Received: 11 May 2023 | Revised: 18 January 2024 | Accepted: 17 February 2024

DOI: 10.33012/navi.646

ORIGINAL ARTICLE

Adaptive Multiple-Model Kalman Filter for GNSS Carrier Phase and Frequency Estimation Through Wideband Interference

Wengxiang Zhao[1] | **Samer Khanafseh**[1] | **Boris Pervan**[1]

[1] Illinois Institute of Technology, IL, USA

Correspondence
Wengxiang Zhao
10 W 32nd St, Chicago, IL 60616. Email: wzhao42@hawk.iit.edu

Abstract

Interference events, both intentional and unintentional, are significant threats to global navigation satellite system (GNSS) service continuity. In the presence of interference, it can be difficult for GNSS receivers to maintain continuous tracking of carrier phase and frequency. To address this issue, in place of a traditional phase-locked loop (PLL), we develop and validate a new estimation-based approach to allow GNSS receivers to correctly estimate carrier phase and frequency under conditions of low carrier-to-noise power. An adaptive Kalman filter forms the core of the estimator, with a multiple-model algorithm to account for discrete navigation data bit transitions. The performance of the estimator in the presence of wideband interference is validated through simulation and experiment, showing performance markedly superior to that of a PLL.

Keywords

adaptive Kalman filter, GPS carrier phase estimation, multiple-model estimator, wideband radio frequency interference

1 | INTRODUCTION

Jamming events on satellite navigation services, deliberate or unintentional, are becoming more frequent. The Newark airport in August 2013, Denver airport in January 2022, and Dallas Fort Worth airport in October 2022 suffered different levels of radio frequency interference (RFI) jamming and Global Positioning System (GPS) service discontinuity. More than 20 GPS service interruptions are reported every day on the U.S. Coast Guard Navigation Center website (https://www.navcen.uscg.gov). The situation is perhaps not surprising, given that electronic systems and devices that can generate RFI, including commercial products specifically designed for this purpose, are readily accessible to anyone.

Carrier tracking in global navigation satellite system (GNSS) receivers is especially vulnerable to RFI. The carrier tracking function is generally implemented via a phase-locked loop (PLL) to reconstruct the received carrier signal. A PLL uses a phase discriminator and loop filter to continuously measure and compensate for the phase difference between the local signal replica and the incoming carrier signal. The PLL also produces the carrier phase range measurements that are needed in most modern navigation applications. During an RFI event, the additive noise pumped into the PLL degrades the the ability of the phase discriminator to measure the true carrier phase, which, in turn, causes an accumulated error in the carrier reconstruction and eventually a loss of phase lock.

Typical methods for PLL tracking of weak signals include extending coherent averaging times (T_{av}) and tightening noise bandwidths. With the use of large T_{av} values (for example, greater than 20 ms for GPS signals), in-phase (I) and quadrature (Q) signal samples must be integrated across navigation data bit transitions. Existing techniques for extending the averaging time over multiple data bits, such as non-coherent memory discriminators (Borio & Lachapelle, 2009) and real-time bit estimation techniques (Stevanovic & Pervan, 2017), introduce biases in the discriminator output in the presence of strong RFI, which, in turn, lead to biases in the reconstructed carrier frequency and ultimately a loss of phase lock.

Tightening the noise bandwidth can be effective in reducing the tracking error variance, given a linear classic control PLL model, but it does not help to reduce the discriminator output variance (Stevanovic & Pervan, 2017). In the presence of strong RFI, the discriminator output can saturate, making the PLL nonlinear and leading to instability and loss of lock.

The PLL is fundamentally a feedback control system, where the I and Q samples, as interpreted by the discriminator, serve as the sensor (measurement) and the loop filter serves as the compensator (controller). However, the feedback control model is not the only approach for addressing the carrier "tracking" problem. As first proposed by Psiaki & Jung (2002) and advocated more recently by Vila-Valls et al. (2017), this problem can also be understood as an estimation problem amenable to Kalman filtering.

Kalman filter implementations are more flexible than PLLs because their component dynamic and measurement models can be designed to suit the needs of specific implementation scenarios. Kalman filters are also optimal estimators for applications with Gaussian input noise, which is the case for wideband RFI. In earlier work, Kalman filters were used to estimate carrier phase through ionospheric scintillation with $T_{av} < 20$ ms by Vila-Valls et al. (2015) and Humphreys et al. (2010) in simulation tests and by Humphreys et al. (2005) and Vila-Valls et al. (2020) in limited experiments. A Kalman filter architecture was also used by Psiaki & Jung (2002) to track numerically simulated weak GPS signals.

A major challenge in using a Kalman filter for GNSS carrier phase "tracking" is that this tracking corresponds to a hybrid stochastic estimation problem, requiring simultaneous estimation of discrete navigation data bits and continuous carrier phase. For the case in which data bits are completely unknown, Psiaki & Jung (2002) addressed the bit transition using a Bayesian bit estimation technique in which equal *a priori* probabilities were assumed for each new data bit. Of course, when the receiver is operating under low signal strength conditions, the data bits are difficult to distinguish; thus, the authors also provided a means for incorporating *a priori* data bit knowledge. For the limiting case in which all data bits are assumed to be known, the problem collapses to assisted GNSS (A-GNSS) (Morton et al., 2021), which makes use of broadcast navigation data from an external source to remove the data bits. In this case, the receiver must obtain new data whenever there is an ephemerides cutover or new navigation data upload, which, in the case of GPS operations, can happen at any time. As we will show, the use of incorrect bits can result in a loss of phase lock and navigation continuity.

Properly utilizing previously decoded navigation data and maintaining an awareness of any ephemeris cutovers and uploads can significantly improve tracking performance. We analyze multiple days of GPS navigation data transitions to develop a data bit prediction strategy that covers any navigation data change. Understanding that navigation data changes happen infrequently for GPS, we then introduce a detection method for such changes to enable adaptive updates of *a priori* bit probabilities when needed, while simultaneously maximizing the use of data bit knowledge for cases in which a navigation data transition has not occurred.

In the latter case, it is possible to have an averaging time longer than 20 ms, but we limit ourselves to $T_{av} = 20$ ms because this setting is also applicable to the case in which data bit changes have been detected. Further, the Kalman filter uses I and Q measurements directly, unlike a PLL, where it is necessary to average out noise prior to the construction of a nonlinear discriminator. The Kalman filter itself effectively "extends" the averaging time to the level permitted by oscillator phase noise and, in the case of moving users, inertial random walk errors. Additionally, longer averaging times result in one carrier frequency output as an average value over the entire period, leading to larger biases in carrier frequency estimation as the averaging period becomes longer.

The applications we are initially targeting are aviation operations threatened by unpredictable wideband RFI events. Currently, there is no reliable method for supplying navigation data bits from an external source to aircraft or related GNSS ground-based augmentation systems (GBASs). We do not address the problem of signal acquisition during interference because our goal is to continue tracking existing satellites through RFI events of limited duration, not to start up and operate continuously in low-signal-strength environments (e.g., indoors with A-GNSS). However, we do leverage the fact that the broadcast ephemeris decoded prior to the onset of RFI, except for events of especially long duration, will still be valid for satellite position and clock determination during the event, corresponding to at least 2 h in the case of GPS (Dunn, 2013).

In our prior work (Zhao & Pervan, 2019), we performed preliminary simulation trials to study the feasibility of a multiple-model (MM, special case of IMM, interacting multiple model) Kalman filter approach for carrier phase estimation, with encouraging results for a static phase and different levels of knowledge of navigation data bits. Additionally, we showed experimental results under interference-free conditions (Zhao & Pervan, 2020b). The MM filter was able to generate the same carrier phase estimates as the PLL in scenarios with nominal signal strength. We also performed experiments under interference, with results showing superior performance of the MM filter over the PLL (Zhao & Pervan, 2020a). When strong wideband RFI was present, the MM filter produced correct carrier phase estimates, whereas the PLL could not track the signal.

In this paper, we provide a review of our prior work (Zhao et al., 2021; Zhao & Pervan, 2019, 2020a, 2020b), present new experimental results quantifying performance for different interference powers, and implement and assess the effectiveness of data bit prediction and navigation data transition detection via the MM approach to further boost estimator performance during wideband RFI events.

Presently, we consider an example application in a GBAS reference station receiver subjected to broadband interference, for instance, from nearby use of personal privacy devices (Pullen & Gao, 2012). However, the developed methods can, and will in future work, also be applied to moving GPS reference receivers with inertial aiding.

In Section 2, the MM algorithm and its Kalman filter components are described. Section 3 presents and discusses simulation results for the MM estimator. Section 4 describes the experimental scenario and setup. Section 5 presents experimental results under normal and interference conditions. Section 6 shows experimental results for the MM estimator with navigation data bit prediction. Section 7 quantifies the probability of detection of new navigation data uploads. In Section 8, we summarize the results and share ideas for future research.

2 | KALMAN FILTER AND MM ALGORITHM

Whereas PLLs are typically fixed structures with predefined discriminators and loop filters, Kalman filters have internal adaptability to rely more heavily on either measurements or phase dynamics depending on the noise levels. Extended coherent averaging times are needed to track through strong interference. If no external information about the navigation data bits is available, the maximum coherent integration time for GPS is 20 ms, which is the duration of one navigation data bit. However, if achievable, even longer coherent integration times could help to significantly average out noise, resulting in cleaner I and Q measurements.

The carrier tracking loop updates the local carrier frequency using I and Q samples averaged over each successive coherent integration window. Wideband interference events contribute additive white Gaussian noise directly to the I and Q measurements. For typical PLL implementations using phase discriminators, these noisy measurements can easily cause phase errors exceeding the pull-in limit of the discriminator, which usually leads to cycle slips and an eventual loss of lock.

The shortcomings of a traditional PLL can be overcome by using a Kalman filter to estimate the carrier frequency and phase directly. The goal in carrier tracking is to produce the best phase and frequency estimates under noisy conditions, which the Kalman filter can optimally accomplish given that the noise is white (which is the case for wideband interference). Figure 1 shows a top-level block diagram of an IMM/Kalman-based carrier phase tracking architecture as it would be implemented in a software-defined receiver (SDR).

