FAST ITERATIVE MAXIMUM-LIKELIHOOD ALGORITHM (FIMLA) FOR MULTIPATH MITIGATION IN NEXT GENERATION OF GNSS RECEIVERS

M. Sahmoudi and M. G. Amin

Wireless Communications and Positioning Lab
Center for Advanced Communications
Villanova University, Villanova, PA 19085, USA
E-mail: {mohamed.sahmoudi, moeness.amin}@villanova.edu

ABSTRACT
In this paper, we efficiently solve the maximum-likelihood (ML) time-delay estimation problem for GNSS signals in a multipath environment. Exploiting the GNSS signal model structure and the spreading code periodicity, we develop an efficient implementation of the Newton iterative likelihood-maximization method by finding simple analytical expressions for the first and second derivatives of the likelihood function. The proposed Fast Iterative ML Algorithm (FIMLA) for time-delay estimation is shown to be an attractive technique for mitigation of closely-spaced multipath for current GPS receivers as well as future modernized GPS and the European Galileo systems based on BOC signals. Simulation results show that FIMLA improves the performance of the ML synchronization algorithm in presence of multipath.

1. INTRODUCTION
It is recognized that multipath remains as a dominant source of ranging error in Global Navigation Satellite Systems (GNSS) positioning [3, 16, 17]. Multipath is a critical issue in the development of high performance positioning applications such as high-productivity survey equipment, high-precision aircraft landing systems, and other applications of indoor navigation. Examples of recently developed high-speed in-receiver multipath mitigation methods include the narrow correlator, the strobe correlator, the Multipath Estimating Delay Lock Loop (MEDLL), Multipath Elimination Technology (MET), and the MMT [18]. Superior methods, like MEDLL, rely on ML estimation-theoretic principles and are driven to approach theoretical performance limits. However, they are complex and expensive to implement because of the need to measure the received signal cross-correlation function with multiple correlators and to process these measurements with complex algorithms [5, 11, 13]. One commonly used technique, a line search approach is applied to find the estimates that maximize the log-likelihood function in an ad-hoc and high computational manner. In the ML-type receivers, each estimated multipath correlation function component is in turn subtracted from the measured correlation function, and finally a standard DLL is applied to the direct path component for delay estimation. In this paper, we implement the maximum likelihood estimator computation based on a Newton type technique that is both effective and of very low computational cost [10]. By exploiting the GNSS signal model and the code periodicity, we simplify the log-likelihood function and derive analytic expressions of it’s first and second derivatives. In contrast to the standard implementation of the Newton approach, which uses a finite difference calculation to find the gradient and the Hessian of a cost function, the variables in our approach are calculated analytically. Further, we apply the ML approach to binary offset-carrier (BOC) signals which is a future common signal for the open service of E1/L1, achieving thus the best possible degree of interoperability in this band [9]. Although the overall performance of these new signals will presumably be better than that of current GPS signal, multipath will remain a major error source. Therefore, we present a performance comparison between the GPS C/A code (i.e. BPSK modulated) signal and the future GNSS BOC signals using the proposed FIMLA receiver in multipath environments.

2. GNSS SIGNALS AND PROPERTIES
2.1. C/A-code GPS signal
Satellite navigation systems use the direct sequence spread spectrum (DSSS) techniques for generating the modulation waveforms [7]. The general baseband DSSS signal has the following form,

\[ c(t) = \sum_{k=-\infty}^{\infty} a_k g(t - kT_c), \tag{1} \]

where \( g(t) \) is the waveform modulation, \( a_k \) are the code PRN values, assumed to be generated as a random coin-flip sequence. The time duration \( T_c \) (resp. frequency \( f_c \)) is the length (resp. chipping rate or code rate) of the PRN chip. Two signal characteristics of great importance for satellite navigation applications are the autocorrelation function and power spectral
density. The current C/A-code signal uses the BPSK signaling with rectangular chips, thus is referred by BPSK-R in the recent literature [7]. For those signals with unit-power,

\[
g_{\text{BPSK-R}}(t) = \begin{cases} \frac{1}{T_c}, & 0 \leq t \leq T_c \\ 0, & \text{elsewhere} \end{cases}
\] (2)

