GNSS Code Tracking in Presence of Data

Kewen Sun∗
*Politecnico di Torino, Dipartimento di Elettronica
C.so Duca degli Abruzzi 24, 10129 Torino, Italy
Email: kewen.sun@polito.it

Abstract—After the acquisition stage of a GNSS receiver has roughly synchronized the received and the locally generated codes typically within less than a half chip period residual offset, a fine synchronization stage, named signal tracking, takes over and continuously maintains and corrects the best possible alignment between the two codes by means of closed loop operations. In this paper, we mainly consider the navigation data sign transition problem present in a code tracking loop. In order to overcome this problem, a novel Assisted-GNSS (A-GNSS) based code tracking technique has been proposed to deal with the data sign transitions present in the code tracking loop of a GNSS receiver in order to enhance the tracking performance. The performance improvement of this proposed tracking technique has been proved by means of simulation analysis. This proposed technique mitigates the data sign transitions impact occurring in the code tracking loop by proving assistance data which enables signal tracking more robust in challenging environments.

Index Terms—Code tracking loop; Carrier tracking loop; Data sign transition; A-GNSS.

I. INTRODUCTION

The next generation of Global Navigation Satellite Systems (GNSS’s), such as Galileo [1] and GPS modernization [2], will use signals with equal code and bit periods, which will introduce a potential bit sign transition in each primary code period of the received signal processed in the code tracking loop of a GNSS receiver, which results in the so called Cross Ambiguity Function (CAF) peak splitting impairment problem [3], [4]. In this paper, a novel data-assisted GNSS code tracking strategy has been proposed to mitigate this impairment effect in order to enhance the tracking performance of a GNSS receiver, particularly in weak signal environments.

In a GNSS receiver, after the acquisition unit has successfully detected the presence of a given satellite and coarsely estimated the offset on the residual carrier and the code phase delay with respect to the local replicas, a fine synchronization stage, named signal tracking, is activated to refine these estimation values, continuously maintain and correct the best possible alignment between the received and locally generated codes by means of closed loop operations. This fine synchronization is fundamental for measuring the pseudo range, based on code phase measurements, or also the carrier phase measurements.

The whole signal tracking process is a two-dimensional (code and carrier) signal replication process. It consists of two inter-operating feedback loops, a Delay Lock Loop (DLL) for code tracking and a Phase Lock Loop (PLL) for carrier tracking (typically a Costas Loop). The tracking operation is made of an iterative procedure during which the carrier tracking loop and the code tracking loop cooperate to provide the best estimates of the Doppler shift (fd and, in some cases, also of the phase of the incoming signal) and of the code phase delay (τ). The outputs of the acquisition phase \( \hat{p}(A) = (\hat{f}_d(A), \hat{\tau}(A)) \) are used to initialize the two tracking loops. When the output of the carrier tracking loop is used to remove the modulation (carrier wipe-off) for the code phase delay estimation, the code tracking loop receives a spreading code still modulated by the navigation data, the correlation function peak will be split along the Doppler shift axis caused by the presence of bit sign transition. The DLL must consider the impact of the data sign reversal, and a possible way should be provided to mitigate this impairment effect. The bit sign transition effect on the CAF present in a code tracking loop has been analyzed from a mathematical point of view in this paper.

GNSS today is expected to work almost anywhere, even, sometimes, indoor poor signal conditions. Considering the data sign transitions problem present in the code tracking loop after the carrier wipe-off procedure, novel Assisted-GNSS (A-GNSS) scheme could be adopted to the data bits removal. A-GNSS improves on standard GNSS performance by providing information through an alternative communication channel. In this sense, A-GNSS works by providing assistance data the same (or equivalent) as those present in the code tracking loop in order to compensate the data bit transitions impact.

In summary, in this paper, a novel A-GNSS based code tracking method has been considered to deal with the data bit transitions present in the code tracking loop of a GNSS receiver, which aims at enhancing the tracking performance. The performance improvement of this proposed tracking technique has been proved by means of simulation analysis. From the performance evaluation results, it emerges that this proposed technique mitigates the data sign transitions impact occurring in the code tracking loop by proving assistance data and enables signal tracking procedure more robust in challenging environments.

