar{d}_{E_{u1_i}} = \bar{d}_{E_{u1_i}}$, otherwise the second group is chosen, i.e. $L_{E_u} = L_{E_{u2}}$, $E_u = E_{u2}$, $\bar{d}_{E_{u1_i}} = \bar{d}_{E_{u2_i}}$. Each possible data combination, from the chosen group, is weighted based on its likelihood, where the summation of the weights is equal to 1. The weights are calculated from

$$\Lambda_{E_u} = \frac{L_{E_u}}{\sum_{E=1}^{N_{dc_u}} L_{E_u}}. \quad (6)$$

The weights Λ_{E_u} are then used to estimate the likelihood of each data bit;

the result is defined as μ_i , $|\mu_i| \leq 1$. If the i^{th} data bit is correctly decoded, then $\mu_i = \tilde{d}_{E_{u_i}}$, otherwise μ_i of each data is calculated as follows. Two weights are calculated for each data bit. The first weight, defined as P_{W_i} , assumes that the data bit is $+1$, and the second weight, defined as N_{W_i} , assumes that the data bit is -1 . P_{W_i} is the summation of Λ_{E_u} of the combinations that have this data bit as $+1$, while N_{W_i} is the summation of Λ_{E_u} of the combinations that have this data bit as -1 . If $P_{W_i} > N_{W_i}$, then $\mu_i = (+1)P_{W_i}$, otherwise, $\mu_i = (-1)N_{W_i}$. The prompt signals are

$$I_{L2C_u}(0) = \frac{1}{\sqrt{N_u}} \sum_{i=g_u}^{g_u+N_u-1} \{\mu_i I_{CM_i}(0) + I_{CL_i}(0)\},$$

$$Q_{L2C_u}(0) = \frac{1}{\sqrt{N_u}} \sum_{i=g_u}^{g_u+N_u-1} \{\mu_i Q_{CM_i}(0) + Q_{CL_i}(0)\}.$$

The estimated average signal level over the u^{th} interval is calculated as

$$\hat{A}_u = \sqrt{I_{L2C_u}(0)^2 + Q_{L2C_u}(0)^2}. \quad (7)$$

The estimated average signal levels over each T_i ms are calculated as

$$\hat{A}_{CM_i} = \sqrt{I_{CM_i}(0)^2 + Q_{CM_i}(0)^2}, \quad (8)$$

$$\hat{A}_{CL_i} = \sqrt{I_{CL_i}(0)^2 + Q_{CL_i}(0)^2}. \quad (9)$$

5.2 EKF Carrier Tracking

EKF model:

$\tilde{\theta}_u$, $\tilde{\omega}_{d_u}$, and $\tilde{\alpha}_{\omega_u}$ are, respectively, the average phase, average radian Doppler shift, and radian Doppler rate over the u^{th} measurement interval. The state vector is

$$x_u = \begin{bmatrix} \tilde{\theta}_u \\ \tilde{\omega}_{d_u} \\ \tilde{\alpha}_{\omega_u} \end{bmatrix}.$$

Where, $\tilde{\omega}_{d_u} = 2\pi \tilde{f}_{d_u}$. $\tilde{\alpha}_{\omega_u} = 2\pi \tilde{\alpha}_u$. Let:

$$\dot{x}_u = F x_u + W_u,$$

$$\begin{bmatrix} \dot{\tilde{\theta}}_u \\ \dot{\tilde{\omega}}_{d_u} \\ \dot{\tilde{\alpha}}_{\omega_u} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{\theta}_u \\ \tilde{\omega}_{d_u} \\ \tilde{\alpha}_{\omega_u} \end{bmatrix} + \begin{bmatrix} W_{\theta_u} \\ W_{\omega_u} \\ W_{\alpha_u} \end{bmatrix}.$$

Where, W_{θ_u} , W_{ω_u} , and $W_{\alpha_{\omega_u}}$ are the noise disturbances of the phase, Doppler shift and Doppler rate, respectively. The transition matrix, which is derived from Taylor series, is $\Phi = I + F t + \frac{1}{2} F^2 t^2$. The measurement length, T_{car_u} , can be changed to optimize the tracking performance. This change is reflected in the design of the dynamic model, which propagates the state vector from an interval to the next. If $T_{car_{u+1}} = T_{car_u}$, then the dynamic model is

$$\begin{bmatrix} \tilde{\theta}_{u+1} \\ \tilde{\omega}_{d_{u+1}} \\ \tilde{\alpha}_{\omega_{u+1}} \end{bmatrix} = \begin{bmatrix} 1 & T_{car_u} & \frac{T_{car_u}^2}{2} \\ 0 & 1 & T_{car_u} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{\theta}_u \\ \tilde{\omega}_{d_u} \\ \tilde{\alpha}_{\omega_u} \end{bmatrix} + \begin{bmatrix} T_{car_u} \\ 0 \\ 0 \end{bmatrix} \omega_{IF} + \begin{bmatrix} W_{\theta,u+1} \\ W_{\omega,u+1} \\ W_{\alpha,u+1} \end{bmatrix}. \quad (10)$$