The corresponding autocorrelation function and power spectral density are given respectively by,

\[
R_{\text{BPSK-R}}(\tau) = \begin{cases} 1 - \frac{|\tau|}{T_c}, & |\tau| \leq T_c \\ 0, & \text{elsewhere} \end{cases}
\] (3)

and

\[
S_{\text{BPSK-R}}(f) = T_c \text{sinc}^2(\pi f T_c),
\] (4)

where \(\text{sinc}(t)\) is the sine cardinal (sinc) function. BPSK-R(\(n\)) is another notation often used to denote BPSK-R signals with \(n \times 1.023\)-MHz chipping rate.

### 2.2. Binary offset carrier (BOC) signal

Binary offset carrier describes a class of modulations recently introduced for the next generation of GNSS. Indeed, modernized GPS and the European Galileo system will use BOC-based signals on different carriers and with different parameters. The main reason for creating BOC signals were on one hand, the need to improve resistance to undesired signals, and on the other hand, the need for improved spectral sharing of the allocated bandwidth with other system signals. The BOC signal is an extension of the BPSK modulation which may be viewed as being the product of a BPSK-R sub-carrier and on the other hand, the need for improved resistance to undesired signals.

The BOC signal is an extension of the BPSK modulation which may be viewed as being the product of a BPSK-R signal with rectangular chips, thus is referred by BPSK-R(\(n\)) in the current C/A-code signal uses the BPSK signaling with rectangular chips, thus is referred by BPSK-R in the recent literature [7]. For those signals with unit-power,

\[
g_{\text{BPSK-R}}(t) = \begin{cases} \frac{1}{T_c}, & 0 \leq t \leq T_c \\ 0, & \text{elsewhere} \end{cases}
\] (2)

The corresponding autocorrelation function and power spectral density are given respectively by,

\[
R_{\text{BPSK-R}}(\tau) = \begin{cases} 1 - \frac{|\tau|}{T_c}, & |\tau| \leq T_c \\ 0, & \text{elsewhere} \end{cases}
\] (3)

and

\[
S_{\text{BPSK-R}}(f) = T_c \text{sinc}^2(\pi f T_c),
\] (4)

where \(\text{sinc}(t)\) is the sine cardinal (sinc) function. BPSK-R(\(n\)) is another notation often used to denote BPSK-R signals with \(n \times 1.023\)-MHz chipping rate.

#### 2.2. Binary offset carrier (BOC) signal

Binary offset carrier describes a class of modulations recently introduced for the next generation of GNSS. Indeed, modernized GPS and the European Galileo system will use BOC-based signals on different carriers and with different parameters. The main reason for creating BOC signals were on one hand, the need to improve resistance to undesired signals, and on the other hand, the need for improved spectral sharing of the allocated bandwidth with other system signals. The BOC signal is an extension of the BPSK modulation which may be viewed as being the product of a BPSK-R signal with a square wave sub-carrier. The main objective of the sub-carrier is to minimize the interference between GPS and Galileo signals. Let \(2T_s\) (resp. \(f_s\)) be the sub-carrier period (resp. frequency). The number of square wave half-periods in a spreading symbol is often selected to be an integer \(k\), such that

\[
T_s = \frac{1}{2T_s}, \quad T_c = \frac{1}{f_c}, \quad k = \frac{T_c}{T_s}.
\] (5)

A frequency reference \(f_0\) was chosen for the future GNSS signals which employ frequencies that are multiples of \(f_0 = 1.023\)-MHz. For the case of BOC signals, the sub-carrier frequency \(f_s\) and the shipping rate \(f_c\) are chosen to be

\[
f_s = m f_0, \quad f_c = n f_0.
\] (6)

Thus, the notation \(\text{BOC}(m, n)\) is often used as a shorthand. A BOC spreading symbol can be described as :

\[
g_{\text{BOC}}(t) = g_{\text{BPSK-R}}(t) \text{sinc}(t),
\] (7)

where \(\text{sinc}(t)\) is the the BOC sub-carrier of period \(2T_s\). When \(k\) is even, a BOC spreading symbol is given by

\[
g_{\text{BOC}}(t) = g_{\text{BPSK-R}}(t) \text{sgn} [\sin(\pi t/T_s + \psi)],
\] (8)

where \(\text{sgn}\) is the signum function and \(\psi\) is a selectable phase angle with two common values 0° and 90°, for which the resultant BOC signals are referred to as sine phased or cosine phased, respectively.