In our interference scenario, we assume that the receiver is already tracking, post-acquisition, before being subjected to an RFI event, and as a result, the Kalman filter estimate error is small immediately prior to the event onset.

2.1 | Dynamic Model

Our goal is to estimate the carrier phase and frequency in real time. Over one pre-detection averaging interval, T_{av}, the total phase change is $\phi_{tot} = \phi_{Dopp} + \phi_{clk}$, where ϕ_{Dopp} is the phase change due to relative movement between the satellite and receiver. For our current development, we assume a static receiver, e.g., a GBAS reference receiver, such that ϕ_{Dopp} is the result of satellite motion only, which is known and removed by using the last decoded broadcast ephemeris. (In future work, user motion can be accounted for by using inertial sensors.) Therefore, we need only a clock phase dynamic model for our Kalman filter.

The power spectral density (PSD) for clock phase noise can be expressed by the conventional power law model:

$$S(f) = \frac{f_0^2}{f^2}\left(h_2 f^0 + h_1 f^{-1} + h_0 f^{-2} + h_{-1} f^{-3} + h_{-2} f^{-4}\right) \qquad (1)$$

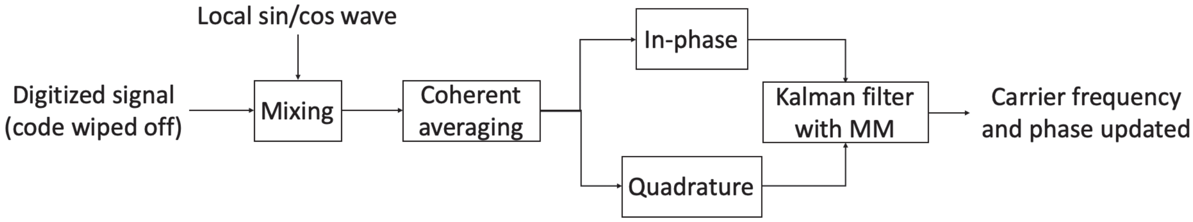

FIGURE 1 Top-level view of MM/Kalman-based carrier phase estimation in an SDR

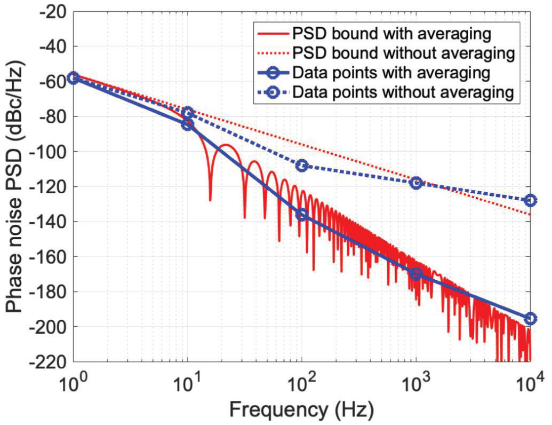

FIGURE 2 Clock phase noise PSD bounding (SpectraTime LPFRS)

The values of the h coefficients will depend on the specific receiver and satellite clocks.

As an example, in the experimental and simulation work described later in this paper, we use independent SpectraTime Low Cost & Profile Frequency Rubidium Standards (LPFRS) for both the receiver and satellite signal simulator. The clock product specifications provide phase noise data points at 1, 10, 100, 1000, and 10000 Hz as –80, –100, –130, –140, and –150 dBc/Hz, respectively. Using the coefficients $h_2 = 0$, $h_1 = 0$, $h_0 = 1.241 \times 10^{-6}$, $h_{-1} = 0$, and $h_{-2} = 2.4819 \times 10^{-12}$, we upper bound the specifications, as shown in Figure 2.

In the figure, the solid lines represent the PSDs after pre-detection averaging with $T_{av} = 20$ ms, and the dashed lines show the results obtained before averaging. The blue circles are data points from the clock phase noise PSD specifications.

For Kalman filter implementation, we now define a state vector to include the signal amplitude (A), clock phase (ϕ_{clk}), and clock frequency (f_{clk}). We also define the following dynamic process model:

$$\begin{bmatrix} A \\ \phi_{clk} \\ f_{clk} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A \\ \phi_{clk} \\ f_{clk} \end{bmatrix}_{k} + w_k \tag{2}$$

where $\Delta t = T_{av}$ is the time increment represented by index k and $w_k \sim N(0, W)$ is time-independent white noise. The process noise covariance matrix is as follows (Chan et al., 2014):

$$W = \begin{bmatrix} \delta^2 & 0 & 0 \\ 0 & S_f \Delta t + \frac{S_g \Delta t^3}{3} & \frac{S_g \Delta t^2}{2} \\ 0 & \frac{S_g \Delta t^2}{2} & \frac{S_f}{\Delta t} + \frac{4}{3} S_g \Delta t \end{bmatrix} \tag{3}$$

where:

$$S_g = 2\pi^2 h_{-2} \tag{4}$$

$$S_f = \frac{h_0}{2} \tag{5}$$

f_0 is the carrier frequency (e.g., $f_0 = f_{L1} = 1575.42$ MHz for the GPS L1 signal), and $\delta^2 \approx 0$ is included to allow for small nominal process noise on the amplitude state.

2.2 | Measurement Model

The measurement model, with the known contributions of ϕ_{Dopp} to the I and Q components removed, is as follows:

$$I_k = d_k A_k \cos(\phi_{clk,k}) + v_{i,k} \tag{6}$$

$$Q_k = d_k A_k \sin(\phi_{clk,k}) + v_{q,k} \tag{7}$$

where $d_k = \pm 1$ is the navigation data bit and $v_{i,k}$ and $v_{q,k}$ are white measurement processes distributed as $[v_{i,k}\ v_{q,k}]^T \sim N(0, V_k)$ with $V = I\sigma^2_{v,k}$. The measurement error variance $\sigma^2_{v,k}$ for a unit amplitude signal is related to the carrier-to-noise ratio $(C/N_{0,k})$ by the following:

$$C/N_{0,k} = 10^{\frac{C/N_{0,k}[\text{dBHz}]}{10}} \tag{8}$$

$$\sigma^2_{v,k} = \frac{1}{2T_{av} C/N_{0,k}} \tag{9}$$

The observation equations (Equations (6) and (7)) are nonlinear in state $\phi_{clk,k}$; thus, for the Kalman measurement update to be conducted, the equations must be linearized about the best available estimate, $\hat{\phi}_{clk,k|k-1}$, obtained from the previous dynamic update.

2.3 | MM Algorithm

The MM algorithm is a dynamic multiple-hypothesis estimator. It assumes that the system obeys one of a finite number of continuous models at a time but is capable of switching between models at discrete intervals. The algorithm has been described comprehensively by Bar-Shalom et al. (2004); thus, only a brief summary is given here, with additional clarifications in the Appendix and with an emphasis on application to the problem at hand—GNSS data bit transitions.

A flowchart of an example two-hypothesis MM estimator is shown in Figure 3. At the top, prior state estimate vectors $x_1(k-1|k-1)$ and $x_2(k-1|k-1)$ and their corresponding error covariance matrices $P_1(k-1|k-1)$ and $P_2(k-1|k-1)$ are obtained from two component Kalman filters executed in the previous cycle. These are input into a "mixing" function to compute modified inputs for the next Kalman filter cycle: $x_{1m}(k-1|k-1)$, $P_{1m}(k-1|k-1)$ and $x_{2m}(k-1|k-1)$, $P_{2m}(k-1|k-1)$. Each Kalman filter represents a specific mode, which may differ from each other in the dynamic model or measurement model (or both). Using its own measurement model, each filter will then perform a measurement update using the same measurement $z(k)$ and a time update with its own dynamic model. Likelihood functions $\Lambda_1(k)$ and $\Lambda_2(k)$ are then evaluated and used to calculate current mode-state (μ_k) and mixing $(\mu_{k|k})$ probabilities, with the latter used to commence the next cycle. The Appendix provides additional details. The output state estimate vector and error covariance matrix, $x(k|k)$ and $P(k|k)$, are computed using the individual Kalman filter results $x_1(k|k)$, $P_1(k|k)$ and $x_2(k|k)$, $P_2(k|k)$ and the mode probability vector μ_k.

In our MM application, two modes are run in parallel, as shown in Figure 4, corresponding to two measurement models with navigation data bit values of $d = 1$

and $d = -1$. The Kalman measurement updates for both modes are performed using the same I and Q measurements. The dynamic models are also the same for both modes, as defined in Section 2.1.

For now, we assume that the data bits are sequentially independent, the mode (bit) transition probabilities are both $\frac{1}{2}$, and the mode transition matrix is a 2×2 matrix with each element equal to $\frac{1}{2}$. The "mixing" process shown in Figure 3 for the standard MM algorithm is not needed in this case because of the uniform structure of the mode transition probability matrix. The proof is provided in Appendix A. The result, as illustrated in Figure 4, is that the final (fused) output state and covariance can be fed directly into both mode filters at the next time step. However, the MM approach can also accommodate more general cases, where there is reduced uncertainty in bit transitions, as will be discussed in Section 6.