If $T_{car_{u+1}} \neq T_{car_u}$, then first the Doppler rate, Doppler shift and phase are propagated to the start of the interval $T_{car_{u+1}}$, so $\alpha_{\omega_{u+1}} = \tilde{\alpha}_{\omega_u}$, $\omega_{d_{u+1}} = \tilde{\omega}_{d_u} + \alpha_{\omega_{u+1}} \frac{T_{car_u}}{2}$, and $\theta_{u+1} = \tilde{\theta}_u + (\omega_{d_{u+1}} + \omega_{IF}) \frac{T_{car_u}}{2} + \alpha_{\omega_{u+1}} \frac{T_{car_u}^2}{6}$. These propagated values are then used in the dynamic model, so

$$\begin{bmatrix} \tilde{\theta}_{u+1} \\ \tilde{\omega}_{d_{u+1}} \\ \tilde{\alpha}_{\omega_{u+1}} \end{bmatrix} = \begin{bmatrix} 1 & \frac{T_{car_{u+1}}}{2} & \frac{T_{car_{u+1}}^2}{6} \\ 0 & 1 & \frac{T_{car_{u+1}}}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{u+1} \\ \omega_{d_{u+1}} \\ \alpha_{\omega_{u+1}} \end{bmatrix} + \begin{bmatrix} \frac{T_{car_{u+1}}}{2} \\ 0 \\ 0 \end{bmatrix} \omega_{IF} + \begin{bmatrix} W_{\theta,u+1} \\ W_{\omega,u+1} \\ W_{\alpha,u+1} \end{bmatrix}. \quad (11)$$

The phase and frequency disturbances are modeled as normal random

walks, each with zero mean, and they are assumed to be independent. The variances are defined as $E[W_{\theta} W_{\theta}^T] = Q_{\theta}$, $E[W_{\omega} W_{\omega}^T] = Q_{\omega}$, $E[W_{\alpha} W_{\alpha}^T] = Q_{\alpha}$. The discrete process noise can be derived as in [10]. The measurement noise is $R_{r+1} = 1$ (the integration is normalized to drive the noise variance to 1).

EKF estimation: The EKF propagates the *average* phase, Doppler shift and Doppler rate from the current interval T_{car_u} to the next one $T_{car_{u+1}}$, and then it updates them. The propagated state vector is x_{u+1}^- , and the updated one is x_{u+1}^+ . If $T_{car_{u+1}}$ is multiple of one CM code length, then estimates of the carrier parameters at each CM code interval are needed for the other modules. If the start of T_{i+1} coincides with the start of $T_{car_{u+1}}$ (i.e. the EKF just generated its new updates), then $\hat{\alpha}_{\omega_{i+1}} = \tilde{\alpha}_{\omega_{u+1}}$, $\hat{\omega}_{d_{i+1}} = \tilde{\omega}_{d_{u+1}} - \hat{\alpha}_{\omega_{i+1}} \frac{T_{car_{u+1}}}{2}$, and $\hat{\theta}_{i+1} = \tilde{\theta}_{u+1} - \hat{\omega}_{d_{i+1}} \frac{T_{car_{u+1}}}{2} - \hat{\alpha}_{\omega_{i+1}} \frac{T_{car_{u+1}}^2}{6}$. But if the start of T_{i+1} lies within the current measurement interval (i.e. no new EKF updates are available), then $\hat{\theta}_{i+1} = \hat{\theta}_i + \hat{\omega}_{d_i} T_i + \hat{\alpha}_{\omega_i} \frac{T_i^2}{2}$.

The measurement is taken as the prompt quad-phase signal, i.e. $z_{u+1} = Q_{L2C_{u+1}}(0) \equiv \tilde{A}_{u+1} \sin(\tilde{\theta}_{e_{u+1}})$. Where, \tilde{A}_{u+1} is the average signal level over the $(u+1)^{th}$ interval, and $\tilde{\theta}_{e_{u+1}}$ is the average phase error. The expected measurement is $\hat{z}_{u+1} = 0$.

The first derivative of h_{u+1} , where $z_{u+1} = h_{u+1} + n_{u+1}$, produces $H_{u+1} = [\hat{A}_{u+1} \cos(\theta_{e_{u+1}}^-) \ 0 \ 0] = [\hat{A}_{u+1} \ 0 \ 0]$. Where, \hat{A} is calculated as in (7). The residual is $Res_{u+1} = z_{u+1} - \hat{z}_{u+1} = Q_{L2C_{u+1}}(0)$. The state vector is propagated using either (10) or (11), while it is updated as $x_{u+1}^+ = x_{u+1}^- + K_{u+1} Res_{u+1}$. Where, K_{u+1} is the EKF gain.

