Multi-millisecond GNSS Maximum Likelihood Bit Synchronization Method

Gerardo Ludovico Puga, Pedro Agustín Roncagliolo, and Javier Gonzalo García
Laboratorio de Electrónica Industrial, Control e Instrumentación (LEICI),
Dto. Electrotecnia, Facultad de Ingeniería, UNLP, La Plata, Argentina.
Emails: { gerardo.puga, agustinr, jgarcia}@ing.unlp.edu.ar

***Abstract*—This paper introduces a new maximum likelihood bit synchronization algorithm that can be tailored to be used with different types of GNSS signals (GPS, GLONASS). A preliminary version of this work was presented in a previous paper by the same authors. In that paper, it was shown that this method allows to determine the positions of the bit edges in the signal using multiple millisecond correlation values. Using longer correlation times for bit synchronization decreases the peak processor load on embedded GNSS receivers that are based on carrier and code tracking loops for signal tracking, since for this kind of processing, longer correlation times allow for lower tracking loop update rates. In that previous paper it was also shown (through the use of simulations) that this new method presented lower error rates compared to the histogram method for fixed length observation intervals. The present paper extends these results in two ways. First, it generalizes the proposed method to N-millisecond integrations, providing a framework for automatic generation of the Viterbi state machine that drives the synchronization. Second, this paper provides a simple analytical expression that can be used to estimate the synchronization error rate of the algorithm, and identifies its most important design parameters.**

***Keywords*-GPS; GLONASS; receiver; data modulation**

I. INTRODUCTION

Global Navigation Satellite Systems (GNSS) enable any vehicle equipped with a receiver to determine its position and velocity based on the signals transmitted by a constellation of satellites placed in carefully controlled orbits. For each signal present in the antenna the receiver determines a pseudorange measurement, which is a quantity related to the true geometrical range (distance) between the receiver and the transmitting satellite. Using four or more of these measurements, the receiver can determine its spatial coordinates with an error in the order of a few meters. User velocity determination is performed in a similar fashion, using quantities related to the Doppler deviation of the signal carrier frequency.

There are currently two GNSS systems in full operational status: GPS, the GNSS System maintained by the United States Department of Defense (DoD), and its russian counterpart GLONASS. There are other GNSS systems being developed by different national entities, but as of 2012 these are still on the planning stage or have been only partially deployed (COMPASS, Galileo). A detailed description of the signal of GPS and GLONASS can be found in the literature [1][2][3][4].

Both GPS and GLONASS satellites transmit data at 50 bps. GPS encodes the bits using bipolar NRZ pulses, while GLONASS uses Manchester coding. Before being able to decode the data frames transmitted by a satellite, the receiver needs to perform a synchronization process during which the position of the data bit edges is determined.

Because of the structure of the signal, there is only partial uncertainty in the knowledge of the position of the bit edges. GPS and GLONASS signals are spread using a DS-SS (Direct Sequence Spread Spectrum) periodic spreading sequence. This spreading sequence is 1023 chips long for GPS, and 511 chips long for GLONASS. In both cases the spreading sequence repeats itself after 1 ms. Data bit sign transitions are aligned with the start of the new period of the spreading sequence. One data bit pulse extends exactly 20 spreading sequence periods. Since the start of the each period of the spreading sequence is known to the receiver, then there are only 20 different possibilities for the data bit sequence alignment. During bit synchronization, the receiver correlates the signal with the periodic spreading sequence using integration intervals that extend over an integer number of spreading sequence periods, and processes the results in order to detect the position of the data bit edges.

The classic algorithm for GPS data bit synchronization is the histogram method [2][8]. This method searches for sign changes in consecutive 1 ms correlation results. The performance is adequate for carrier-to-noise-density C/N_0 (which is a quantity related to the signal-to-noise ratio) ratios above 30 dB, which is the normal operating condition for most outdoor GNSS receivers. Because of its simplicity, this method has been extensively used for general purpose embedded receivers.

There are other more complex methods that present higher sensitivity, allowing receivers to perform data bit synchronization under extremely low C/N_0 conditions such as those endured by GPS receivers for indoor applications and street level car receivers. These algorithms work by finding the bit edge candidate position that maximizes the recovered average bit energy [6][7][10], which is equivalent to choosing the maximum likelihood candidate [6]. These methods can work with C/N_0 levels down to 12 dB [7].

Slightly modified versions of these algorithms can be used for GLONASS. The modifications are necessary because

 ISBN: 978-1-61208-264-6

GLONASS uses Manchester coding for its data bits instead of bipolar NRZ.