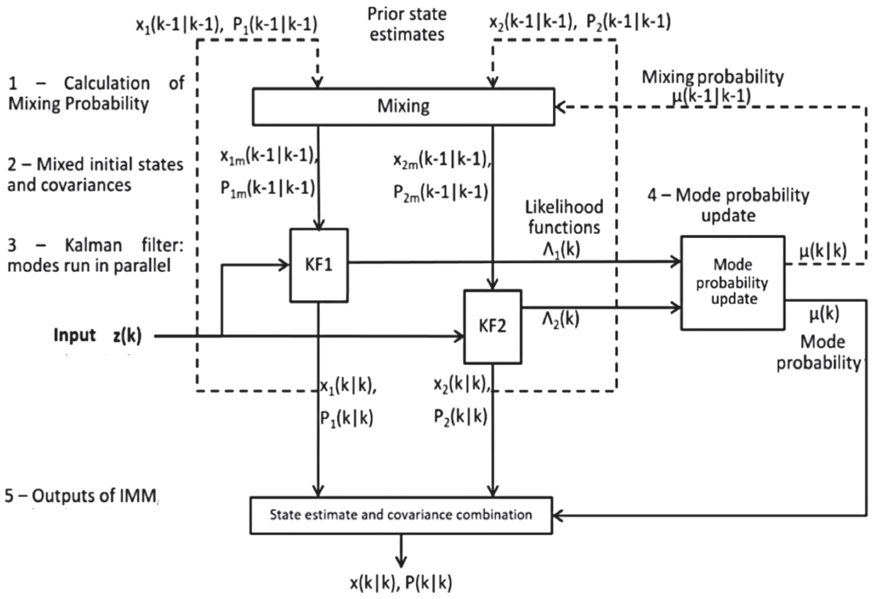
$x_1(k-1|k-1)$, $P_1(k-1|k-1)$
Prior state estimates
$x_2(k-1|k-1)$, $P_2(k-1|k-1)$
1 – Calculation of Mixing Probability
Mixing
Mixing probability $\mu(k-1|k-1)$
2 – Mixed initial states and covariances
$x_{1m}(k-1|k-1)$, $P_{1m}(k-1|k-1)$
$x_{2m}(k-1|k-1)$, $P_{2m}(k-1|k-1)$
Likelihood functions
4 – Mode probability update
3 – Kalman filter: modes run in parallel
KF1
$\Lambda_1(k)$
Mode probability update
$\mu(k|k)$
Input z(k)
KF2
$\Lambda_2(k)$
$\mu(k)$
Mode probability
$x_1(k|k)$, $P_1(k|k)$
$x_2(k|k)$, $P_2(k|k)$
5 – Outputs of IMM
State estimate and covariance combination
x(k|k), P(k|k)

FIGURE 3 MM algorithm; KF: Kalman filter

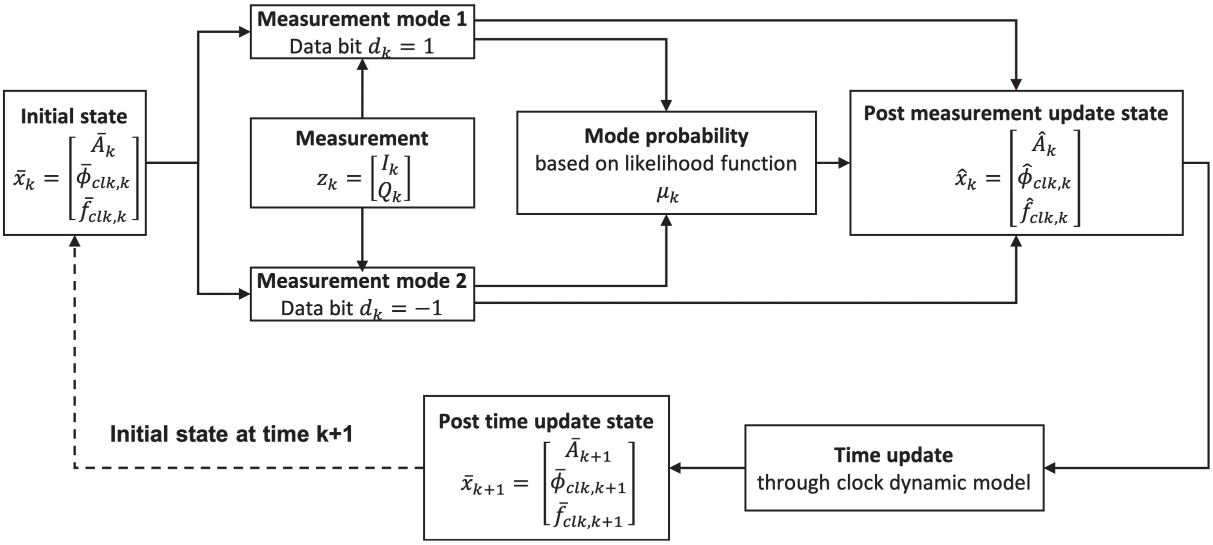
Measurement mode 1
Data bit $d_k = 1$
Initial state
$\bar{x}_k = \begin{bmatrix} \bar{A}_k \\ \bar{\Phi}_{clk,k} \\ \bar{f}_{clk,k} \end{bmatrix}$
Measurement
$z_k = \begin{bmatrix} I_k \\ Q_k \end{bmatrix}$
Mode probability
based on likelihood function
μ_k
Post measurement update state
$\hat{x}_k = \begin{bmatrix} \hat{A}_k \\ \hat{\Phi}_{clk,k} \\ \hat{f}_{clk,k} \end{bmatrix}$
Measurement mode 2
Data bit $d_k = -1$
Initial state at time k+1
Post time update state
$\bar{x}_{k+1} = \begin{bmatrix} \bar{A}_{k+1} \\ \bar{\Phi}_{clk,k+1} \\ \bar{f}_{clk,k+1} \end{bmatrix}$
Time update
through clock dynamic model

FIGURE 4 MM algorithm in our case

3 | SIMULATION

In our simulation tests, we integrated the measurement models and the dynamic model into the component Kalman filters to evaluate the carrier frequency estimation performance of the MM algorithm. We generated "true" carrier phase and frequency time profiles as well as random data bits. We also added white noise into the true signal to simulate the effect of an interference event. The initial interference-free signal carrier-to-noise ratio was set to 51 dB-Hz, and we evaluated the effects of subsequent interference events that reduced C/N_0 to 25 dB-Hz and 15 dB-Hz.

We started the simulation with a steady-state covariance matrix corresponding to a high initial signal power. A 20-ms coherent averaging time was applied throughout the simulation, and the total simulation duration was 4 min. The interference events started at 110 s for all scenarios. The results of the simulation were captured as carrier phase and frequency estimation errors between the MM Kalman filter output and the corresponding true inputs.

Figures 5 and 6 show results for the case with 25-dB-Hz interference, and Figures 7 and 8 present results for the 15-dB-Hz case. The resulting phase and frequency errors have a zero mean throughout the run. After the interference onset,

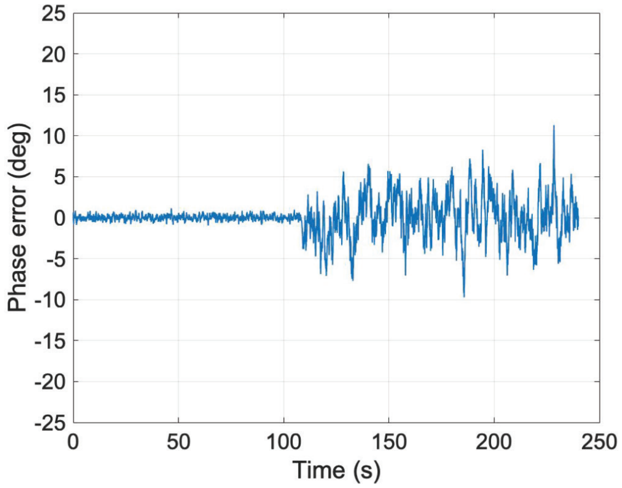

FIGURE 5 Phase estimation error for the 25-dB-Hz simulation

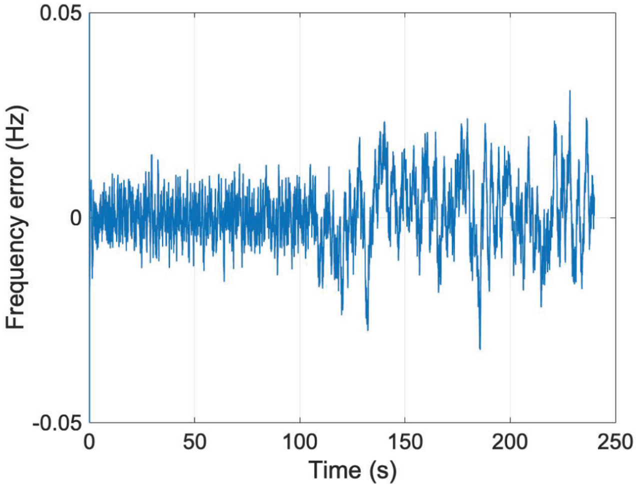

FIGURE 6 Frequency estimation error for the 25-dB-Hz simulation

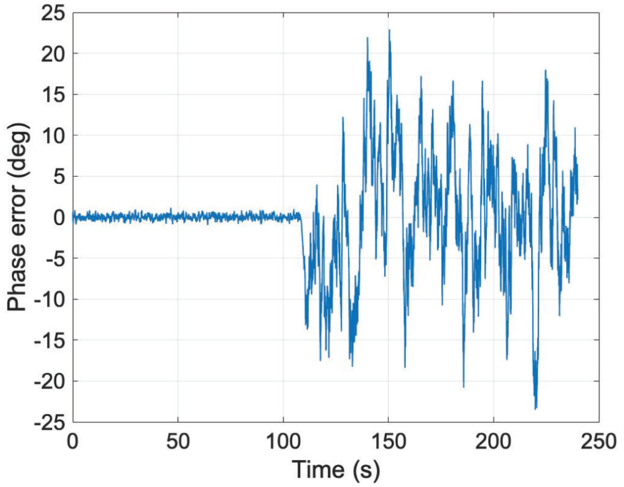

FIGURE 7 Phase estimation error for the 15-dB-Hz simulation

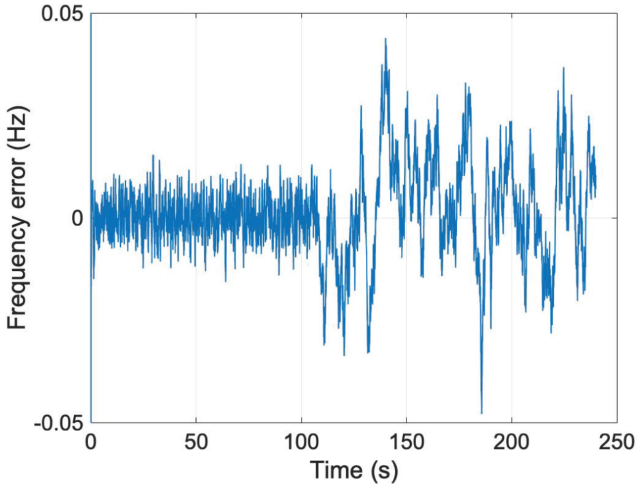

FIGURE 8 Frequency estimation error for the 15-dB-Hz simulation

the estimator-predicted standard deviations of phase and frequency error increase but remain within a controlled range: 2.73° and 0.0097 Hz for the phase and frequency error, respectively, for the 25-dB-Hz case and 7.96° and 0.0142 Hz for the phase and frequency error for the 15-dB-Hz case. In both cases, the computed sample deviations were nearly identical to the predicted values derived by the covariance analysis of Zhao et al. (2021).