The autocorrelation and power spectrum are dependent on both the chip rate and characteristics of the square wave sub-carrier. The autocorrelation functions for BOC signals resemble saw teeth, piecewise linear functions between the peak values \([2, 7]\). The BOC subscripts s and c refer to sine-phased and cosine-phased, respectively. The power spectral density for a sine-phased BOC modulation is \([2, 7]\) :

\[
S_{BOC_s}(f) = \begin{cases} f_c \left[ \frac{\sin(\pi f T_s)}{\pi f} \cos(\pi f T_s) \right]^2, & k \text{ even} \\ f_c \left[ \frac{\sin(\pi f T_s)}{\pi f} \cos(\pi f T_s) \right]^2, & k \text{ odd} \end{cases}
\] (9)

The shape of this frequency spectrum enables the interoperability between GPS and future Galileo \([2]\).

### 3. ML TIME DELAY ESTIMATION

In a multipath-free situation, the received GPS signal from a given satellite can be written as,

\[
s(t) = A c(t - \tau) e^{j 2\pi f_s t} + \eta(t)
\] (10)

where \(A\) is the signal amplitude, \(c(t)\) is the transmitted GNSS signal, \(\tau\) is the time delay (transit time), \(f_s\) is the intermediate frequency that includes Doppler frequency, and \(\eta(t)\) denotes an additive white Gaussian noise with variance \(\sigma^2\). We assume that the signal has been stripped of the data. We cast the time delay estimation problem as a maximum-likelihood estimation problem in time domain. Sampled data are collected during the interval of time \(T_0\), where all the signal parameters are assumed to be constants. For convenience, we assume that the integration time is a multiple of the GNSS signal period. The statistical ML method calculates those estimates by maximizing the following likelihood function

\[
P(\tau, A, s) \propto \exp \left\{ \frac{-1}{2\sigma^2} \int_{T_0} |s(t) - A c(t - \tau) e^{j 2\pi f_s t}|^2 dt \right\}
\]

The log-likelihood function can, then, be expressed as,

\[
\Gamma(\tau, A) = - \int_{T_0} |s(t)|^2 dt + 2Re \left\{ A^* \int_{T_0} s(t) c(t - \tau) e^{-j 2\pi f_s t} dt \right\} - |A|^2 \int_{T_0} c^2(t - \tau) dt
\] (11)

If we consider \(K\) periods of the GNSS code, we can write \(T_0 = KT\), where \(T\) is the GNSS code period (1 ms in the present commercial GPS L1 signals). Then, the integral in the last term in the above equation may be simplified by exploiting the periodicity of the GNSS code over the integration period.

\[
\int_{T_0} c^2(t - \tau) dt = \sum_{i=1}^{K} \int_{(i-1)T}^{iT} c^2(t - \tau) dt = K \Phi_c
\] (12)

where
\[ \Phi_c = \int_{(i-1)T}^{iT} c^2(t - \tau)dt = \int_T c^2(t - \tau)dt = \int_T c^2(t)dt \]

because the GNSS code \( c(t - \tau) \) is T-periodic, so \( \Phi_c \) is a quantity independent of the period index \( i \) and the delay \( \tau \). It corresponds to the code power over one code period, i.e. \( T = 1023T \) for the current GPS and BOC(1, 1) signals.