Both the histogram and the energy maximization algorithms need to work on correlation samples obtained using 1 ms integration times in order to determine the bit alignment with millisecond level resolution. This means that the phase and code tracking loops in the receiver must be updated at a rate of 1 kHz in order to determine the correlation parameters (carrier initial phase and frequency, spreading sequence initial phase and chip rate) that need to be configured at the start of each correlation. This update rate is higher than the update rate typically used during later processing stages (100 Hz), but the amount of processing per update is roughly the same in both cases. Thus, the synchronization stage of GNSS receivers that use these algorithms represents an important fraction of the worst-case processor load of the design. This worst-case processor load is an important metric for real-time systems since it determines whether the system will be able to fulfil all of its processing deadlines in due time [5]. Also, since the synchronization stage represents a very small portion of the total processing time of a given GNSS signal, using the histogram, the bit energy maximization, or any other method that works on 1 ms correlation samples can lead to receiver designs with high peak-to-mean processor load ratios.

The worst-case processor load for a given design can be reduced using data bit synchronization algorithms that can achieve millisecond resolution using multi-millisecond correlation samples that require lower tracking loop update rates. This improvement in the worst-case processor load should not come at the expense of a decrease in the reliability of the synchronization algorithm.

In a previous work [11], a 3 ms maximum likelihood bit synchronization algorithm was proposed. The algorithm was simulated and compared against two other methods (histogram and a maximum bit energy algorithm similar to the one shown in [6]) on the grounds of error rates of each method for fixed size observation intervals.

The present paper extends these results in two ways. First, it generalizes the proposed method to N-millisecond correlation samples, providing a set of tools to help the design of the Trellis diagram that guides the synchronization state machine transitions. Second, this paper provides a simple analytical expression that can be used to estimate the synchronization error rate of the algorithm, and identifies its most important design parameters.

It is important to mention that while our previous work was aimed at achieving bit synchronization of 50 bps data signals of GLONASS and GPS, the present paper focuses on bit synchronization of a generic 100 bps bipolar NRZ. This is because both GPS and GLONASS signals can be thought as 100 bps bipolar NRZ signals whose data bits are encoded with a repetition code, and thus this 100 bps synchronization can be used as the first stage of a full 50 bps GPS/GLONASS bit synchronization algorithm. This is the approach used in a dual system GPS/GLONASS receiver that is being developed by the GNSS group in the Facultad de Ingeniería of the Universidad Nacional de La Plata.

The rest of this paper is organized as follows. Section II presents a basic baseband signal model that can be used to work with N-millisecond correlation samples and enunciates a few base assumptions. Later, the fundamental ideas of the maximum likelihood synchronization method are presented. Section III presents several simple rules to automatically build the synchronization Viterbi state machine for any given value of N. These rules were developed in order to code the simulations that were in turn used to validate the analytical expression of the probability of synchronization error that is introduced in Section IV. Simulation results are compared against theoretically calculated values of the probability of synchronization error in Section V. Finally, the conclusion is in Section VI.

II. Algorithm

A. N-millisecond Correlation Samples Model

The following sequence $I_1[i]$ can be used as the model of 1 ms correlation values produced by a GNSS receiver when processing a signal with data encoded as a sequence of 100 bps bipolar NRZ symbols.

$$I_1[i] = B[\lfloor (i+m)/10 \rfloor] + n_1[i], \tag{1}$$

where $B[n]$ is a random process that models the random bit sequence ($B[n] = \pm 1$), m is the unknown initial bit alignment (in milliseconds), $n_1[i]$ is random Gaussian random variable, independent for each i, and such that $E\{n_1[i]\} = 0$, $Var\{n_1[i]\} = \sigma_1^2$.

This model assumes that the beginning of the 1 ms correlation intervals are aligned so that they differ from each and every data bit edge by an integer number of milliseconds. This is a safe assumption since typically GNSS receivers are free to arrange the correlation intervals in the most convenient way, and the 1 ms correlation interval duration fits an integer number of times within the bit duration.

The previous model also requires some sort of normalization of the amplitude of the samples so that $E\{|B[n]|\} = 1$.