4 | EXPERIMENTAL SCENARIO AND SETUP

4.1 | Experimental Setup

GPS radio frequency signals were generated using a Spectracom GSG-6 GNSS simulator. Two LPFRS clocks were used as time references for the satellites and receiver. A Universal Software Radio Peripheral (USRP) N200 was used as the radio frequency front-end to collect the sampled data for post processing. The IMM estimator was implemented in an SDR. Figure 9 shows an overview of the experimental setup.

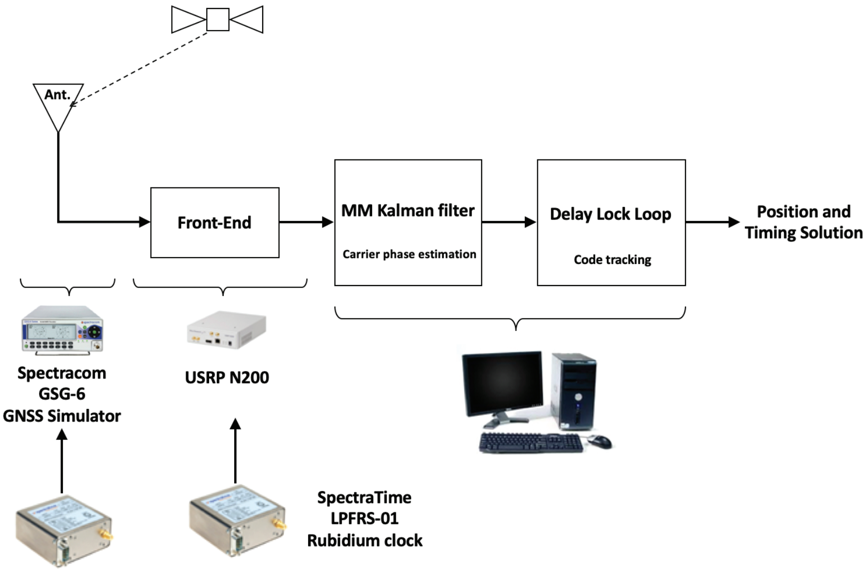
Ant.
Front-End
MM Kalman filter
Carrier phase estimation
Delay Lock Loop
Code tracking
Position and Timing Solution
Spectracom GSG-6 GNSS Simulator
USRP N200
SpectraTime LPFRS-01 Rubidium clock

FIGURE 9 Experimental setup

TABLE 1
PRN C/N_0 Values in the Experimental Test

	PRN 1	PRN 30	PRN 7	PRN 11	PRN 28
Before interference	51 dB-Hz	51 dB-Hz	51 dB-Hz	51 dB-Hz	51 dB-Hz
During interference	51 dB-Hz	45 dB-Hz	35 dB-Hz	25 dB-Hz	15 dB-Hz

4.2 | Experimental Scenario

The experiment involved a static GPS reference receiver operating over a 4-min period with five satellites (pseudorandom noise [PRN] 1, 7, 11, 28, 30). The normal signal strength prior to the onset of the RFI event was set to 51 dB-Hz. The onset of the RFI event was 2 min after the beginning of the experiment so that the receiver had sufficient time for acquisition and for locking on to the nominal signals.

The interference event was instigated by reducing C/N_0 from 51 dB-Hz (which was the starting signal strength for all five satellites) as follows: no drop for PRN 1, 6-dB-Hz drop for PRN 30, 16-dB-Hz drop for PRN 7, 26-dB-Hz drop for PRN 11, and 36-dB-Hz drop for PRN 28. These scenarios are described in Table 1.

5 | ESTIMATOR PERFORMANCE

The MM carrier frequency estimate errors were compared with those of a traditional PLL. For the PLL, the carrier frequency estimate error is the same as the frequency tracking error because the PLL does not have an “estimation” process.

To make as fair a comparison as possible, we also applied traditional interference mitigation methods for the PLL, including tightening the noise bandwidth and setting the pre-detection averaging time to the maximum of 20 ms (the MM filter and the PLL were both running with a 20-ms averaging time).

Preliminary experimental results showed that rapid tightening of the PLL bandwidth after acquisition caused the PLL to become unstable. Thus, the PLL noise bandwidth was tightened gradually, starting at 15 Hz and ending at 1 Hz, before

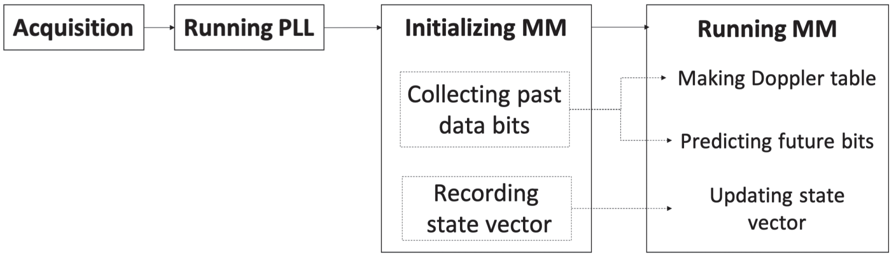

FIGURE 10 MM filter running logic

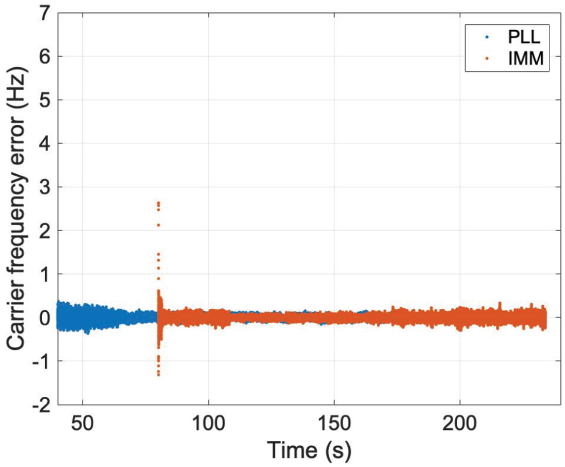

FIGURE 11 Carrier frequency estimation error for PRN 1

the MM filter was initialized. Bandwidths lower than 1 Hz resulted in unstable PLL performance.

The receiver started without interference for acquisition and initial PLL tracking. During the initial tracking period, information useful to the MM estimator was collected in preparation for the start of the MM filter. This information is shown in the box in Figure 10, including the past navigation data bits and the initial conditions of the MM filter state vector.

The past navigation data bits were used to generate a Doppler table to remove satellite motion from the received carrier frequency and to (partially) predict navigation data bits (used later in Section 6). The initial state vector included the signal amplitude (calculated from $\sqrt{I^2+Q^2}$), clock frequency (calculated by subtracting the satellite motion from the PLL carrier frequency), and clock phase (calculated by integrating the clock frequency).

In the timelines shown in the results, the MM filter was initialized at 80 s, and the interference event onset was at 110 s. After initialization, the MM estimator and the PLL ran in parallel independently to enable a comparison of their responses to the interference events.

5.1 | Validation

We first assessed the results for PRN 1, which had no interference. In this case, the MM filter was expected to have approximately the same level of carrier frequency estimation error as the PLL. Figure 11 shows the carrier frequency estimation error results for the MM filter (red) and the PLL (blue).

The results show that the tracking error of the PLL decreased slowly as we tightened the PLL noise bandwidth. At 80 s, the coherent averaging time of the PLL was increased to 20 ms, and the MM filter was initialized (with the same T_{av}). At 80 s, after a short period of MM convergence, the variances of the carrier frequency estimation error (MM) and tracking error (PLL) were nearly the same. Thus, in the interference-free case, we see that the MM filter preserves the same carrier frequency estimation ability as the traditional PLL.

5.2 | Wideband Interference

Next, we performed the same test as in Section 5.1 for the other PRNs, which experienced different levels of interference starting at 110 s. Figure 12 shows the carrier frequency estimation error for PRN 7, whose carrier-to-noise ratio decreased from 51 dB-Hz to 35 dB-Hz at 110 s. As shown in the figure, the PLL and the IMM filter operate with no interference from 80 to 110 s with similar output error variances. After the interference onset, both error variances increase, but the MM filter output shows a notably smaller increase compared with the PLL output. At 35 dB-Hz, the PLL is still functional, as its tracking error output maintains a zero mean; thus, there is no loss of lock.

Figure 13 shows the results for PRN 11, which has an interference event at 110 s in which C/N_0 is decreased to 25 dB-Hz. Beginning at 110 s, the carrier frequency tracking error of the PLL drifts away from zero (recall that we have already applied interference mitigation methods, including tightening the noise bandwidth and increasing the coherent averaging time). Because the frequency tracking error does not have a zero mean, the PLL is no longer able to maintain a lock on the carrier. In contrast, the MM filter was able to maintain a zero mean on the carrier frequency estimation error output. The frequency estimation error variance is 0.0818 Hz during the 25-dB-Hz jamming period.