A square root EKF is used with the carrier tracking module because of its stability. The square root EKF is derived from the Carlson algorithm as in [16], and its implementation is similar to the one used with the L1 C/A signal in [10], chapter 5.

5.3 EKF Code Tracking

The early, prompt and late signals are employed in the EKF equations. A combination of coherent and incoherent integrations is used to form the total integration. Define N_{acv} as the number of coherent integrations that are added incoherently to form the total integration in the v^{th} integration interval, T_{codev} , and N_{vk} as the number of CM codes in the k^{th} coherent integration of the N_{acv} integrations. If $\tilde{\beta} \leq \gamma_{BER}$, then the in-phase and quad-phase of the total integration are

$$I_{L2C_v}(\delta) = C_N \sum_{k=1}^{N_{acv}} \frac{1}{N_{vk}} \left\{ \sum_{i=g_{vk}}^{g_{vk}+N_{vk}-1} [(1-F_{d_i}) \hat{d}_i + F_{d_i} F_{slip_u} d_{L_i}] I_{CM_i}(\delta) + I_{CL_i}(\delta) \right\}^2,$$

$$Q_{L2C_v}(\delta) = C_N \sum_{k=1}^{N_{acv}} \frac{1}{N_{vk}} \left\{ \sum_{i=g_{vk}}^{g_{vk}+N_{vk}-1} [(1-F_{d_i}) \hat{d}_i + F_{d_i} F_{slip_u} d_{L_i}] Q_{CM_i}(\delta) + Q_{CL_i}(\delta) \right\}^2.$$

Where, $C_N = 1/\sqrt{2N_{acv}}$. If $\tilde{\beta} > \gamma_{BER}$, a method similar to the one described in the carrier tracking is used, with each coherent integration, to estimate the data bits and their likelihoods. The likelihood of each possible data bit combination is calculated as in (6) and the weights are calculated as in (6). So,

$$I_{L2C_v}(\delta) = C_N \sum_{k=1}^{N_{acv}} \frac{1}{N_{vk}} \left\{ \sum_{i=g_{vk}}^{g_{vk}+N_{vk}-1} \mu_i I_{CM_i}(\delta) + I_{CL_i}(\delta) \right\}^2,$$

$$Q_{L2C_v}(\delta) = C_N \sum_{k=1}^{N_{acv}} \frac{1}{N_{vk}} \left\{ \sum_{i=g_{vk}}^{g_{vk}+N_{vk}-1} \mu_i Q_{CM_i}(\delta) + Q_{CL_i}(\delta) \right\}^2.$$

Using the output of the carrier tracking, the *expected* signals are calculated as

$$\hat{I}_{L2C_v}(\delta) = C_N \sum_{k=1}^{N_{acv}} \frac{1}{N_{vk}} \left\{ \sum_{i=g_{vk}}^{g_{vk}+N_{vk}-1} \hat{A}_i \cos(\hat{\theta}_{e_i}) \hat{R}(\delta) \right\}^2,$$

$$\hat{Q}_{L2C_v}(\delta) = C_N \sum_{k=1}^{N_{acv}} \frac{1}{N_{vk}} \left\{ \sum_{i=g_{vk}}^{g_{vk}+N_{vk}-1} \hat{A}_i \sin(\hat{\theta}_{e_i}) \hat{R}(\delta) \right\}^2.$$

Differentiating I_{L2C_v} and Q_{L2C_v} give

$$\begin{aligned} \frac{\partial I_{L2C_v}(\delta)}{\partial \tau} &= C_N \sum_{k=1}^{N_{acv}} \frac{1}{N_{vk}} \left\{ \sum_{i=g_{vk}}^{g_{vk}+N_{vk}-1} \hat{A}_i \cos(\hat{\theta}_{e_i}) \right\}^2 \frac{\partial \hat{R}(\delta + \tau_{ev})^2}{\partial \tau}, \\ \frac{\partial Q_{L2C_v}(\delta)}{\partial \tau} &= C_N \sum_{k=1}^{N_{acv}} \frac{1}{N_{vk}} \left\{ \sum_{i=g_{vk}}^{g_{vk}+N_{vk}-1} \hat{A}_i \sin(\hat{\theta}_{e_i}) \right\}^2 \frac{\partial \hat{R}(\delta + \tau_{ev})^2}{\partial \tau}. \end{aligned}$$