Samples generated when using N-millisecond correlation times can be modeled as

$$I_N[j] = \sum_{a=0}^{N-1} I_1[N\,j + a]. \tag{2}$$

Based on the definition of $I_1[i]$

$$I_N[j] = S[j] + n_N[j], \tag{3}$$

where $n_N[j] = n_1[Nj] + ... + n_1[Nj + N - 1]$ is also a Gaussian random variable such that $E\{n_N[i]\} = 0$, and $Var\{n_N[i]\} = N\sigma_1^2$. The term $S[j]$ is the integrated value of the data signal during the N-milliseconds interval. It can be seen that depending on the presence and position of a data bit sign change during the correlation interval, the value of this term will be $S[j] = \pm(N-2k)$, where $k \in [0, N]$ is the position of the data bit sign change in milliseconds, relative to the start of the integration time.

B. Synchronization

In this section, we will skim over the main ideas of the synchronization algorithm since they were already discussed in [11].

Given a data bit sequence $B[n]$ and an alignment m, the sequence of N-millisecond correlation values $S[j]$ that will be observed is completely determined. This is also true the other way around: given a sequence of L observed correlation samples $I_N[j]$, the data bit sequence $\tilde{B}[n]$ during the observation interval and the bit alignment $\tilde{m}$ can be estimated if we find the sequence of expected correlation values $\tilde{S}[j]$ that best matches $I_N[j]$.

Using the Maximum Likelihood criterion it can be shown that the highest probability candidate $\tilde{S}[i]$ is the one that minimizes the log-likelihood index J

$$\tilde{S}[i] = \min_{o} J(I_N, S^{(o)}), \tag{4}$$

which is defined as

$$J(I_N, S^{(o)}) = \sum_{a=0}^{L-1} (I_N[a] - S^{(o)}[a])^2. \tag{5}$$

In order to find the maximum likelihood candidate a Viterbi algorithm is used. One state is defined for each possible alignment of the beginning of the next N-millisecond correlation relative to the start of the current bit interval and for each sign. Since there are 10 possible different alignments and two bit values (± 1), 20 states need to be defined. Notice that the size of the state machine is independent of the length of the correlation interval N.

The state numbering scheme that will be used throughout this paper is shown in Figure 1. While any kind of state naming scheme is possible, this scheme is particularly useful because it allow us to enunciate a few very simple rules to automate the creation of the state machine transitions for any correlation length N.

There are $20+2N$ transitions in this state machine. States that are closer to the start of the bit than a correlation interval length have two possible arrival paths from previous states; the rest is only linked to a single previous state. Figure 2 shows the Trellis diagram for the case when $N = 3$. In Section III, the rules used to automate the creation of this Trellis will be given.

Initially each state has no accumulated quadratic error. For each observed integration value $I_N[j]$, the quadratic errors of the observed values against the expected values along each possible transition within the Trellis are calculated and added to the accumulated quadratic error of the state at the

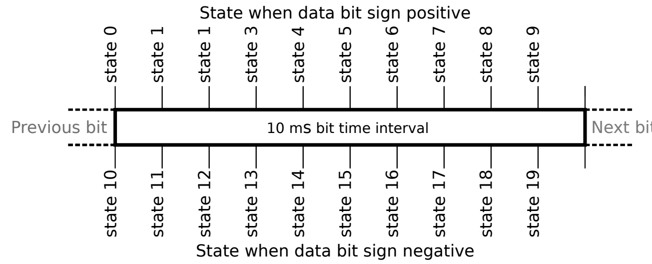

Figure 1. States are defined one for each possible alignment of the beginning of next N-millisecond correlation relative to the start of the current bit interval, and for each possible sign.

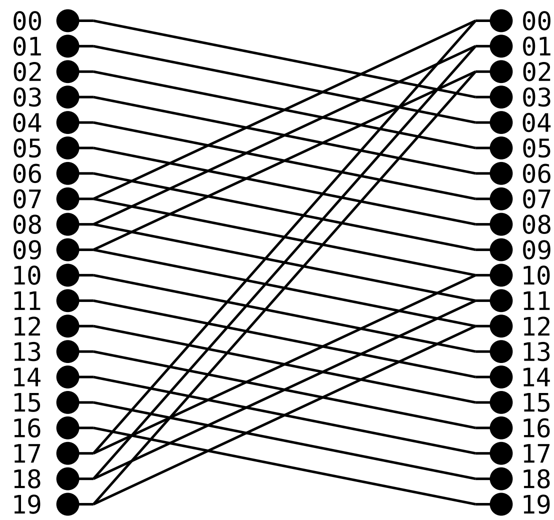

Figure 2. State transition map for $N = 3$.