Figure 14 shows the results for PRN 28, which experiences a carrier-to-noise ratio of 15 dB-Hz after interference onset at 110 s. As expected, given the results of the previous case at 25 dB-Hz, the PLL again lost lock. In contrast, the MM filter was able to produce zero-mean carrier frequency estimation results at 15 dB-Hz, albeit with a larger variance than in the 25-dB-Hz case. The standard deviation of the filter-predicted frequency estimation error was 0.1688 Hz during the 15-dB-Hz jamming period. Again, the actual sample standard deviation of the frequency

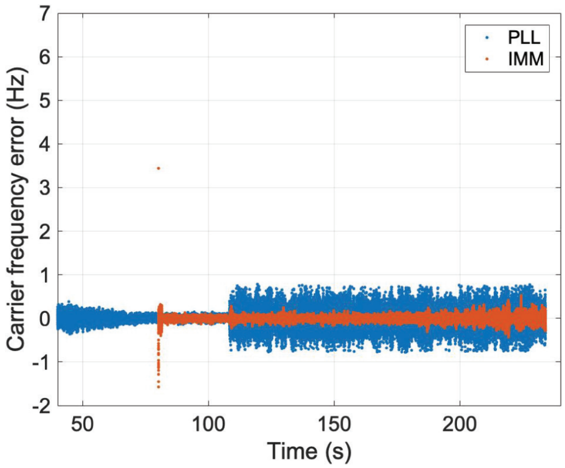

FIGURE 12 Carrier frequency estimation error for PRN 7

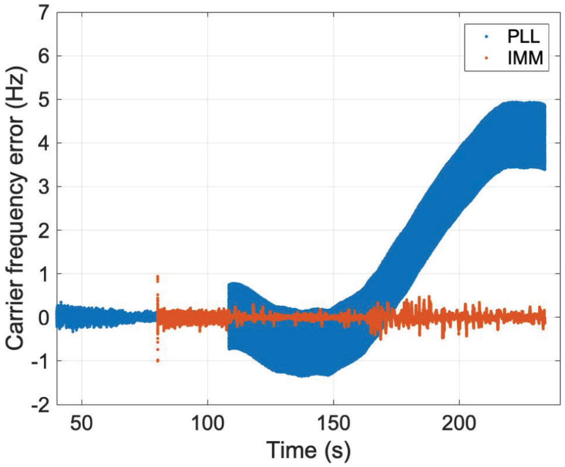

FIGURE 13 Carrier frequency estimation error for PRN 11

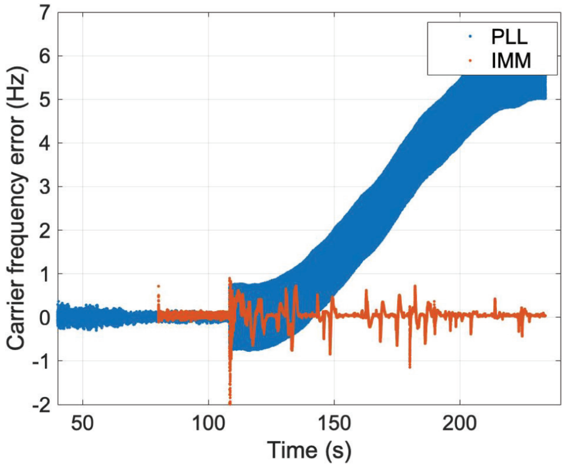

FIGURE 14 Carrier frequency estimation error for PRN 28

error output is nearly the same as the filter-predicted value for all scenarios. The frequency estimation error variance in the experimental test results is larger than the simulation value because the simulation used identical measurement error models for the filter and the random noise generation in the I and Q samples and also assumed perfect (zero-error) linearization points in the measurement model.

The results of these experiments show that the MM filter is more stable and reliable than the PLL under wideband interference. The integrated clock dynamic model and the two-hypothesis MM implementation help the estimator to distinguish data bits under noisy conditions and produce accurate estimates of carrier phase and frequency.

6 | DATA-BIT-ASSISTED ESTIMATOR

Before we can develop a strategy for data bit assistance, we must understand how the navigation data bits are updated during ephemeris cutovers and navigation data uploads. For aviation applications, which are of primary interest in this study, there is no reliable mechanism for supplying navigation data bits from an external source, as is done with A-GNSS.

In a new GPS upload, the ground control segment uploads to the satellite all elements of navigation messages for the next day (or longer). Information in all of the frames is updated, including the ephemeris, clock parameters, almanac, and so on. A new upload usually happens once a day and can happen at any time.

A GPS data cutover occurs when the satellite begins broadcasting the next segment of ephemeris data from the last upload to users. These cutovers normally happen every 2 h at the top of the hour.

6.1 | Upload-Robust Data Bit Prediction Strategy

Navigation data messages are standardized, well-structured binary bits broadcast by satellites to communicate with GPS receivers about ephemerides, almanacs, satellite health status, and other information. For this work, we focus only on the GPS L1 signal and its "legacy" navigation data bits. The navigation data message is modulated on the carrier at 50 bps and contains five subframes, each of which has 300 bits, with each bit being 20 ms in length. Subframes 4 and 5 each have 25 separate pages. At every 30 s, GPS satellites will transmit one frame: 1500 bits including subframes 1, 2, and 3 and one page each from subframes 4 and 5. Thus, collection of the whole navigation data message (superframe) takes at least 12.5 min.

The navigation data bit structures are pre-defined in IS-GPS-200H (Dunn, 2013). Some bits never change or rarely change, which opens the possibility of predicting future data bits using old data bit information. We consider the case in which a receiver decodes a sufficient number of subframes before RFI onset to enable partial data bit prediction during the event. This assumption is reasonable, given that an aircraft would not take off in the presence of RFI and that any external augmentation systems that are vulnerable to RFI operate continuously.

Using navigation data bits collected over several days, we have categorized the bits that remain predictable through ephemeris cutovers and new uploads. Some bits never change (e.g., preamble bits) or are easily predictable (e.g., time-of-week and subframe identification bits). For a number of other parameters (e.g., related to space vehicle (SV) clock corrections and certain orbit elements), the most significant bits do not change. Based on the information gained, we have developed a partial bit prediction strategy that predicts only the bits that remain unchanged through navigation data uploads. This strategy is conservative most of the time because a larger number of bits would remain predictable even through ephemeris cutovers, but it is necessary to ensure tracking continuity through uploads, which can happen at any time. The prediction strategy (i.e., exactly which bits are predicted) is detailed in Appendix B.

Figure 15 shows example data bit prediction maps for the five upcoming subframes based on the five previous subframes. The data bits from the last five subframes (50 rows × 30 columns = 1500 bits in total, each block representing one data bit) are shown on the left. The bits predictable through a subsequent upload are shown on the right. Dark blue blocks represent bit "$d = 1$," light blue blocks indicate bit "$d = -1$," and unpredictable bits are shown in grey. For predictable bits, we use a prior probability of 1 in the MM estimator; for bits deemed unpredictable, we use a prior probability of 1/2. The figure provides a rough visual idea of how many data bits are predictable. With this bit prediction strategy, an average of approximately 38% of bits are predicable over the entire set of 25 frames.

In the following section, we show results for the MM filter with and without data bit prediction. We also show results for an assisted-GPS-like prediction strategy, i.e., where all bits are assumed to be predictable through an upload.

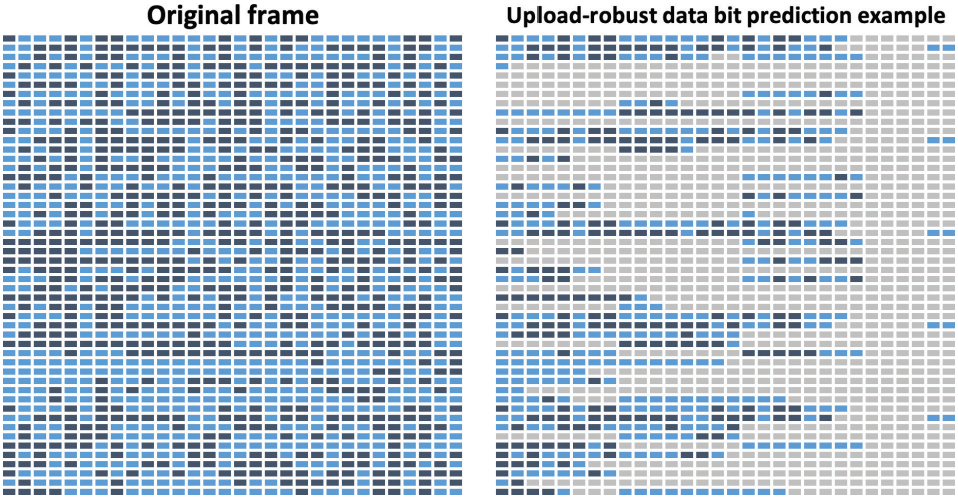

FIGURE 15 Example of data bit prediction

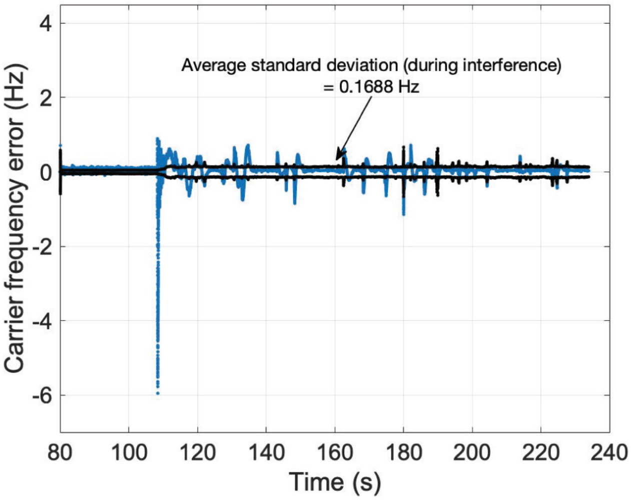

FIGURE 16 Carrier frequency estimation error for PRN 28 without bit prediction

6.2 | Data Bit Prediction Results

Figures 16 and 17 show carrier frequency estimation error results for PRN 28 (15-dB-Hz case) and the standard deviation during interference without and with upload-robust data bit prediction. Blue points represent the MM filter carrier frequency estimation error output, and black points indicate the corresponding standard deviation. Comparing the results in the two figures, we observe that the standard deviation of the carrier frequency estimate error decreases when the predictable bits are exploited. These experimental results show that modest (≈ 40%) data bit prediction can help the MM filter to improve estimation results.

After a new navigation data upload, approximately 31% of the bits change, on average. In the next test, we evaluate the performance of a 100% bit prediction strategy (e.g., A-GNSS) through a navigation data upload by deliberately seeding 31% incorrect bits that are randomly placed.