EKF model: A second order EKF is used, where it makes use of higher orders in the Taylor series expansion. The dynamic model is $\tau_{e_{v+1}} = \Phi \tau_{e_v} + W_{\tau, v+1}$, where W_{τ} is a noise term, and $\Phi = 1$. The propagated code delay error is $\tau_{e_{v+1}}$. EKF keeps track of the estimated start of the code, so the propagated code delay error is zero. The

measurement vector is $z_{v+1} = h_{v+1} + n_{v+1}$, where n_{v+1} is the measurement noise. z_{v+1} has 3 measurements from the early, prompt, and late signals, thus

$$\check{z}_{v+1} = \begin{bmatrix} I_{L2C_{v+1}}(-\Delta) + Q_{L2C_{v+1}}(-\Delta) \\ I_{L2C_{v+1}}(\Delta) + Q_{L2C_{v+1}}(\Delta) \\ I_{L2C_{v+1}}(0) + Q_{L2C_{v+1}}(0) \end{bmatrix},$$

where, $z_{v+1} = 1/\sqrt{2} \check{z}_{v+1}$. The *expected* measurement \hat{z}_{v+1} and the linearization vector $H_{1v+1} = \frac{\partial h_{v+1}}{\partial \tau}$ can be found as in section 5.3. The second derivative of h_{v+1} produces a number of 1x1 matrices equal to the number of the measurements. Where the f^{th} matrix is $H_{2v+1,j} = \frac{\partial^2 h_{v+1,j}}{\partial \tau^2} |_{\tau_{e_{v+1}} = \tau_{e_{v+1}}^-}$. $h_{v+1,j}$ is the f^{th} row of h_{v+1} . The autocorrelation function is approximated by a quadratic function of the form $\hat{R}(\tau) = a \left(\frac{\tau}{T_{chip}} \right)^2 + b \left(\frac{\tau}{T_{chip}} \right) + c$. $\frac{\partial \hat{R}(\tau)}{\partial \tau}$ and $\frac{\partial^2 \hat{R}(\tau)}{\partial \tau^2}$ are used in H_{1v+1} and H_{2v+1} . The constants a , b , and c are determined by setting $\hat{R}(\tau)$ to 1 at the prompt signal, and to $1 - \frac{|\Delta|}{T_{chip}}$ at the early and late signals, and then solving

$$\begin{bmatrix} 1 - \frac{|\Delta|}{T_{chip}} \\ 1 - \frac{|\Delta|}{T_{chip}} \\ 1 \end{bmatrix} = \begin{bmatrix} \left(-\frac{\Delta}{T_{chip}} \right)^2 & -\frac{\Delta}{T_{chip}} & 1 \\ \left(\frac{\Delta}{T_{chip}} \right)^2 & \frac{\Delta}{T_{chip}} & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}.$$

The measurement noise matrix is

$$R_{n_{v+1}} = \begin{bmatrix} 1 & \left(1 - \frac{2\Delta}{T_{chip}}\right)^2 \\ \left(1 - \frac{2\Delta}{T_{chip}}\right)^2 & 1 \\ \left(1 - \frac{\Delta}{T_{chip}}\right)^2 & \left(1 - \frac{\Delta}{T_{chip}}\right)^2 \\ \left(1 - \frac{\Delta}{T_{chip}}\right)^2 & \left(1 - \frac{\Delta}{T_{chip}}\right)^2 \\ \left(1 - \frac{\Delta}{T_{chip}}\right)^2 & 1 \end{bmatrix}.$$

Two types of second order EKF are implemented, which are truncated second order filter (TS) and Gaussian second order filter (GS) [17]. Their designs are similar to those introduced in [10], chapter 5, for the L1 C/A signal.

The EKF produces *propagated* values every CM code interval, and produces *updated* values based on the length of $T_{code_{v+1}}$. If the start of T_{i+1} coincides with the start of $T_{code_{v+1}}$, then the estimated start time of the next received code is $t_{start_{v+1}} = t_{start_{i+1}} = t_{start_i} + T_i + K_{v+1}(Res_{v+1} - b_{v+1})$. Where, $t_{start_{i+1}}$ is the estimated start of $T_{code_{v+1}}$, $t_{start_{i+1}}$ is the estimated start of T_{i+1} , K_{v+1} is the gain of the second order EKF, Res_{v+1} is the EKF residual, and b_{v+1} is found from the second derivative of h_{v+1} as in [10], chapter 5. If the start of T_{i+1} lies within the current measurement interval (i.e. no new EKF updates are available), then $t_{start_{i+1}} = t_{start_i} + T_i$.

Code delay separation: A large code delay separation between early

and late signals can help in maintaining code lock if the code delay estimation error is large, while a small separation allows for a more accurate estimation. One of the methods, developed in [10], chapter 5, for the L1 C/A signal, to take advantage of various separations is employed in this paper. It can use more than one code delay separation in the same time, and it can change the delay separation adaptively. The code delay separation can be large at the beginning of the tracking and immediately after a re-acquisition process. Otherwise, it can be small, or more than one delay separation can be used.