origin of this transition. If there is a single input path to the destination state of the transition, this updated accumulated quadratic error becomes the accumulated quadratic error of the destination state. If instead there are two possible arrival paths to the destination state, then the path with the highest updated accumulated quadratic error can be safely discarded since it can be anticipated that no matter what the future sequence of observed values is, in the end this candidate sequence will have a total accumulated error higher than at least one other candidate. The surviving path at the merge point then determines the quadratic error of the destination state. This way, at each step $2N$ candidates are eliminated at the merge points in the Trellis, and $2N$ new candidates are created at the fork points. After the last input sample $I_N[j]$ has been processed, the maximum likelihood observed sequence $\tilde{S}[j]$ can be retrieved from the transition history of the candidate with the least accumulated error among the 20 survivors. If only the maximum likelihood alignment estimation is required, all the information needed is in the index number of the state with the least error: if s is the state number with the least error, then at the start of the next correlation interval $mod(s, 10)$ milliseconds will have

 ISBN: 978-1-61208-264-6

passed after the last bit edge position.

III. Trellis Construction Rules

Because of the regularity of the state transitions, the Trellis for any value $N \in [2, 10]$ can be generated using a few very simple rules.

Let st be the state number, using the same numbering scheme shown in Figure 1. Each state has at least one input transition that comes from the st_{ss}-*th* state

$$st_{ss} = mod(10 + st - N, 10) + 10\lfloor st/10 \rfloor, \quad (6)$$

where mod is the remainder of the integer division. The expected observed value along this transition is

$$E_{ss} = (-1)^{\lfloor st/10 \rfloor} N. \quad (7)$$

If $mod(st, 10) < N$ there is a second input transition, from the st_{ds}-*th* state

$$st_{ds} = mod(st_{ss} + 10, 20). \quad (8)$$

This transition models the observed correlation value when there is a data bit sign change from the previous bit to the current one. Because of the sign change, the integrated areas at each side of the bit edge partially cancel each other, and thus the expected observed value depends on the alignment the the state relative to the bit start

$$E_{ds} = (2\, mod(st, 10) - N)\, (-1)^{\lfloor st/10 \rfloor}. \quad (9)$$

These rules were used to generate the trellis in Figure 2.

IV. Error expression

Figure 3 shows the values of the final accumulated quadratic errors of the 20 states after having processed a signal with carrier-to-noise-density (C/N_0) ratio of 35 dB using 5 ms integrations ($N = 5$). Each state has an associated candidate correlation value sequence that can be recovered from the history of transitions within the Trellis. Pairs of states s and $s+10$ are both associated to candidates with the same bit alignment. These pairs of states also have similar accumulated quadratic errors because of the way the Viterbi algorithm forks and merges candidates as it moves forward.

The state with the least accumulated quadratic error determines the maximum likelihood bit alignment. Accumulated quadratic errors grow higher the farther away we move from the state with the maximum likelihood solution. The reason for this is that the surviving candidates associated to state numbers close to the one with the least error are not random but are in fact very similar to the maximum likelihood candidate. The farther away from the maximum likelihood candidate that we move, the smaller the amount of likeliness, and thus the higher the accumulated error.

This observation about the likeliness of the surviving candidates is not only qualitative. For high enough signal-to-noise ratios, it can be safely assumed that the second and third most likely alignments are associated to candidate correlation sequences that followed the same bit sequence than the maximum likelihood correlation sequence, but whose bit edges are displaced ±1ms from the true value.

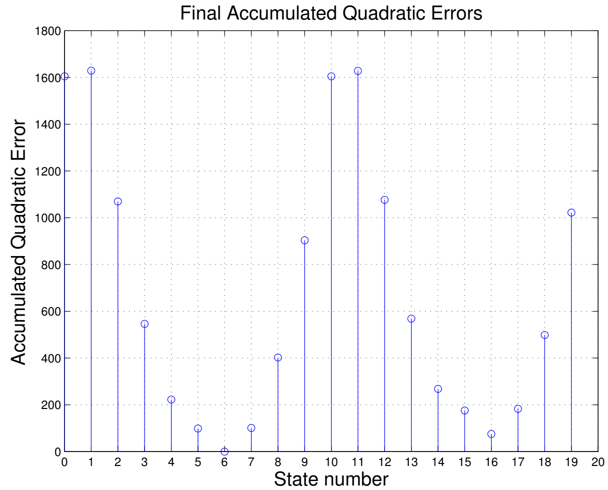

Figure 3. Typical accumulated errors distribution. This plot was generated performing a test run of the simulated algorithm with $N = 5$ and $C/N_o = 35$ dB.