The correlation of the received demodulated signal \( s(t) \), at frequency \( f_d \), with the receiver code \( c(t - \tau) \) yields \( R_{sc}(\tau) \),

\[ R_{sc}(\tau) = \int_{(i-1)T}^{iT} s(t)c(t - \tau)e^{-j2\pi f_d t}dt \]

where the subscript \( i \) refers to the integration time interval \([ (i-1)T, iT] \) and \( R_{sc}(\tau) \) is the cross-correlation function of the demodulated received signal with the local code replica over one GNSS-code period. Thus, the log-likelihood function can be written in the following compact form,

\[ \Gamma(\tau, A) = -\int_0^{T_0} |s(t)|^2 dt + 2R\{A^*R_{sc}(\tau)\} - K\Phi_c|A|^2 \]  

(14)

The cost function (14) can be minimized by setting the partial derivatives of \( \Gamma(\tau, A) \) to zero. The useful parameter for satellite positioning is the time-delay. However, the amplitude is required for delay estimate computation. The estimating equations to be solved are,

\[
\begin{cases} \\
\frac{\partial \Gamma(\tau, A)}{\partial \tau} = 2R\{A^*\frac{\partial R_{sc}(\tau)}{\partial \tau}\} = 0 \\
\frac{\partial \Gamma(\tau, A)}{\partial A} = 2R_{sc}(\tau) - 2K\Phi_c A^* = 0 \\
\end{cases}
\]

(15)

Accordingly,

\[ \hat{A} = \frac{R_{sc}(\tau)}{K\Phi_c} \]  

(16)

Contrary to the amplitude case, there is no explicit solution to the estimating time delay equation. This is because the total cross-correlation function \( R_{sc}(\tau) \) depends on the time delay parameter through \( c(t - \tau) \), which does not provide any direct expression of the delay. To overcome this difficulty, we implement the ML estimator iteratively using the Newton method. Due to space limitation, we present only the delay estimation algorithm, named FIMLA (for Fast Iterative Maximum Likelihood Algorithm).

\[ \hat{\tau}_{k+1} = \hat{\tau}_k - \frac{R\{R_{sc}^*(\hat{\tau}_k)\frac{\partial R_{sc}(\hat{\tau}_k)}{\partial \tau}\}}{R\{R_{sc}^*(\hat{\tau}_k)\frac{\partial^2 R_{sc}(\hat{\tau}_k)}{\partial \tau^2}\}} \]

(17)

In general, a Newton algorithm may converge to saddle points of maximum or minimum values. The point of convergence depends on the initial estimate. In GPS context, according to equation (14) the log-likelihood cost function depends on the delay through the cross-correlation function which has basically no local minima since it is of a triangle form. Surprisingly, the proposed algorithm converges to the correct maximum using a reasonable initial value. However, it is well known that the BOC waveform proposed for the future GNSS has a correlation function containing multiple peaks with magnitudes that are nearly equal to the magnitude of the central peak. This means that a signal tracker may lock onto the wrong peak, thus producing a tracking error. To overcome this difficulty, we suggest the initialization of the FIMLA algorithm by the result of the double-delta correlator [4, 8].

4. IMPLEMENTATION OF FIMLA IN A MULTIPATH-FREE ENVIRONMENT

In order to implement FIMLA, given by equation (17), we need to compute the first and second derivatives of the cross-correlation function \( R_{sc}(\tau) \). To do so, we propose to use the definition of a derivative function. If the function \( f \) is differentiable in point \( x \), then we have \( f'(x) = \lim_{\delta \to 0}[f(x + \delta) - f(x - \delta)]/2\delta \), and we may write

\[ f'(x) = \frac{f(x + \delta) - f(x - \delta)}{2\delta} + O(\delta^2) \]  

(18)

Identically, it is straightforward to show that the expression for the second derivative to be,

\[ f''(x) = \frac{f(x + \delta) + f(x + \delta) - 2f(x)}{\delta^2} + O(\delta^2) \]  

(19)

By neglecting the error term \( O(\delta^2) \), we obtain a good approximation for the first and second derivatives by finite differences. Of course, smaller values of \( \delta \) yield better approximation of \( f'(x) \) and \( f''(x) \). By applying this derivative approximations, we can summarize FIMLA as follows,

**Step 1. Initialization.** We initialize the algorithm by any adequate delay estimate method to compute an initial value \( \hat{\tau}_0 \).