Define $N_{\delta_{v+1}}$ as the number of the code delays that are used in the $(v+1)^{th}$ integration interval, and $\delta_{v+1,q}$ as the q^{th} delay separation in that interval. The z_{v+1} , \hat{z}_{v+1} , H_{1v+1} vectors will have a size of $(3N_{\delta_{v+1}})$, and $R_{n_{v+1}}$ will have a size of $(3N_{\delta_{v+1}}) \times (3N_{\delta_{v+1}})$. If $z_{v+1}(\delta_{v+1,q})$ defines the measurements that are generated using the q^{th} delay separation, then $z_{v+1} = [z_{v+1}(\delta_{v+1,1}) \dots z_{v+1}(\delta_{v+1,N_{\delta_{v+1}}})]^T$. The measurement noise is $R_{n_{v+1}} =$

$$\begin{bmatrix} \sigma_{v+1}(1,1) & \dots & \sigma_{v+1}(1,N) \\ \vdots & \ddots & \vdots \\ \sigma_{v+1}(N,1) & \dots & \sigma_{v+1}(N,N) \end{bmatrix}.$$

Where, $\sigma_{v+1}(q,g)$ is a 3x3 matrix that contains the noise variance between the delay separations $\delta_{v+1,q}$ and $\delta_{v+1,g}$. If $q = g$, σ_{v+1} is equal to $R_{n_{v+1}}$ in (12).

6 CNAV NAVIGATION MESSAGE DECODING

The CNAV message is Forward Error Correction (FEC) convolutionally encoded by a rate of $1/2$ and a constraint length of 7. The data input rate is 25 bit/s, and the data output rate is 50 symbol/s. The encoder has one input F_0 , 6 stages register (F_1, \dots, F_6) that stores the last 6 inputs, and 2 outputs G_1 and G_2 . The encoder polynomial (G_1, G_2) in octal representation is (171,133), i.e. converting the $-1/1$ data representation into $1/0$ representation, the outputs are generated from $G_1 = F_0 \oplus F_1 \oplus F_2 \oplus F_3 \oplus F_6$, and $G_2 = F_0 \oplus F_2 \oplus F_3 \oplus F_5 \oplus F_6$. The register has 2^6 states, where each state generates two pairs of the outputs G_1 and G_2 . One pair is generated if $F_0 = 1$, while the other pair is generated if $F_0 = -1$.

A decoder based on the Viterbi Algorithm (VA) is implemented in this paper. The VA is a dynamic programming technique [14] that works recursively to find the optimal sequence of states, or the optimal path. In this decoder, each path represents one possible sequence of N_D data bits. Because an optimal path consists of optimal sub-paths, after each recursive iteration, the current non-optimal sub-paths are discarded, while the optimal sub-paths are tracked. Each optimal sub-path is the one that generates the minimum weight leading up to one of the states. Every N_D iterations, all the tracked paths

are inspected to find the one with the overall minimum total weight, which is then used to decode a sequence of N_D bits.

The decoder works on a pair of noisy input data, in each iteration, that represent the transmitted encoder's output, G_1 and G_2 . The decoder has 2^6 states, where each state, M , has two pairs of possible outputs. One pair, $G_{1,M-1}$ and $G_{2,M-1}$, assumes that $F_0 = -1$, while the other pair, G_{1,M_1} and G_{2,M_1} , assumes that $F_0 = 1$. A state with a binary representation of $[B6\ B5\ B4\ B3\ B2\ B1]$ can make a transition either to $[0\ B6\ B5\ B4\ B3\ B2]$ or $[1\ B6\ B5\ B4\ B3\ B2]$. Figure 2 shows a partial state diagram of the decoder. Figure 3 shows a partial trellis graph, which is derived from the state diagram by following all the possible states transitions throughout its operation. It should be noted that a state with a binary representation of $[B6\ B5\ B4\ B3\ B2\ B1]$ will have transitions from either the state $[B5\ B4\ B3\ B2\ B1\ 0]$ or $[B5\ B4\ B3\ B2\ B1\ 1]$. Thus, if $0 \leq M \leq 31$, the two states making transitions to M are $2M$ and $2M+1$. But, if $32 \leq M \leq 63$, the two states making transitions to M are $2(M-31)$ and $2(M-31)+1$. Each transition is assigned a weight, which is derived from the noisy input data and either $G_{1,M-1}$ and $G_{2,M-1}$ or G_{1,M_1} and G_{2,M_1} . A cumulative path weight, Γ_{M_j} , is main-

tained with each state, M , and updated after each iteration, j . In each iteration, each state M is inspected to find the state that its transition to M leads to the minimum Γ_{M_j} . The history of the states that made transitions to each state is kept in one row in the matrix Y , which has 64 rows and N_D columns.

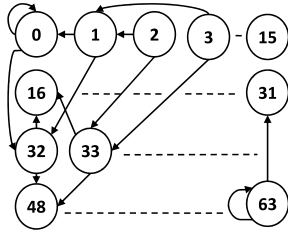


Figure 2. State diagram (partial) of the CNAV decoding.