II. TRADITIONAL GNSS TRACKING STRUCTURE

Once the acquisition unit has successfully detected the presence of a given satellite vehicle (SV) and estimated the offset on the residual carrier and the code phase delay with respect to the local replicas, a fine synchronization stage, named signal tracking, is activated to refine these values, keep track and demodulate the navigation data from the specific satellite. This fine synchronization is fundamental for measuring the
pseudo range, based on code phase measurements, or also the carrier phase measurements.

The whole signal tracking process is a two-dimensional (code and carrier) signal replication process. It consists of two inter-operating feedback loops, a DLL for code tracking and a PLL for carrier tracking (typically a Costas Loop). Fig. 1 shows a high-level block diagram of the functionalities for a typical GNSS receiver. Obviously, an n parallel channels receiver will have n sets of blocks corresponding to each independent tracking loops. In a GNSS receiver, the digitized intermediate frequency (IF) signal is input to each of these parallel channels.

The tracking is made of an iterative procedure during which the carrier tracking loop and the code tracking loop cooperate to provide the best estimates of the Doppler frequency shift ($\hat{f}_d$ and, in some cases, also of the phase $\hat{\phi}$ of the incoming signal) and of the code phase delay $\hat{\tau}$.

The outputs of the acquisition stage $P_i^{(A)}(\hat{f}_d, \hat{\tau})$ ($i = 1, \cdots, n$) are used to initialize the two tracking loops. After the initialization they must operate together at the same time since:

- A good estimate of the code phase delay is not possible until the residual modulation due to the Doppler shift is removed;
- A good estimate of the residual Doppler shift is not possible until the code signal is canceled out, in order to allow the carrier loop to operate on a pure tone signal.

For these reasons the best estimate is obtained after several steps of approximation during which the output of the carrier tracking loop is used to remove the modulation (carrier wipe-off) for the code estimation, and the output of the code tracking loop is used to cancel out the code signal (code wipe-off) for the carrier estimation. This scheme is clearly illustrated in Fig. 2.

To demodulate the navigation data successfully an exact carrier wave replica has to be generated. To track a carrier wave signal, a carrier tracking loop is often adopted. The carrier tracking loop is a feedback loop able to finely estimate the frequency ($f_{IF} + f_d$) of a noisy sinusoidal wave and to track the frequency changes if the user and the satellite move. Most receivers also track the phase term $\phi$ present in the carrier wave signal.

Fig. 3 illustrates a block diagram of a GNSS receiver carrier tracking loop. The programmable designs of the carrier predetection integrators, the carrier loop discriminators, and the carrier loop filters characterize the receiver carrier tracking loop. These three functions determine the two most important performance characteristics of the receiver carrier loop design: the carrier loop thermal noise error and the maximum line-of-sight (LOS) dynamic stress threshold. Since the carrier tracking loop is always one of the most sensitive blocks in a stand-alone GNSS receiver, its threshold characterizes the unaided GNSS receiver performance [2].

The purpose of the carrier tracking loop is to generate an estimate of the phase or frequency and Doppler shift of the received GNSS RF carrier. It does this by generating a replica of the IF carrier which is in phase with incoming signal. Carrier tracking loops which provide an estimate of phase are PLL’s.

The PLL provides an estimate of the observed phase $\phi_i$ of the received GNSS signal seen at the receiver’s antenna. It generates an estimate of $\phi_i$ by internally generating a replica of the measured signal at the antenna and synchronizing the phase of this replica with that of the measured signal. The phase of the internal replica is the output of the PLL, which is denoted as $\phi_o$. We would like a PLL to provide an accurate estimate of the observed phase. That is, we would like to
minimize the difference between $\varphi_i$ and $\varphi_o$. Therefore, the PLL tries to minimize the tracking error $\delta \varphi$, which is the difference between the received signal phase and the phase of the replica. That is

$$\delta \varphi = \varphi_i - \varphi_o \quad (1)$$

In Fig. 3, the two first multiplications wipe off the carrier and the PRN code of the input signal. To wipe off the PRN code, the prompt output from the early-late code tracking loop is used. The carrier loop discriminator block is used to find the phase error on the local carrier wave replica. The output of the discriminator, which is the phase error (or a function of the phase error), is then filtered and used as a feedback to the numerically controlled oscillator (NCO), which adjusts the frequency of the local carrier wave. In this way the local carrier wave could be an almost precise replica of the input signal carrier wave.