**Step 2. Cross-correlation computation.** If the signal is sampled at \( f_c = 1/T_c = 2MHz \) and the number of samples is \( N \) (i.e. \( T_0 = NT_c \)), we compute the cross-correlation using:

\[ R_{sc}(\tau + kT_c) = \sum_{j=1}^{N} s(jT_c)e^{j(\tau + (k - j)T_c)} \]  

(20)

The sampling rate may be set \( T_c = T_e/2 \) for example.

**Step 3. Cross-correlation derivatives computation.** Using (18) and (19), we get

\[ \frac{\partial R_{sc}(\hat{\tau}_k)}{\partial \tau} = \frac{1}{2\delta}[R_{sc}(\hat{\tau}_k + \delta) - R_{sc}(\hat{\tau}_k - \delta)] \]

\[ \frac{\partial^2 R_{sc}(\hat{\tau}_k)}{\partial \tau^2} = \frac{1}{\delta^2}[R_{sc}(\hat{\tau}_k - \delta) + R_{sc}(\hat{\tau}_k + \delta) - 2R_{sc}(\hat{\tau}_k)] \]

**Step 4. k-th Iteration of FIMLA.** We update the iteration delay estimate using the Newton rule:

\[ \hat{\tau}_{k+1} = \hat{\tau}_k - \frac{R\{R_{sc}^*(\hat{\tau}_k)\frac{\partial R_{sc}(\hat{\tau}_k)}{\partial \tau}\}}{R\{R_{sc}^*(\hat{\tau}_k)\frac{\partial^2 R_{sc}(\hat{\tau}_k)}{\partial \tau^2}\}} \]  

(21)
5. FI MLA FOR MULTIPATH PARAMETER ESTIMATION IN GPS RECEIVERS

In this section, we extend the proposed FI MLA for GPS time-delay estimation to a multipath environment. Among all existing ML approaches for delay estimation in GPS literature, the paper [14] investigates the iterative approach, however, it provides little details and an ad-hoc implementations. Indeed, authors of this paper focused on the performance analysis and derive the Cramer-Rao Bound for multipath environments. The goal in our paper is to revisit and reanalyze the iterative ML approach in [14] and apply it to future GNSS receivers. We derive the mean equations and simplify the estimation algorithm by assuming constant amplitudes over the observation period. It results in an efficient algorithm using the existing DLL architecture, which is also very simple and easy to implement. In evaluating the multipath performance of the ML receiver, we deal with the BOC signals which have only been considered in the context of receiver correlators [6]. Simulations show that the results of this iterative ML algorithm, in a multipath environment, are similar to those of the conventional DLL in a single-path environment.

For simplicity of presentation, we consider only one reflected path superimposed on the direct-path signal, in which \( \phi_1 = \phi_2 = 2 \pi f_d t \), where \( f_d \) denotes the modulation frequency including the Doppler frequency. So, the received signal can be written as,

\[
s(t) = a_1 c(t - \tau_1) e^{j 2 \pi f_d t} + a_2 c(t - \tau_2) e^{j 2 \pi f_d t} + \eta(t) \tag{22}
\]

The time-delay and amplitude parameters are assumed to be constant over the observation time \( T_0 \). In the case of white Gaussian noise, the ML estimates are the delays and amplitudes values that maximize the probability density function (pdf),

\[
\text{exp} \left\{ - \frac{1}{2 \sigma^2} \int_{T_0} |s(t)|^2 dt \right\}
\]