The results in Figures 18 and 19 show large estimation errors for both carrier frequency and phase during the RFI event, which starts at 100 s. We assume that a navigation data upload occurs at 100 s, but the MM filter is not aware of the upload

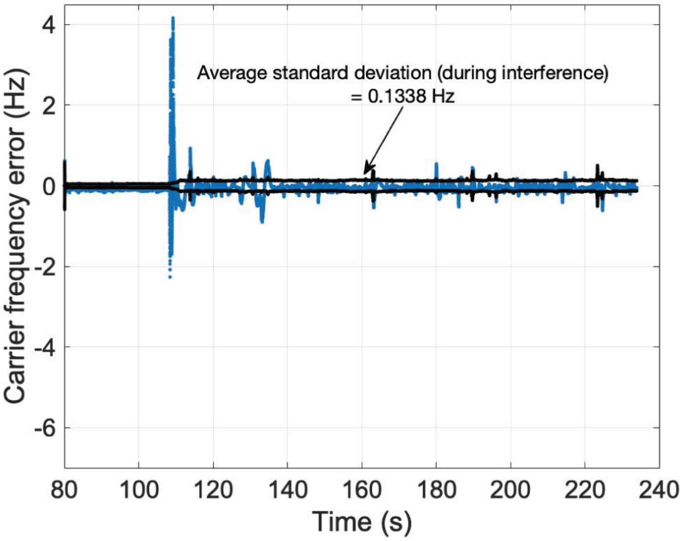

FIGURE 17 Carrier frequency estimation error for PRN 28 with upload-robust bit prediction

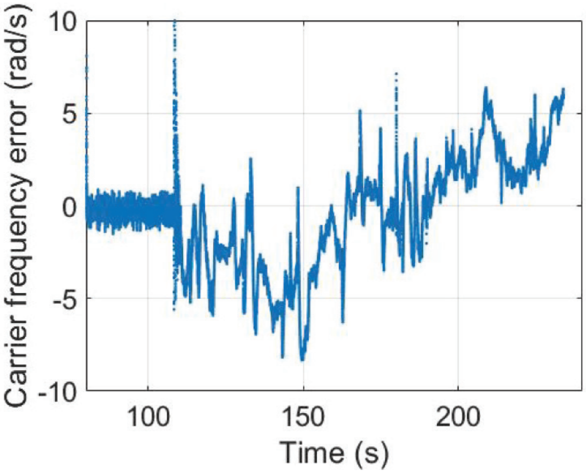

FIGURE 18 Carrier frequency estimation error for PRN 28 with 100% bit prediction attempted during upload

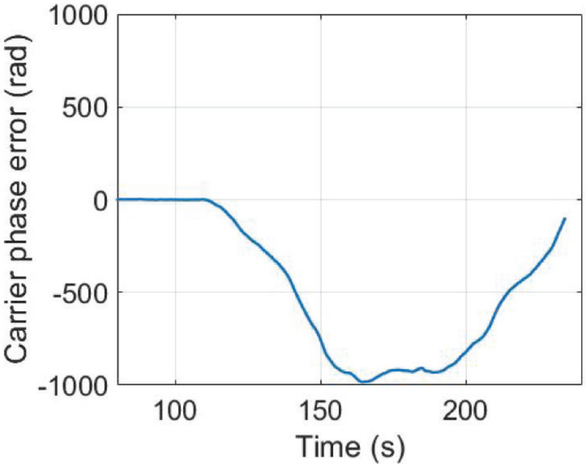

FIGURE 19 Carrier phase estimation error for PRN 28 with 100% bit prediction attempted during upload

and therefore still applies the 100% bit prediction strategy. As the results show, there is a high risk of losing continuity if the 100% bit prediction strategy (e.g., A-GNSS) is applied without any detection of ephemeris uploading.

7 | ADAPTIVE APPROACH FOR NEW UPLOADS

Using the upload-robust strategy described in Section 6.1 guarantees good performance regardless of whether there is a new upload. However, it is undeniable that further potential performance improvement is possible during periods when the data bits remain unchanged. For this reason, it is worth considering using a 100% bit prediction strategy during those times and then adaptively switching to upload-robust prediction if a new ephemeris cutover or upload is detected. This adaptability to real-time ephemeris updates enables the MM filter to optimize the utilization of existing ephemeris data while ensuring uninterrupted service continuity.

7.1 | IODE Check

Issue of data ephemeris (IODE) bits provide users a convenient way to detect any change in the ephemeris representation parameters. The IODE is a string of eight bits located in three places: word 8 of subframe 1, word 3 of subframe 2, and word 10 of subframe 3. Compared with those in the immediately previous frame, any difference in the IODE bits in a new block of subframes indicates that either an ephemeris cutover or a navigation upload has occurred. GPS ephemeris cutovers always happen on the hour, normally at every two hours; thus, if the detection does not happen on the hour, the detection signifies a new upload. There are other bits that can also serve as a means for detecting changes in navigation data, but none occurring so early in the message and so frequently (every 6 s).

Given the risk of losing continuity, as shown in the carrier phase and frequency error plots in Figures 18 and 19, where the carrier phase error increased by 40 rad (nearly seven cycles) in just 6 s, the best strategy is to use the upload-robust bit prediction until the data transition test is passed. For each subframe we receive, we perform an IODE check against the previous IODE. If the IODEs are the same, we assert that the navigation data have not changed, and we move to a 100% bit prediction strategy for the remainder of the navigation data bits. Obviously, interference events greatly increase the difficulty of correctly verifying whether the IODE bits have changed. We denote the probability of making a false claim P_f as the probability of incorrectly estimating a single bit; we will quantify this probability later in Section 7.2.

Because there are redundant IODEs available in each block of five subframes of navigation data, we will evaluate two methods for checking these IODEs. The first is a "single-check" method, in which we compare the IODE in current subframe 1 to the IODE in subframe 1 of the previous frame. The second is a "triple-check" method, in which we check all three IODEs from current subframes 1, 2, and 3 to ensure that they are the same and then compare them with the IODEs in the previous frame.

There is a trade-off between making mistakes in IODE change detection and improving performance. The single-check method allows us to apply a 100% bit prediction strategy for the remaining four subframes if no new upload is detected, but with a higher risk of losing continuity. For the triple-check method, the risk of losing continuity is lower, but the 100% bit prediction strategy can be applied only to the last two subframes.

There is little consequence in "detecting" a navigation data change when there actually is none because the data bit prediction strategy simply remains in upload-robust mode. However, the opposite is not true. If a new navigation data upload is not detected and we switch to 100% prediction mode, we can expect to lose continuity, as shown in Figures 18 and 19. However, the probability of this happening is relatively low, as there are typically only one or two new uploads per day out of 2880 frames received, which means that the probability of the current frame being from a new upload is approximately $\frac{1}{1440}$ per frame or equivalently $\frac{1}{120}$ per hour (as the frame update interval is 30 s). Therefore, the risk of losing continuity due to missing an IODE change in a new frame is approximately $\frac{1}{120}P_f$ for a single IODE check and $\frac{1}{120}P_f^3$ for a triple IODE check per hour.

7.2 | Continuity Risk Quantification

It is not easy to determine with 100% confidence that certain bits have changed during broadband interference because of the increased noise. Figure 20 shows example data bit probabilities obtained from the MM filter output for different levels of C/N_0.

The green bar represents normal operation conditions with $C/N_0 = 45$ dB/Hz, with the data bit probability samples from the MM filter output tightly lumped near 1 and 0, indicating that the MM filter can easily distinguish whether the current bits are 1 or 0. As the C/N_0 degrades to 35 dB/Hz, the blue samples shift towards the center point (50% probability) and start to overlap. When C/N_0 is lowered to 15 dB/Hz, the red samples spread out more uniformly between 0% to 100%, indicating that the MM filter has notable difficulty in determining the bits.

Consider the example I–Q scatterplot shown in Figure 21, where the amplitude of the I and Q samples is normalized to 1. In this example case, the carrier phase is set to 45°, and the red dot is the ideal I–Q sample location without noise. The blue dots are the simulated I and Q samples for $C/N_0 = 15$ dB/Hz. The averaging time is $T_{av} = 20$ ms.

Whenever the blue I–Q samples enter the region in the lower left half plane bounded by the black dashed line, the MM filter will make a false determination

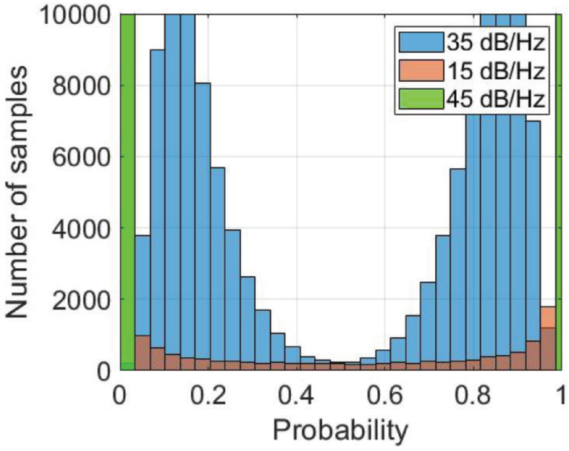

FIGURE 20 Data bit probability for different levels of C/N_0

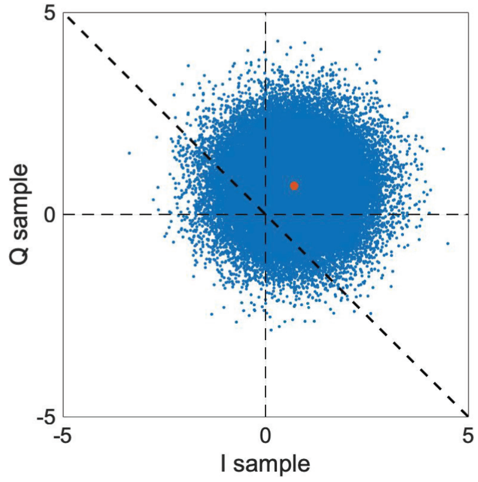

FIGURE 21 Example I–Q sample distribution

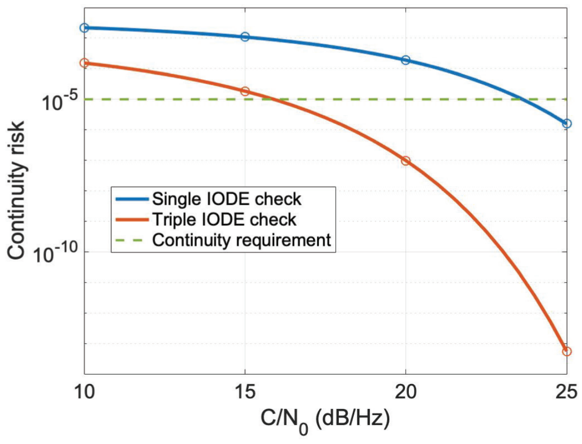

FIGURE 22 Continuity risk per hour for the two types of IODE checks

on the data bit. The probability of a wrong bit determination is a function of current carrier phase, C/N_0, and averaging time, T_{av}. The I and Q samples are independently and identically distributed as $N(1, \sigma^2)$, where σ is related to C/N_0 and T_{av} by Equations (8) and (9). Given $C/N_0 = 15$ dB/Hz and $T_{av} = 20$ ms, the probability of false bit determination P_f is 13%.