### III. Carrier Tracking Loops

If the GNSS receiver is tracking a data channel signal, it must be noticed that after the code wipe-off procedure the PLL receives a continuous wave signal still modulated by the navigation data. The problem with using an ordinary PLL is that it is insensitive to $180^\circ$ phase shifts. Due to navigation bit transitions, a PLL used in a GNSS receiver has to be insensitive to $180^\circ$ phase shifts.

Any carrier loop that is insensitive to the presence of data modulation is usually called a Costas loop. Typically a Costas loop implementation of the PLL model is utilized for carrier tracking and navigation data bit decoding. The Costas loop tolerates the presence of data modulation on the received signal (it is insensitive to $180^\circ$ phase reversal due to data bit transitions) and then provides a carrier phase reference. Obviously, also the Costas loop requires prior PRN code despreading (code wipe-off) in order to correctly perform the carrier tracking.

Fig. 4 shows a Costas loop. One property of this loop is that it is insensitive for $180^\circ$ phase shifts due to navigation bits. The carrier wipe-off process used in the generic receiver design requires only two multiplications. Assuming that the carrier loop is in phase lock and that the replica cosine function is in phase with the incoming SV carrier signal (converted to IF), this results in a cosine squared product at the in-phase (I) output, which produces maximum $I_{PS}$ amplitude (signal plus noise) following the code wipe-off and integrate and dump processes. The second multiplication is between a $90^\circ$ phase-shifted carrier replica sine function and the incoming SV carrier. This results in a cosine x sine product at the quadrature (Q) output, which produces minimum $Q_{PS}$ amplitude (noise only). For this reason, the Costas loop tries to keep all energy in the I arm. To keep the energy in the I arm, some kind of feedback to the oscillator is needed. If it is assumed that the code replica is perfectly aligned, the multiplication in the I arm yields the following sum:

$$d(n) \cos(2\pi F_D n + \varphi_i) \cos(2\pi F_D n + \varphi_o)$$

where $\varphi_i$ is the incoming signal carrier phase, $\varphi_o$ is the phase of the local replica of the carrier phase, and $\delta \varphi$ is the phase difference between the phase of the input signal and the phase of the local replica carrier, called tracking error. The multiplication in the quadrature arm gives the following:

$$d(n) \cos(2\pi F_D n + \varphi_i) \sin(2\pi F_D n + \varphi_o)$$

The output of the phase discriminator goes through a loop filter to generate a signal which drives the NCO. The NCO generates a replica signal whose phase is synchronized to that of the incoming signal. Both the carrier tracking loop (Costas loop) and the following described code tracking loop (Delay Lock Loop DLL) have an analytical linear phase lock loop model that can be exploited to predict and analyze the performance.

![Costas loop used to track the carrier wave.](Fig. 4)
The code tracking loop is a feedback loop able to finely estimate the residual code delay \( d_z = \tau - \hat{\tau}(A) \) by means of a Delay Lock Loop (DLL) called an early-late tracking loop, which is a feedback system to control the behavior of the system itself. The main task of a DLL is to align a local replica code (called prompt code) to the code received from each detected satellite. The information about this relative delay between the incoming and the local codes is contained in the correlation peak, therefore the idea could be to finely estimate the correlation value. However, a search of the maximum of the correlation peak is not an effective approach, and it is not used in conventional GNSS receivers. It is then necessary to adopt a strategy insensitive to the absolute correlation peak value, based on a discrimination function that is null only when the incoming and the local codes are synchronized (null-seeker).