By taking the logarithm of equation (23) and simplifying the results, the log-likelihood function can be written as,

\[
\Gamma(\tau_1, \tau_2, a_1, a_2) = - \int_{0}^{T_0} |s(t)|^2 dt + 2 \Re \{ a_1^* R_{sc}(\tau_1) \} - K |a_1|^2 \Phi_c + 2 \Re \{ a_2^* R_{sc}(\tau_2) \} - K |a_2|^2 \Phi_c - 2 \Phi(\tau - \tau_1) \Re \{ a_1 a_2^* \}, \tag{24}
\]

where

\[
\Phi(\tau) = \int_0^{T_0} c(t) c(t - \tau) dt
\]

is the GNSS-code autocorrelation function over the observation period \( T_0 = K T = K (1023 f_c) = K \) ms. To compute the maximum likelihood estimates, we set the partial derivatives of \( \Gamma(\tau_1, \tau_2, a_1, a_2) \) with respect to each of the four parameters to zero. The derivatives with respect to the amplitudes are temporary ignored to deal with the delay parameters. So, the two equations to be solved are:

\[
\begin{align*}
\frac{\partial \Gamma(\tau_1, \tau_2)}{\partial \tau_1} &= 0 \\
\frac{\partial \Gamma(\tau_1, \tau_2)}{\partial \tau_2} &= 0
\end{align*} \tag{25}
\]

Substituting (24) in (25) and replacing the correlations functions \( R_{sc}(\tau_1) \), \( R_{sc}(\tau_2) \) and \( \Phi(\tau - \tau_2) \) by their integral forms,

\[
\begin{align*}
\frac{\partial \Gamma(\tau_1, \tau_2)}{\partial \tau_1} &= 2 \Re \{ a_1^* R_{sc}(\tau_1) \} - 2 \Phi(\tau - \tau_1) \Re \{ a_1 a_2^* \} = 0 \\
\frac{\partial \Gamma(\tau_1, \tau_2)}{\partial \tau_2} &= 2 \Re \{ a_2^* R_{sc}(\tau_2) \} - 2 \Phi(\tau - \tau_1) \Re \{ a_1 a_2^* \} = 0
\end{align*}
\]

By using the fact that \( \Re(z) = \Re(z^*) \) for any complex number \( z \), we can write,

\[
\begin{align*}
\frac{\partial \Gamma(\tau_1, \tau_2)}{\partial \tau_1} &= 2 \Re \{ a_1^* R_{sc}(\tau_1) - a_2 \Phi(\hat{\tau}_2 - \hat{\tau}_1) \} = 0 \\
\frac{\partial \Gamma(\tau_1, \tau_2)}{\partial \tau_2} &= 2 \Re \{ a_2^* R_{sc}(\tau_2) - a_1 \Phi(\hat{\tau}_2 - \hat{\tau}_1) \} = 0
\end{align*}
\]

Accordingly,

\[
\begin{align*}
\frac{\partial \Gamma(\tau_1, \tau_2)}{\partial \tau_1} &= \Re \{ a_1^* \int_{T_0} s(t) e^{-j 2 \pi f_d t} - a_2 c(t - \tau_2) c(t - \tau_2) dt \} = 0 \\
\frac{\partial \Gamma(\tau_1, \tau_2)}{\partial \tau_2} &= \Re \{ a_2^* \int_{T_0} s(t) e^{-j 2 \pi f_d t} - a_1 c(t - \tau_1) c(t - \tau_2) dt \} = 0
\end{align*}
\]

Equivalently, we can write:

\[
\begin{align*}
\Re \{ a_1^* \frac{\partial}{\partial \tau_1} \int_{T_0} s(t) e^{-j 2 \pi f_d t} - a_2 c(t - \tau_2) c(t - \tau_2) dt \} &= 0 \\
\Re \{ a_2^* \frac{\partial}{\partial \tau_2} \int_{T_0} s(t) e^{-j 2 \pi f_d t} - a_1 c(t - \tau_1) c(t - \tau_2) dt \} &= 0
\end{align*}
\]

More compactly, the delays estimating equations are given by,

\[
\begin{align*}
\frac{\partial \Gamma(\tau_1, \tau_2)}{\partial \tau_1} &= 2 \Re \{ a_1^* R_{sc}(\tau_1) \} = 0 \\
\frac{\partial \Gamma(\tau_1, \tau_2)}{\partial \tau_2} &= 2 \Re \{ a_2^* R_{sc}(\tau_2) \} = 0
\end{align*} \tag{26}
\]

where

\[
R_{sc}(\tau_1) = \int_{T_0} s(t) e^{-j 2 \pi f_d t} - a_2 c(t - \tau_2) c(t - \tau_2) dt
\]

\[
R_{sc}(\tau_2) = \int_{T_0} s(t) e^{-j 2 \pi f_d t} - a_1 c(t - \tau_1) c(t - \tau_2) dt
\]

The main issue is how to efficiently implement equations (26). Both equations are similar to their counterparts the multipath-free case (15). Then, an efficient strategy to resolve (26) suggests a sequential implementation of the FI MLA developed above.