For an example (typical) navigation service continuity risk requirement of 10^{-5} per hour, we can evaluate whether either of the two types of IODE checks meet the requirement. Figure 22 shows the continuity risk as a function of C/N_0 and $T_{av} = 20$ ms for both checks. The single-IODE-check method fails to meet the continuity requirement for $C/N_0 <= 24$ dB/Hz. The triple-IODE-check method manages to maintain continuity down to approximately 15 dB/Hz.

We evaluated the performance improvement of the MM filter using the same data as in Section 6.2. The results are shown in Figures 23 and 24. Compared with the results in Figure 17, the single-IODE-check method reduces the standard deviation of carrier frequency estimation error from 0.1388 Hz to 0.0832 Hz, but does not meet the example continuity requirement for $C/N_0 \leq 24$ dB/Hz. The triple-IODE-check method achieves a smaller improvement of nearly 17%, reducing the standard deviation to 0.1107 Hz, but significantly lowers the continuity risk.

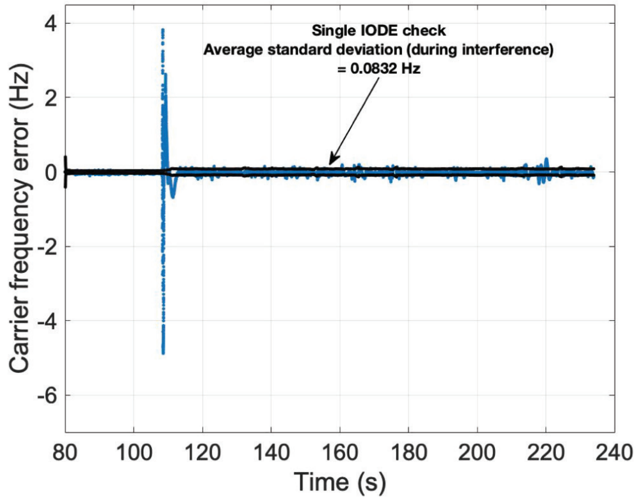

FIGURE 23 Carrier frequency estimation error for PRN 28 with upload-robust bit prediction and single IODE check

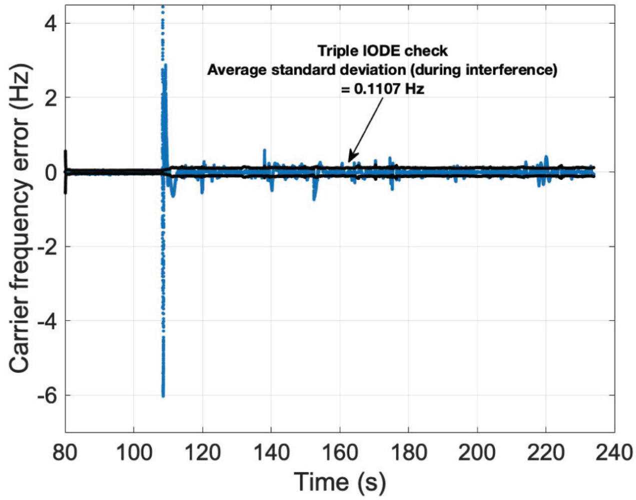

FIGURE 24 Carrier frequency estimation error for PRN 28 with upload-robust bit prediction and triple IODE check

8 | CONCLUSION

In this paper, we proposed an MM carrier phase estimation algorithm to enable GPS receivers to track carrier phase and frequency through wideband RFI events. We demonstrated the effectiveness of the estimator and its component Kalman filters in carrier phase tracking through simulations and experimental tests. Even under strong wideband interference events with $C/N_0 = 15$ dB-Hz, we show that the new estimator is able to accurately estimate carrier phase and frequency, whereas PLLs cannot maintain a lock. Finally, we show experimental results obtained by using the estimator with new navigation data bit prediction strategies to further improve performance.

ACKNOWLEDGMENTS

We thank our sponsors at the Federal Aviation Administration for supporting this research. The views and opinions expressed in this paper are those of the authors and do not necessarily reflect those of any other organization or person.

CONFLICT OF INTEREST

The authors declare no potential conflicts of interest.

REFERENCES

Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: Theory, algorithms, and software*. John Wiley Sons. https://doi.org/10.1002/0471221279

Borio, D., & Lachapelle, G. (2009). A non-coherent architecture for GNSS digital tracking loops. *Annales Des Télécommunications*, *64*(9–10), 601–614. https://doi.org/10.1007/s12243-009-0114-1

Chan, F.-C., Joerger, M., & Pervan, B. (2014). Stochastic modeling of atomic receiver clock for high integrity GPS navigation. *IEEE Transactions on Aerospace and Electronic Systems*, *50*(3), 1749–1764. https://doi.org/10.1109/taes.2014.120402

Dunn, M. J. (2013). *Global Positioning System Directorate Systems Engineering & Integration: Interface Specification IS-GPS-200*. https://www.gps.gov/technical/icwg/IS-GPS-200H.pdf

Humphreys, T. E., Psiaki, M. L., & Kintner, P. M. (2010). Modeling the effects of ionospheric scintillation on GPS carrier phase tracking. *IEEE Transactions on Aerospace and Electronic Systems*, *46*(4), 1624–1637. https://doi.org/10.1109/taes.2010.5595583

Humphreys, T. E., Psiaki, M. L., Kintner, P. M., & Ledvina, B. M. (2005). GPS carrier tracking loop performance in the presence of ionospheric scintillations. *Proc. of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2005)*, Long Beach, CA, 156–167. https://doi.org/10.15781/t2222rp1p

Morton, Y. J., Van Diggelen, F., Spilker, J. J., Jr, Parkinson, B. W., Lo, S., & Gao, G. (2021). *Position, navigation, and timing technologies in the 21st century: Integrated satellite navigation, sensor systems, and civil applications, Volume 1*. Wiley-IEEE Press. https://doi.org/10.1002/9781119458449

Psiaki, M. L., & Jung, H. Y. (2002). Extended Kalman filter methods for tracking weak GPS signals. *Proc. of the 15th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 2002)*, Portland, OR, 2539–2553. https://gps.mae.cornell.edu/EKF_Tracking_Cornell_ION2002.pdf

Pullen, S., & Gao, G. X. (2012). GNSS jamming in the name of privacy: Potential threat to GPS aviation. *Inside GNSS*, *7*(2), 34–43. https://insidegnss.com/wp-content/uploads/2018/01/marapr12-Pullen.pdf

Stevanovic, S., & Pervan, B. (2017). Coasting through wideband interference events using robust carrier phase tracking. *Proc. of the 30th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2017),* Portland, OR, 4184–4196. https://doi.org/10.33012/2017.15389

Vila-Valls, J., Closas, P., & Fernandez-Prades, C. (2015). Advanced KF-based methods for GNSS carrier tracking and ionospheric scintillation mitigation. *Proc. of the 2015 IEEE Aerospace Conference*, Big Sky, MT, 1–10. https://doi.org/10.1109/aero.2015.7118930

Vila-Valls, J., Closas, P., Navarro, M., & Fernandez-Prades, C. (2017). Are PLLs dead? A tutorial on Kalman filter-based techniques for digital carrier synchronization. *IEEE Aerospace and Electronic Systems Magazine*, *32*(7), 28–45. https://doi.org/10.1109/maes.2017.150260

Vila-Valls, J., Linty, N., Closas, P., Dovis, F., & Curran, J. T. (2020). Survey on signal processing for GNSS under ionospheric scintillation: Detection, monitoring, and mitigation. *NAVIGATION*, *67*(3), 511–536. https://doi.org/10.1002/navi.379

Zhao, W., Khanafseh, S., & Pervan, B. (2021). Clock dynamics analysis in IMM filter for carrier phase estimation through interference. *Proc. of the 34th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2021),* St. Louis, MO, 3322–3334. https://doi.org/10.33012/2021.18024

Zhao, W., & Pervan, B. (2019). IMM mthods for carrier phase tracking and navigation data bit estimation through interference. *Proc. of the 2019 International Technical Meeting of the Institute of Navigation*, Reston, VA, 248–256. https://doi.org/10.33012/2019.16692

Zhao, W., & Pervan, B. (2020a). Data bit assisted adaptive IMM filter for carrier phase tracking through interference. *Proc. of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2020)*, 3098–3107. https://doi.org/10.33012/2020.17705

Zhao, W., & Pervan, B. (2020b). Experimental validation of IMM algorithm for carrier phase tracking through interference. *Proc. of the 2020 International Technical Meeting of the Institute of Navigation*, San Diego, CA, 315–319. https://doi.org/10.33012/2020.17145

How to cite this article: Zhao, W., Khanafseh, S., & Pervan, B. (2024). Adaptive multiple-model Kalman filter for GNSS carrier phase and frequency estimation through wideband interference. *NAVIGATION, 71*(2). https://doi.org/10.33012/navi.646

APPENDIX

A | MM ALGORITHM SIMPLIFICATION

Following Bar-Shalom et al. (2004), the standard MM mixing process for mode *j*, with r modes in total, is as follows:

$$\hat{x}_{jm}(k-1|k-1)=\sum_{i=1}^{r}\hat{x}_i(k-1|k-1)\mu_{i|j}(k-1|k-1) \qquad j=1,\ldots,r \tag{A1}$$

where the mixing probabilities $\mu_{i|j}(k-1|k-1)$ are as follows:

$$\mu_{i|j}(k-1|k-1)=\frac{1}{\bar{c}_j}p_{ij}\mu_i(k-1) \qquad i,\, j=1,\ldots,r \tag{A2}$$

p_{ij} is the (*i*, *j*)-th element of the transition probability matrix, and $\mu_i(k-1)$ is the mode probability. We have the following:

$$\bar{c}_j=\sum_{i=1}^{r}p_{ij}\mu_i(k-1) \qquad j=1,\ldots,r \tag{A3}$$

The mode probabilities are given by the following:

$$\mu_j(k)=\frac{1}{c}\Lambda_j(k)\bar{c}_j \qquad j=1,\ldots,r \tag{A4}$$

where:

$$c=\sum_{i=1}^{r}\Lambda_j(k)\bar{c}_j \tag{A5}$$

and $\Lambda_j(k)$ is the likelihood function for mode *j* at time *k*.