A linear model for the analog version of the PLL depicted in Fig. 4 is shown in Fig. 5. This linear PLL model is more suitable for analytical work, which could still be the basis of performance prediction. Thus, for simplicity, an analog model could be used at this stage. The second-order PLL system contains a first-order loop filter and a voltage controlled oscillator (VCO). The transfer functions of an analog loop filter and a VCO are

\[
F(s) = \frac{1}{s} + \frac{\tau_2 s + 1}{\tau_1} \quad (8)
\]

\[
N(s) = \frac{K_o}{s} \quad (9)
\]

where \( F(s) \) and \( N(s) \) are the transfer functions of the loop filter and NCO, respectively. \( K_o \) is the NCO gain. The transfer function of a linearized analog PLL is

\[
H(s) = \frac{K_d F(s) N(s)}{1 + K_d F(s) N(s)} \quad (10)
\]

where \( K_d \) is the gain of the phase discriminator. Substituting Equations (8) and (9) into the closed-loop transfer function in Eq. (10) yields

\[
H(s) = \frac{2\omega_n s + \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (11)
\]

where the natural frequency \( \omega_n = \sqrt{\frac{K_o K_d}{\tau_1}} \), and the damping ratio \( \xi = \frac{\tau_2 \omega_n}{\omega_n} \).

Another carrier tracking loop, which is able to track the frequency of the incoming signal’s carrier ignoring its phase term \( \varphi_1 \), is known as Frequency Lock Loop (FLL). The FLL is often used to initialize the frequency wipe-off system, by providing a frequency estimate quite close to the correct one. Once the FLL has locked a frequency \( \hat{f}_{d,FLL} \), the PLL can refine the frequency estimation working in a narrow band around \( \hat{f}_{d,FLL} \).

IV. CODE TRACKING LOOPS

After the acquisition stage of a GNSS receiver has accomplished a rough alignment between the incoming and the local codes within a fraction of a chip interval, such an initial estimate of the code phase delay should be further refined. Furthermore, because of the relative motion between satellite and user receiver and the instability of clocks, correction must be made continuously. This process is performed by a code tracking loop, in order to keep track of the code phase of a specific code in the signal. The output of such a code tracking loop is a perfectly aligned replica of the code.

The code tracking loop is a feedback loop able to finely estimate the residual code delay \( d_z = \tau - \hat{\tau}(A) \) by means of a Delay Lock Loop (DLL) called an early-late tracking loop, which is a feedback system to control the behavior of the system itself. The main task of a DLL is to align a local replica code (called prompt code) to the code received from each detected satellite. The information about this relative delay between the incoming and the local codes is contained in the correlation peak, therefore the idea could be to finely estimate the correlation value. However, a search of the maximum of the correlation peak is not an effective approach, and it is not used in conventional GNSS receivers. It is then necessary to adopt a strategy insensitive to the absolute correlation peak value, based on a discrimination function that is null only when the incoming and the local codes are synchronized (null-seeker).

The idea behind the DLL is to correlate the input signal with three replicas of the code seen in Fig. 6. The first step is converting the PRN code to baseband, by multiplying the incoming signal with a perfectly aligned local replica of the carrier wave. Afterwards the signal is multiplied with three code replicas (early, prompt, and late). The three replicas are nominally generated with a spacing of \( \pm \frac{1}{2} \) chip. After this second multiplication, the six outputs are integrated and dumped. The output of these integrations is a numerical value indicating how much the specific code replica correlates with the code in the incoming signal. The DLL design in Fig. 6 has the advantage that it is independent of the phase on the local carrier wave. If the local carrier wave is in phase with the
input signal, all the energy will be in the in-phase arm. But if the local carrier phase drifts compared to the input signal, the energy will switch between the in-phase and the quadrature arms.

If the code tracking loop performance has to be independent of the performance of the phase lock loop, the code tracking loop has to use both the in-phase and quadrature arms to track the code. The DLL needs a feedback to the PRN code generators if the code phase has to be adjusted.