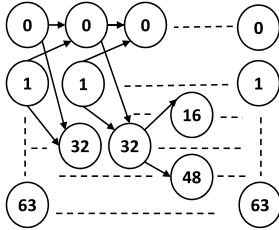


Figure 3. Trellis graph (partial) for the CNAV decoding.

The CNAV decoding module receives the inputs \hat{A}_i and \hat{d}_i from the carrier tracking module every T_i ms,

but it is updated every two consecutive T_i ms. It works as follows:

(1) Γ_{M_0} , for $M = 0, \dots, 63$, are initialized to zero.

(2) G_{1,M_1} , G_{2,M_1} , $G_{1,M_{-1}}$ and $G_{2,M_{-1}}$ are calculated, for each state, using the G_1 and G_2 equations, and then converted to ± 1 representation. There are 4 possible output combinations of G_1 and G_2 ($[-1 -1]$, $[-1 1]$, $[1 -1]$, $[1 1]$); these combinations are defined by the vectors, G_{ocy} , $Y = 1, \dots, 4$

(3) In the j^{th} iteration, the decoder processes the last two batches of data received from the carrier tracking module, i.e. \hat{d}_{i-1} , \hat{d}_i , \hat{A}_{i-1} , and \hat{A}_i . For each state, two weight functions are calculated as $W_{1,M_j} = (1/W_j) \{ |\hat{d}_{i-1} \hat{A}_{i-1} - \bar{A} G_{1,M_1}| + |\hat{d}_i \hat{A}_i - \bar{A} G_{2,M_1}| \}$, and $W_{-1,M_j} = (1/W_j) \{ |\hat{d}_{i-1} \hat{A}_{i-1} - \bar{A} G_{1,M_{-1}}| + |\hat{d}_i \hat{A}_i - \bar{A} G_{2,M_{-1}}| \}$. Where, \bar{A} is the expected signal level. W_j is a scale value used to prevent the cumulative weight from growing too large. It can be calculated as

$$W_{i_j} = \sum_{Y=1}^4 |\hat{d}_{i-1} \hat{A}_{i-1} - \bar{A} G_{ocy}(1)| + |\hat{d}_i \hat{A}_i - \bar{A} G_{ocy}(2)|.$$

(4) Two temporary cumulative summations, for each state M , are calculated as $\eta_{M_1} = \Gamma_{M_{j-1}} + W_{1,M_j}$, and $\eta_{M_2} = \Gamma_{M_{j-1}} + W_{-1,M_j}$. Each state will have two transitions from two other states. The transition that comes from the state with the minimum cumulative weight

is chosen, and the other one is discarded. Thus, the cumulative weights are updated as follows

$$\Gamma_{M_j} = \min \{ \eta_{(2M)_1}, \eta_{(2M+1)_1} \},$$

for $M = 0, \dots, 31$

$$\Gamma_{M_j} = \min \{ \eta_{(2(M-31))_2}, \eta_{(2(M-31)+1)_2} \},$$

for $M = 32, \dots, 63$.

The matrix Υ is updated to record the states that made transitions to each state. For $M = 0, \dots, 31$, if $\eta_{(2M)_1} < \eta_{(2M+1)_1}$, then $\Upsilon(M, j) = 2M$, otherwise $\Upsilon(M, j) = 2M + 1$. For $M = 32, \dots, 63$, if $\eta_{(2(M-31))_2} < \eta_{(2(M-31)+1)_2}$, $\Upsilon(M, j) = (2(M-31))$, otherwise $\Upsilon(M, j) = (2(M-31) + 1)$.

(5) Concluding a sequence of data is done every N_D iterations. The state, $M_{min_{N_D}}$, that generates the minimum $\Gamma_{M_{N_D}}$, for $M = 0, \dots, 63$, is used to backtrack the transitions to construct the path information from Υ , and consequently decode the navigation data. This is done as follows:

(5.1) The path information is initialized as $P(N_D) = M_{min_{N_D}}$, and the path is constructed as

$$P(k) = \Upsilon(P(k+1), k+1),$$

for $k = N_D - 1, N_D - 2, \dots, 0$.

(5.2) For a state with a binary representation of $[B6 \ B5 \ B4 \ B3 \ B2 \ B1]$, if this state makes a transition to state $[0 \ B6 \ B5 \ B4 \ B3 \ B2]$, then $F_0 = +1$, but if it makes a transition to

$[1 \ B6 \ B5 \ B4 \ B3 \ B2]$, then $F_0 = -1$. A transition to a state between 0–31 indicates the input data is +1, while a transition to a state between 32–63 indicates the input data is -1. Thus, the decoded data, i.e. the actual navigation message, can be concluded directly from the path information in $P(k)$, where $k = 1, \dots, N_D$. That is, if $0 \leq P(k) \leq 31$, then $d_k = 1$, but if $32 \leq P(k) \leq 63$, then $d_k = -1$.