Given measurement vector $z(k)$ and observation matrix $H_j(k)$, the Kalman filter innovation vector is $\gamma_j(k)=z(k)-H_j(k)\hat{x}(k|k-1)$, and its covariance matrix is $G_j(k)=V_j(k)+H_j(k)^T P_j(k|k-1)H_j(k)$. The likelihood function for mode *j* at time k is then given as follows:

$$\Lambda_j(k)=\exp\left(-\frac{1}{2}\gamma_j(k)^T G_j(k)^{-1}\gamma_j(k)\right) \tag{A6}$$

Example

For $r=2$, as in our application and for the case in which no prior knowledge of the data bits is available, $p_{11}=p_{12}=p_{21}=p_{22}=\frac{1}{2}$. We then have the following:

$$\bar{c}_1=\bar{c}_2=\frac{1}{2}(\mu_1(k-1)+\mu_2(k-1))=\frac{1}{2} \tag{A7}$$

By substituting into Equation (A2), we obtain the following:

$$\begin{aligned} \mu_{1|1}(k-1|k-1) &= \mu_{1|2}(k-1|k-1) = \mu_1(k-1) \\ \mu_{2|1}(k-1|k-1) &= \mu_{2|2}(k-1|k-1) = \mu_2(k-1) \end{aligned} \tag{A8}$$

Therefore, we have the following relationship:

$$\begin{aligned} \hat{x}_{1m}(k-1|k-1) = \hat{x}_{2m}(k-1|k-1) \\ &= \hat{x}_1(k-1|k-1)\mu_1(k-1) + \hat{x}_2(k-1)\mu_2(k-1) \\ &= \hat{x}(k-1) \end{aligned} \tag{A9}$$

as claimed.

Following Bar-Shalom et al. (2004) and taking mode 1 as an example, the standard covariance matrix mixing process is as follows:

$$\begin{aligned} P_{1m}(k-1|k-1) = \mu_{1|1}(k-1|k-1)\{P_1(k-1|k-1) + [\hat{x}_1(k-1|k-1) - \hat{x}_{1m}(k-1|k-1)] \\ \cdot[\hat{x}_1(k-1|k-1) - \hat{x}_{1m}(k-1|k-1)]^T\} \\ + \mu_{2|1}(k-1|k-1)\{P_2(k-1|k-1) + [\hat{x}_2(k-1|k-1) - \hat{x}_{1m}(k-1|k-1)] \\ \cdot[\hat{x}_2(k-1|k-1) - \hat{x}_{1m}(k-1|k-1)]^T\} \end{aligned} \tag{A10}$$

By substituting Equations (A8) and (A9) into Equation (A10), we obtain the following:

$$\begin{aligned} P_{1m}(k-1|k-1) = \mu_1(k-1)\{P_1(k-1|k-1) + [\hat{x}_1(k-1|k-1) - \hat{x}(k-1|k-1)] \\ \cdot[\hat{x}_1(k-1|k-1) - \hat{x}(k-1|k-1)]^T\} \\ + \mu_2(k-1)\{P_2(k-1|k-1) + [\hat{x}_2(k-1|k-1) - \hat{x}(k-1|k-1)] \\ \cdot[\hat{x}_2(k-1|k-1) - \hat{x}(k-1|k-1)]^T\} \end{aligned} \tag{A11}$$

where the right-hand side is the expression for the (combined) output covariance, $P(k-1|k-1)$, so that the following holds:

$$P_{1m}(k-1|k-1) = P(k-1|k-1) \tag{A12}$$

Similarly, we have the following:

$$P_{2m}(k-1|k-1) = P(k-1|k-1) \tag{A13}$$

The mode probabilities themselves are obtained by substituting Equation (A7) into Equations (A5) and (A4):

$$\mu_1(k) = \frac{\Lambda_1(k)}{\Lambda_1(k) + \Lambda_2(k)} \tag{A14}$$

$$\mu_2(k) = \frac{\Lambda_2(k)}{\Lambda_1(k) + \Lambda_2(k)} \tag{A15}$$

B | DATA BIT PREDICTION STRATEGY

Tables B1 and B2 present the details of our data bit prediction strategy. In these tables, we show how many bits we can predict among the total bits of each parameter in the navigation message over a long time period including an ephemerides cutover and new upload.

TABLE B1
Data Bit Prediction Strategy for Subframes 1, 2, and 3

	Subframe 1		Subframe 2		Subframe 3	
Word	**Parameter**	**Predictable/ Total bits**	**Parameter**	**Predictable/ Total bits**	**Parameter**	**Predictable/ Total bits**
Word 1	Preamble	8/8	Preamble	8/8	Preamble	8/8
	Telemetry message	14/14	Telemetry message	14/14	Telemetry message	14/14
	Integrity status flag	1/1	Integrity status flag	1/1	Integrity status flag	1/1
Word 2	Time of week	17/17	Time of week	17/17	Time of week	17/17
	Alert flag	1/1	Alert flag	1/1	Alert flag	1/1
	Anti-spoof flag	1/1	Anti-spoof flag	1/1	Anti-spoof flag	1/1
	Subframe ID	3/3	Subframe ID	3/3	Subframe ID	3/3
Word 3	Week number	10/10	IODE	0/8	C_{ic}	0/16
	C/A or P on L2	2/2	C_{rs}	5/16	Ω_0 MSBs	8/8
	URA index	2/4				
	SV HEALTH	6/6				
	IODC MSBs	2/2				
Word 4	L2 P data flag	1/1	Δn	5/16	Ω_0 LSBs	2/24
			M_0 MSBs	0/8		
Word 5			M_0 LSBs	0/24	C_{is}	0/16
					i_0 MSBs	8/8
Word 6			C_{UC}	0/16	i_0 LSBs	7/24
			e MSBs	8/8		
Word 7	T_{GD}	8/8	e LSBs	7/24	C_{rc}	5/16
					ω MSBs	8/8
Word 8	t_{oc}	4/16	C_{US}	0/16	ω LSBs	0/24
			$\sqrt{A}$ MSBs	8/8		
Word 9	a_{f2}	8/8	$\sqrt{A}$ LSBs	7/24	$\dot{\Omega}$	10/24
	a_{f1}	16/16				
Word 10	a_{f0}	0/22	t_{oe}	4/16	IODE	0/8
			Fit interval flag	1/1	IDOT	3/14
			AODO	0/5		

URA: User Range Accuracy
SV: Space Vehicle
IODC: Issue of Data, Clock
IODE: Issue of Data, Ephemeris
MSB: Most Significant Bit
LSB: Least Significant Bit
AODO: Age of Data Offset
IDOT: Rate of Inclination Angle

TABLE B2
Data Bit Prediction Strategy for Subframes 4 and 5

	Subframe 4 and 5 almanac pages		Subframe 4 page 18		Subframe 5 page 25	
Word	**Parameter**	**Predictable/Total bits**	**Parameter**	**Predictable/Total bits**	**Parameter**	**Predictable/Total bits**
Word 1	Preamble	8/8	Preamble	8/8	Preamble	8/8
	Telemetry message	14/14	Telemetry message	14/14	Telemetry message	14/14
	Integrity status flag	1/1	Integrity status flag	1/1	Integrity status flag	1/1
Word 2	Time of week	17/17	Time of week	17/17	Time of week	17/17
	Alert flag	1/1	Alert flag	1/1	Alert flag	1/1
	Anti-spoof flag	1/1	Anti-spoof flag	1/1	Anti-spoof flag	1/1
	Subframe ID	3/3	Subframe ID	3/3	Subframe ID	3/3
Word 3	DATA ID	2/2	DATA ID	2/2	DATA ID	2/2
	SV ID	6/6	SV (PAGE) ID	6/6	SV (PAGE) ID	6/6
	e	8/16	α_0	8/8	t_{oa}	2/8
			α_1	4/8	WN_a	8/8
Word 4	t_{oa}	0/8	α_2	2/8	SV HEALTH 1–4	24/24
	δ_i	8/16	α_3	8/8		
			β_0	2/8		
Word 5	$\dot{\Omega}$	8/16	β_1	8/8	SV HEALTH 5–8	24/24
	SV HEALTH	8/8	β_2	8/8		
			β_3	8/8		
Word 6	$\sqrt{A}$	15/24	A_1	0/24	SV HEALTH 9–12	24/24
Word 7	Ω_0	6/24	A_0 MSBs	0/24	SV HEALTH 13–16	24/24
Word 8	ω	8/24	A_0 LSBs	0/8	SV HEALTH 17–20	24/24
			t_{ot}	2/8		
			WN_t	2/8		
Word 9	M_0	2/24	Δt_{LS}	2/8	SV HEALTH 21–24	24/24
			WN_{LSF}	0/8		
			DN	0/8		
Word 10	a_{f0} MSBs	5/8	Δt_{LSF}	8/8		
	a_{f1}	11/11				
	a_{f0} LSBs	0/3				

WN: Week Number
DN: Day Number