This, in turn, can be used to justify the following statement: the probability of incorrectly identifying the bit edge position using this algorithm is the probability of choosing instead a candidate sequence whose alignment differs by ±1 ms from the correct value.

Let S^α be the correct candidate for the received data bit sequence. Let S^β be the candidate that would have been correct if the data bit sequence was the same, but the bit edges were delayed by 1 ms. The probability of mistakenly choosing S^β over S^α as the maximum likelihood candidate is the probability of the event

$$\begin{aligned} J(I_N, S^\alpha) &> J(I_N, S^\beta) \\ \sum_{a=0}^{L-1} (I_N[a] - S^\alpha[a])^2 &> \sum_{a=0}^{L-1} (I_N[a] - S^\beta[a])^2. \end{aligned} \quad (10)$$

Since the bit sequences that generated S^β and S^α only differ by 1 ms, the sequences S^β and S^α differ only at F sequence values, one for each data bit sign change that occurred (since N $\leq$ 10, there can be at most a single bit sign transition during each correlation interval). Many terms then cancel out, leaving

$$\sum^{F} (I_N[a] - S^\alpha[a])^2 > \sum^{F} (I_N[a] - S^\beta[a])^2. \quad (11)$$

It is easy to see that since the original data bit sequences only differ by a single millisecond, the following equality holds for the remaining terms:

$$S^\alpha[a] - S^\beta[a] = \pm 2, \quad (12)$$

 ISBN: 978-1-61208-264-6

and thus

$$\sum^{F}(I_N[a]-S^{\alpha}[a])^2 \quad > \quad \sum^{F}(I_N[a]-S^{\alpha}[a]\mp 2)^2.$$

Replacing $I_N[a]=S^{\alpha}[a]+n_N[a]$:

$$\begin{aligned}\sum^{F}(n_N[a])^2 \quad &> \quad \sum^{F}(n_N[a]\mp 2)^2\\ 0 \quad &> \quad \sum^{F}(\mp 4n_N[a]+4)\\ \sum^{F}\pm n_N[a] \quad &> \quad F. \end{aligned} \qquad (13)$$

$n_N[a]$ are independent gaussian random varibles, thus:

$$P\left\{\sum^{F}\pm n_N[a]>F\right\} \quad = \quad Q\left(\frac{\sqrt{F}}{\sigma_N}\right), \qquad (14)$$

where $Q(x)$ is the probability that a Gaussian random variable will obtain a value larger than x standard deviations above the mean.

Finally, since there are two candidates that differ by exactly 1 ms among the surving candidates of the Viterbi processing, the final synchronization error expression is:

$$P_e \quad = \quad 2Q\left(\sqrt{\frac{2\,F\,\frac{C}{N_o}}{1000N}}\right), \qquad (15)$$

where in the last expression $\sigma_N^2=N\sigma_1^2$ was replaced by its expression as a function of C/N_0.

It can be seen that the most important parameters in order to determine the synchronization error rate are the number of data bit sign transitions observed F, the length of the correlation interval N and the carrier-to-noise-density C/N_0 ratio of the signal. While the value N is usually imposed by the constraints in other parts of the receiver (e.g. the update rates of the tracking loops), and the C/N_0 ratio is an external constraint imposed on the system by the environment, the designer can choose to increase or decrease the probability of error by choosing the minimum number of bit sign changes that need to be observed during synchronization before deciding the most probable bit alignment of the signal.

Using F as a design variable imposes a compromise between the error rate and the time it takes to perform synchronization. GLONASS signals use Manchester coding, which inserts a forced sign transition during the bit time. Thus it is safe to expect between 50 and 100 bit sign changes each second. GPS, on the other hand, encodes the bits using NRZ; because of this the expected data bit sign changes will be in the range between 0 and 50 each second.

V. Simulations

Figure 4 shows the probability of synchronization error as a function of the correlation integration length N for $N=2,3,5,10$ and the signal carrier-to-noise-density ratio, C/N_0. The values of probability of error calculated using (15) are also shown in the same figure for comparison. For these simulations, random 100 bps BPSK bit sequences were simulated, each one with F bit sign transitions and random initial edge alignments. The value of F was chosen to be 25, which is often taken as the expected number of bit sign transitions during each second for GPS signals (the expected number for GLONASS is 75).

It can be seen that the theoretical values predicted by (15) match the probability of error obtained during simulations for values of C/N_0 higher than 27 dB. This range includes the operating range for most outdoor GNSS receivers. Theoretical and simulated curves start to differ for values below that because, as the signal-to-noise ratio drops, the base hypothesis about the probability of incorrectly identifying the bit edge position being equal to the probability of mistakenly choosing a candidate whose alignment differs by ± 1 ms from the correct value stops being reliable.