7 SIMULATION AND RESULTS

The algorithms are tested using simulated L2C signals, with the low quality temperature compensated crystal oscillator (TCXO) clock and the high quality ovenized crystal oscillator (OXO) clock. The sampling rate is $f_s = 5700$ kHz. The initial phase error, Doppler shift, Doppler shift error, Doppler rate, and Doppler rate error are modeled as uniformly distributed random variables (UDRV) in the ranges of $(-\pi, \pi)$ radians, $(-10, 10)$ kHz, $(-10, 10)$ Hz, $(-300, 300)$ Hz/s, and $(-10, 10)$ Hz/s, respectively. The code delay error is modeled to take a value between $(0, 1/f_s)$. This value is obtained by generating a UDRV between $(0, \tau_{Precision} f_s)$, and dividing the result by $(\tau_{Precision} f_s^2)$, where $\tau_{Precision}$ is an integer constant. The oscillator phase and frequency noises are modeled with normal random walks as in

[18], and the variances are derived as in [19]. The values of the phase and frequency random walk intensities for the TCXO clock are $S_f = 5 * 10^{-21}$ s and $S_g = 5.9 * 10^{-20} s^{-1}$, and for the OXO clock are $S_f = 5 * 10^{-23}$ s and $S_g = 1.5 * 10^{-22} s^{-1}$. The CM code is modulated by a convolutionally encoded message with a rate of 1/2 and a constraint length of 7.

Four C/N_0 , of 10, 15, 18 and 20 dB-Hz, are tested. Each test is run for 600 seconds. The performance is compared with that of the L1 C/A signal using the algorithms developed in [10], chapter 5. The carrier tracking is tested using two integration lengths of 20 ms and 80 ms. The results of the C/A tracking are obtained using 20 ms integration. Figure 4 shows the standard deviation (SD) of the Doppler shift estimation error, using TCXO clocks, while figure 5 shows the SD using the higher quality OXO clock.

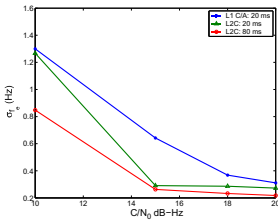


Figure 4. Standard deviation of the Doppler shift estimation error vs. C/N_0 , using TCXO clock.

Figure 6 shows the SD of the phase es-

timization error, using TCXO and OXO clocks. As can be seen, the L2C tracking gave more accurate estimation than the L1 C/A tracking. This is partly because both the data-less CL signal and the data modulated CM signal are used in the L2C tracking, while the L1 C/A signal does not have a data-less portion. In both L2C and L1 C/A tracking, the better quality OXO clock gave a remarkably better performance. Figure 7 shows the Doppler shift tracking error history for the 15 dB-Hz signal, with the low quality TCXO clock, during the first 180 seconds of operation. The upper graph is generated with 20 ms integration, while the lower graph is generated with 80 ms integration. The \pm square root of the covariance, σ , of EKF is also shown. Figure 8 shows the phase tracking error history for the same test cases, where the upper graph is generated with 20 ms integration, while the lower graph is generated with 80 ms integration.

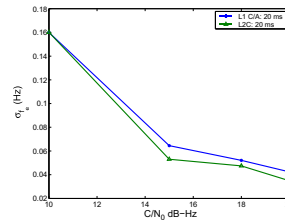


Figure 5. Standard deviation of the Doppler shift estimation error vs. C/N_0 , using OXO clock.

The code tracking is tested using two integration lengths of 20 and 200 ms. Two code delay separations of $1/f_s$ and $2/f_s$ are used simultaneously. The results are compared with the L1 C/A signal using the larger integration length, i.e. the 200 ms integration. Figure 9 shows the SD of the code delay estimation error, using TCXO and OXO clocks. As can be seen, the 200 ms integration of the L1 C/A signal gave comparable performance to the L2C signal when using only 20 ms integration. The L2C signal gave a significantly more accurate delay estimation when using the same integration length as the L1 C/A signal, 200 ms. This is because of the two longer codes of the L2C signal (the CM code is 10 times longer than the C/A code).

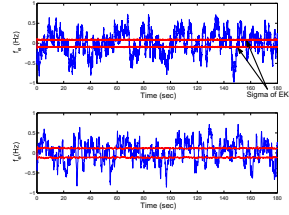


Figure 7. Doppler shift tracking error history and \pm sigma of EKF for C/N_0 of 15 dB-Hz, using TCXO clock.

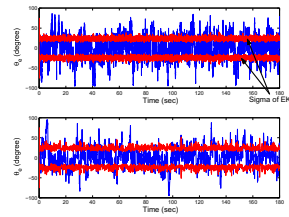


Figure 8. Phase tracking error history and \pm sigma of EKF for C/N_0 of 15 dB-Hz, using TCXO clock.