The requirements of a DLL discriminator is dependent on the type of application and the noise in the signal. The space between the early, prompt, and late codes determines the noise bandwidth in the delay lock loop. If the discriminator spacing is larger than $\frac{1}{2}$ chip, the DLL would be able to handle wider dynamics and be more noise robust; on the other hand, a DLL with a smaller spacing would be more precise. In a modern GNSS receiver the discriminator spacing can be adjusted while the receiver is tracking the signal. The advantage from this is that if the signal-to-noise ratio suddenly decreases, the receiver uses a wider spacing in the correlators to be able to handle a more noisy signal, and hereby a possible code lock loss could be avoided.

In this paper, the implemented code tracking loop discriminator is the normalized early minus late power. This discriminator is described as

$$D = \frac{(I_{ES}^2 + Q_{ES}^2) - (I_{LS}^2 + Q_{LS}^2)}{(I_{ES}^2 + Q_{ES}^2) + (I_{LS}^2 + Q_{LS}^2)} \quad (12)$$

where $I_{ES}$, $Q_{ES}$, $I_{LS}$ and $Q_{LS}$ are four outputs of the six correlators shown in Fig. 6. The normalized early minus late power discriminator is chosen because it is independent of the performance of the PLL as it uses both the in-phase and quadrature arms. The normalization of the discriminator makes the discriminator adapt to be used with signals with different signal-to-noise ratios and different signal strengths.

V. THE PROBLEM OF DATA PRESENT IN THE CODE TRACKING LOOP

After the acquisition system has coarsely estimated the Doppler shift $f_d^{(A)}$ and the code phase delay $\tau^{(A)}$, the tracking stage is activated to refine the estimation of the two parameters $f_d$ and $\tau$, keep track and demodulate the navigation message from the specific satellite. As described in section II, the tracking operation is an iterative procedure during which the carrier tracking loop and the code tracking loop cooperate to provide the best estimates of the Doppler frequency shift and the code phase delay. After the output of the carrier tracking loop is used to remove the modulation (carrier wipe-off) for the code phase estimation, the code tracking loop receives a spreading code still modulated by the navigation data, which is shown in Fig. 7. The DLL must consider the impact of the data bits and try to find a possible way to mitigate this effect. In this paper, Assisted-GNSS (A-GNSS) technique will be exploited to overcome this problem, which will be illustrated in section V-A.

A. A-GNSS

GNSS was originally designed for military tasks, which was expected to work outside with a relatively clear view of the sky. Today GNSS is used for many more civilian than military purposes. In particular, the system demands of civilian applications far exceed those seen before. In very poor signal conditions, for example in a city, these signals may suffer multipath where signals bounce confusingly off buildings, or be weakened by passing through walls or tree cover. When first turned on in these conditions, some non-assisted GNSS navigation devices may not be able to work out a position due to the fragmentary signal, rendering them unable to function until a clear signal can be received continuously.

GNSS is expected to work almost anywhere, even, sometimes, indoors; push-to-fix applications have emerged where a single position is expected almost instantly; and all of this must be delivered in a way that adds little or no cost, size, or power consumption to the host device. These requirements are what drove the development of A-GNSS.

To calculate a position (or fix), a GNSS receiver must first find and acquire the signal from each satellite and then decode the data from the satellites. But before it can acquire each satellite signal, it must find the correct frequency for that satellite and the correct code delay. Each satellite appears on a different frequency, thanks to the Doppler shift induced by the high speeds at which the satellites move. The observed
Doppler shift is a function of the location from which the receiver is acquiring the satellite signal. Before the receiver knows where it is, it cannot calculate the Doppler shift. Standard GNSS receivers with no a priori knowledge of these frequency variables would exhaustively search a large range of frequencies made up largely by the effects of satellite motion and receiver oscillator offset, and a small contribution from receiver velocity. However, even if the GNSS receiver has the correct frequency, it must still find the correct code delay for the correlators to generate a correlation peak. The receiver without any a priori knowledge of code delay will also have to search all possible code delay bins. This gives the GNSS receiver a two-dimensional search space for each satellite. We call this the frequency / code delay search space. Having found a signal, it is then necessary to decode data to find the position of the satellite. Only after this satellite position data is decoded could a GNSS receiver compute the position [6].