5.1. Amplitudes estimation using FI MLA

According to the previous equations, the time-delay estimation requires the amplitudes values, which are unknown in real scenarios. The corresponding estimating equations related to the amplitudes parameters \( a_1 \) and \( a_2 \), are

\[
\begin{align*}
\frac{\partial \Gamma(\tau_1, \tau_2, a_1, a_2)}{\partial a_1} &= 0 \\
\frac{\partial \Gamma(\tau_1, \tau_2, a_1, a_2)}{\partial a_2} &= 0
\end{align*} \tag{27}
\]
Computing the above derivatives using equation (24), provide
\[
\begin{align*}
\frac{\partial \tilde{r}(\tau_1, \tau_2, a_1, a_2)}{\partial a_1} & = R_c^* (\tau_1) - \Phi (\tau_2 - \tau_1) = 0 \\
\frac{\partial \tilde{r}(\tau_1, \tau_2, a_1, a_2)}{\partial a_2} & = R_c^* (\tau_2) - \Phi (\tau_2 - \tau_1) = 0
\end{align*}
\]

Thus, the amplitude estimates can be expressed as,
\[
\begin{align*}
\tilde{a}_1 & = \frac{\Phi (\tau_2 - \tau_1) R_c (\tau_2 - \tau_1)}{\Phi (\tau_2 - \tau_1)^2 - \Phi_c^2} \\
\tilde{a}_2 & = \frac{\Phi (\tau_2 - \tau_1) R_c (\tau_2 - \tau_1)}{\Phi (\tau_2 - \tau_1)^2 - \Phi_c^2}
\end{align*}
\]
(28)

We note that the time-delay estimates used in the above equation (28) are the estimates computed in the previous iteration in the output of the delay lock loop. For the implementation purpose, we express the amplitude parameter estimates using the real parts $Q_P$ and the imaginary parts $I_P$ of the inter-correlation between the demodulated received signal and the code phase replicas, as detailed next. Since,
\[
\begin{align*}
R_c (\tau_1, k) & = \int_{T_0} s(t) c(t - \tau_1, k) e^{-j2p_0 f_d t} dt \\
R_c (\tau_2, k) & = \int_{T_0} s(t) c(t - \tau_2, k) e^{-j2p_0 f_d t} dt
\end{align*}
\]
(29)

the amplitude estimates are computed as follows,
\[
\begin{align*}
\tilde{a}_{1, k+1} & = g_k [\Phi (\tau_2 - \tau_1, k) Q_P (\tau_1, k) - \Phi (\tau_1, k)] \\
& + j g_k [\Phi (\tau_2 - \tau_1, k) I_P (\tau_1, k) - \Phi (\tau_1, k)] \\
\tilde{a}_{2, k+1} & = g_k [\Phi (\tau_2 - \tau_1, k) Q_P (\tau_1, k) - \Phi (\tau_1, k)] \\
& + j g_k [\Phi (\tau_2 - \tau_1, k) I_P (\tau_1, k) - \Phi (\tau_1, k)]
\end{align*}
\]
(30)

where $g_k = \Phi (\tau_2 - \tau_1, k)^2 - \Phi_c^2$ is the denominator term.

5.2. Delay estimates computation using FIMLA

Similar to the single-path case, the iterative algorithm for time-delay estimation is given by the following equations
\[
\begin{align*}
\hat{\tau}_{1, k+1} & = \hat{\tau}_{1, k} - \frac{\partial R_c^{(1)} (\tau_1, k)}{\partial \tau_1} \\
\hat{\tau}_{2, k+1} & = \hat{\tau}_{2, k} - \frac{\partial R_c^{(2)} (\tau_2, k)}{\partial \tau_2}
\end{align*}
\]
(31)

(32)

This algorithm can be easily generalized to the several multipath case by updating each path-delay by the same FIMLA algorithm using the corresponding cross-correlation function after subtracting the other paths contributions in a sequential procedure.