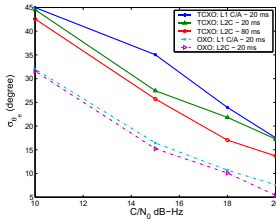


Figure 6. Standard deviation of the phase estimation error vs. C/N_0 , using TCXO and OXO clocks.

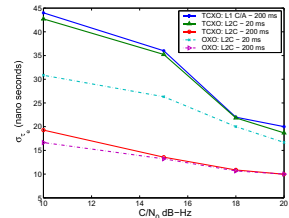


Figure 9. Standard deviation of the code delay estimation error vs. C/N_0 , using TCXO and OXO clocks.

8 SUMMARY AND CONCLUSIONS

This paper presented code and carrier tracking, and CNAV message decoding algorithms for the new L2C GPS signal to work with weak signals. The tracking algorithms were based on EKF. A square root EKF was used with the carrier tracking to achieve a stable performance, while a second order EKF was used with the code tracking to achieve high accuracy. The CNAV decoding algorithm was based on the Viterbi algorithm, and it decoded the convolutionally encoded message.

Several capabilities were implemented in the tracking algorithms to increase the time to lose lock and achieve high performance. Long coherent integration was used, and the effect of high BER was minimized by weighting the estimated data bits based on their likelihood before incorporating them into the EKF equations. A performance monitoring mechanism was implemented to detect large code and carrier errors. A code acquisition-like module and carrier re-initialization algorithms were implemented to refine the large errors, and avoid losing lock and having to reacquire the signal, which wastes time and resources. The decoded message was re-encoded and passed on to the tracking modules to be used as an aid to remove the effect of the data signs.

The algorithms were tested for C/N_0

in the range of 10–20 dB-Hz, and the results were compared with that of the L1 C/A signal. The results gave promising performance for such weak signals, and provided higher estimation accuracy than those of the L1 C/A signal.

REFERENCES

- [1] R.D. Fontana, T.A. Stansell, and W. Cheung. The Modernized L2 Civil Signal. *GPS World*, September 2001.
- [2] R.D. Fontana, W. Cheung, P.M. Novak, and T.A. Stansell. The New L2 Civil Signal. In *Proc. ION GPS*, pages 617–631, Salt Lake City, UT, September 11–14 2001.
- [3] R. F. Leandro, R. B. Langley, M. C. Santos, L. Skeov, and T. Thirumurthi. The GPS L2C Signal - A Preliminary Analysis of Data Quality. *GPS World*, 18(8):42–47, October 2008.
- [4] *Interface Specifications (IS) GPS 200D*. NAVSTAR GPS, March 2006.
- [5] <http://www.insidegnss.com/node/914>. Air Force Will Add Navigation Message to L2C Signal in Fall 2009, November 3 2008.
- [6] B. Parkinson and J. Spilker. *Global Positioning System: The*

- ory and Applications. AIAA, 1996.
- [7] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, Second edition, 2006.
- [8] J. B. Y. Tsui. *Fundamentals of Global Positioning System Receivers: A Software Approach, Second Edition*. Wiley-Interscience, 2004.
- [9] M. L. Psiaki and H. Jung. Extended Kalman Filter Methods for Tracking Weak GPS Signals. In *Proc. ION GPS*, Portland, OR, September 24–27, 2002.
- [10] N. I. Ziedan. *GNSS Receivers for Weak Signals*. Artech House, Norwood, MA, July, 2006.
- [11] M. L. Psiaki, T. E. Humphreys, A. P. Cerruti, S. P. Powell, and P. M. Kintner. Tracking L1 C/A and L2C Signals through Ionospheric Scintillations. In *ION GNSS*, pages 246–268, September 25–28 2007.
- [12] K. Muthuraman, R. Klukas, and G. Lachapelle. Performance Evaluation of L2C Data/Pilot Combined Carrier Tracking. In *ION GNSS*, pages 1658–1666, September 16–19 2008.
- [13] B. M. Ledvina, M. L. Psiaki, S. P. Powell, and P. M. Kintner. Real-Time Software Receiver Tracking of GPS L2 Civilian Signals using a Hardware Simulator. *ION GNSS*, pages 1598–1610, September 13–16 2005.
- [14] T. Cormen, C. Leisersen, R. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Third edition, 2009.
- [15] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. WCB McGraw-Hill, third edition edition, 1991.
- [16] P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 1. Academic Press, 1979.
- [17] P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 2. Academic Press, 1982.
- [18] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. J Wiley, 1992.
- [19] A. J. Van Dierendonck, J. B. McGraw, and R. G. Brown. Relationship Between Allan Variances and Kalman Filter Parameters. In *Proc. 16th PTTI Application and Planning Meeting, NASA Goddard Space Flight Center*, pages 273–293, November 27–29, 1984.