A-GNSS improves on standard GNSS performance by providing information, through an alternative communication channel, that allows the GNSS receiver to know what frequencies to expect before it even tries, and then the assistance data provides the satellite positions for use in the position computation. Fig. 8 shows an overview of an A-GNSS system. Having acquired the satellite signals, all that is left to do is to take range measurements from the satellites, the A-GNSS receiver can do so more quickly, and with weaker signals, than an unassisted receiver, and then the A-GNSS receiver can compute the position. Furthermore, because the A-GNSS receiver is designed to know in advance what frequencies to search, the typical architecture of the receiver changes to allow longer dwell times, which increase the amount of energy received at each particular frequency. This increases the sensitivity of the A-GNSS receiver and allows it to acquire signals at much lower signal strengths.

Concerning the data bits problem present in the code tracking loop after the carrier wipe off procedure, A-GNSS scheme could be adopted to the data bits removal. In this sense, A-GNSS works by providing assistance data the same (or equivalent) as those present in the code tracking loop, in order to compensate the data bit transitions impact. The assistance data often comes from the cellular network.

VI. SIMULATION ANALYSIS AND PERFORMANCE EVALUATION

In order to investigate the performance of the proposed A-GNSS based code tracking technique, simulations have been performed on the Galileo E1 OS BOC(1,1) signals. The Galileo E1 OS BOC(1,1) signals are simulated by N-FUELS signal generator which was developed by Navigation Satellite Signal Analysis and Simulation (NavSAS) research group of Politecnico di Torino, Italy, where the carrier-to-noise ratio \((C/N_0)\) is 46 dB-Hz, the sampling frequency \(f_s\) is 16.3676
MHz, the intermediate frequency $f_{IF}$ is 4.1304 MHz and the front-end quantization level is of 4 bits. Assistance data are provided in the A-GNSS based code tracking loop.

To clearly verify the impact of data bit transitions in a code tracking loop, the simulation campaigns are divided into two cases: the conventional code tracking and the data-assisted code tracking, aiming to have a whole view of the performance comparison among them. In the simulation campaigns, the predetection time is 8 ms in the Integrate and Dump block of a code tracking loop.

Firstly, simulation test has been done for the conventional code tracking without data assistance. In Fig. 9, the scatter plot of the demodulated in-phase (I prompt) and quadrature (Q prompt) components is reported, where the navigation data bits are present in the correlator outputs of the code tracking loop. As expected, two bubbles appear due to the navigation bit sign transitions. Fig. 10 shows in-phase prompt accumulation I prompt, which indicates the navigation bits in the incoming signal. In Fig. 11, the early, prompt and late (E-P-L) correlation envelopes of the code tracking loop in presence of the data are depicted.

Then, we consider the data-assisted code tracking case. Fig. 12 shows the scatter plot of the demodulated I prompt and Q prompt components, note that only one bubble appears, this is because the navigation data bits are wiped off by the assistance data. This can be further verified in Fig. 13, where the values of the demodulated in-phase component I prompt always have the same sign. In Fig. 14, the E-P-L correlation envelopes for the assisted code tracking loop are reported, respectively. With respect to the results presented in Fig. 11 for the conventional code tracking loop, much improved prompt correlation envelope values are achieved.

VII. CONCLUSION

In this paper, a novel A-GNSS based code tracking technique has been proposed to deal with the data bit transitions present in the code tracking loop of a GNSS receiver, which aims at enhancing the code tracking performance. The performance evaluation for this proposed tracking technique has been made by means of simulation analysis. From the performance analysis results, it emerges that this proposed technique mitigates the data sign transitions impact occurring in the code tracking loop by providing assistance data from cellular networks, which enables signal tracking procedure of a GNSS receiver more robust in challenging environments.

REFERENCES