6. SIMULATION STUDY

In searching for optimal estimators for a given parameter $\theta$, we need to adopt some optimality criterion. A natural one is the mean square error (MSE), defined as
\[
MSE(\hat{\theta}) = E \left\{ (\hat{\theta} - \theta)^2 \right\}
\]
(33)

where $Var(\hat{\theta})$ denotes the variance of the estimator and $B(\hat{\theta})$ is the corresponding bias. This measures the average mean square deviation of the estimator from the true value. To keep the figures simple, only positive bias errors are shown for an in-phase relationship between primary and secondary paths. We use the variances, the bias, and the Root MSE (RMSE) as performance metrics. It should be mentioned that we compare MEDLL, FIMLA and the Narrow Correlator algorithms using a pre-correlation bandwidth of 24 MHz as typically used in the literature.

- Multipath error of FIMLA in one multipath case using current C/A-code GPS signals. In this experiment, we consider a two-path model in which the two signals have the same Doppler frequency and corrupted by a complex Gaussian noise. In this experiment, the parameters have been set as follows. The signal-noise ratio (SNR) = -20dB, integration time $T_0 = 10$ ms, and the power of the direct signal to the multipath ratio (SMR) is SMR = 5dB. To achieve good performance, we combine the maximum-likelihood with the narrow-correlator ones by implementing the FIMLA using narrow cross-correlations output by choosing $\delta = 0.25T_c$. Note that a smaller value can be used to reduce the multipath-bias and then obtaining better performance but we remark that this will increase slightly the estimate variance. From extensive simulations tests, we conclude that a best value of $\delta$ is between $\delta = 0.1T_c$ and $\delta = 0.25T_c$. Figure 1 shows the bias and variance of the delay estimate using FIMLA with comparison to the widely implemented non-coherent narrow correlator DLL. Definitely, the multipath effect was greatly reduced using the proposed algorithm.

- Multipath error of FIMLA in two multipath case using next generation of GNSS receivers with BOC signals. In this experiment, we consider also the future BOC-based GNSS signals in the performance comparison. Here, we consider a more complicated scenario when two multipath signals are received by the GPS receiver with powers according to the following SMRs : 6 dB and 15 dB. The second multipath is held static at 0.5 chips delay. Figure 2 plots the RMSE performance index versus the relative delay to the first multipath for the proposed FIMLA algorithm using BPSK-R(1) signals (FIMLA-BPSK), FIMLA using BOC(1,1) signals, MEDLL receiver [16] and the non-coherent narrow-correlator (NCNC). We observe that the MEDLL receiver suffer a degradation in accuracy due to the addition of one more multipath signal while FIMLA is more robust to additional multipath reflections due to the efficient procedure that is based on. Both has
Fig. 1. Delay estimate bias $|E(\hat{\tau}_1 - \tau_1)|$ and variance $\text{Var}(\hat{\theta})$ versus multipath delay $(\tau_2 - \tau_1)$.

Fig. 2. RMSE in presence of three multipath signals.

significantly better performance than the standard correlator receiver. However, the FIMLA algorithm using BOC(1,1) signals shows better overall multipath performance since it is less sensitive to medium-delay multipath. Indeed, for a path delay of approximately 120 m, the resulting ranging error for the FIMLA-BOC(1,1) decreases to zero.

7. CONCLUSION

We have proposed an efficient implementation of the Newton method for ML synchronization in the context of GNSS positioning in multipath environments. Analytic derivations and simulations experiments show that the developed FIMLA algorithm may be an attractive and fast candidate for iterative resolution of the ML multipath mitigation problem in GNSS receivers. Both current C/A-code GPS and the future BOC(1,1) GNSS signals have been considered.

8. REFERENCES