VI. Conclusion

This paper introduced a new maximum likelihood bit synchronization algorithm that can be tailored to be used with different types of GNSS signals (GPS, GLONASS). A preliminary version of this work was presented in a previous paper by the same authors. In that paper, it was shown that this method allows to perform bit synchronization using 3 ms correlation interval samples, while at the same time it improves the probability of error compared to the histogram method.

The present paper expands our previous work by exploring the use of different correlation lengths N. As a test case, we work on the bit synchronization of a 100 bps NRZ signal, since that is the kind of synchronization that is used during the first stage of bit edge disambiguation phase of a dual-system GPS/GLONASS receiver being developed in the Facultad de Ingeniería of the Universidad Nacional de La Plata.

An analytical expression to calculate the probability of synchronization errors is also provided in this paper. This expression relates the probability of error to the different parameters of the algorithm, providing the designer with valuable information during the development and testing stage of a receiver. The expression was verified using simulations, and proved to be accurate in typical use scenarios.

It is important to state that the algorithm itself and the results herein presented are in no way limited to the kind of codification and data rate presented in this paper. If a designer found more convenient to use specialized algorithms for each type of signal (50 bps NRZ GPS and 50 bps Manchester coded GLONASS signal) on a dual-system receiver, or in the design of an only GPS or only GLONASS receiver, the results provided in this paper could be reused with only slight modifications, e.g., using larger

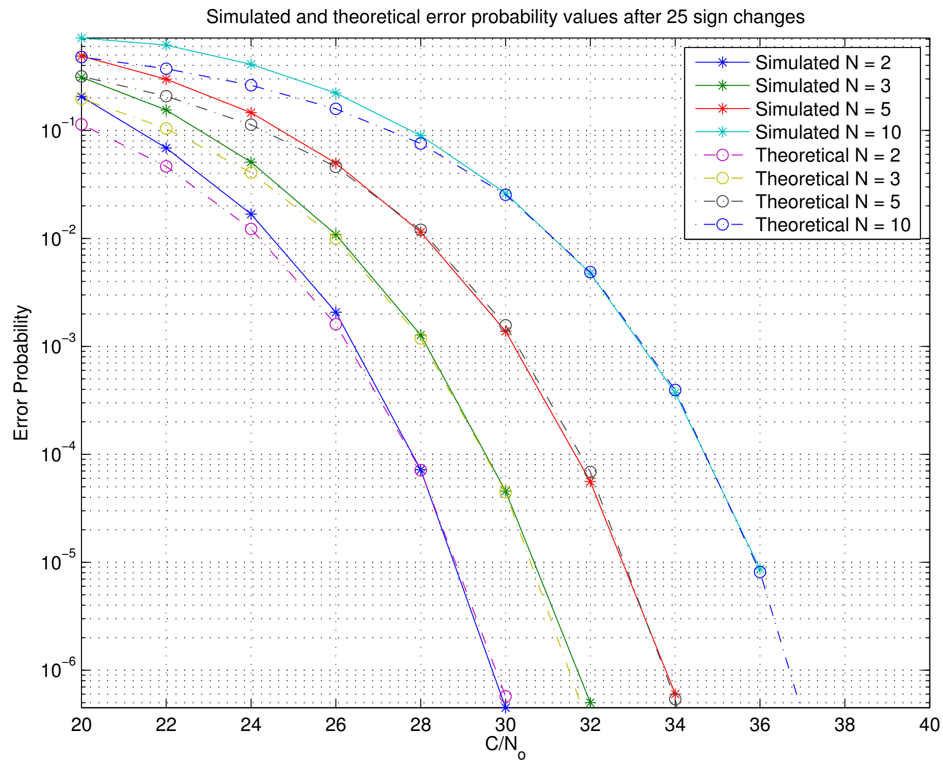

Figure 4. Probability of synchronization error for $F = 25$, and correlation integration lengths $N = 2$, $N = 3$, $N = 5$, $N = 10$. In solid line: simulation results; dashed lines: theoretical values using (15).

state machines (40 states for 50 bps signals) and specialized state machine transitions for each GPS and GLONASS. Such was the approach in our previous work [11].

REFERENCES

[1] E. Kaplan, "Understanding GPS: Principles & Applications". Artech House, 1996.

[2] B. W. Parkinson and J. J. Spilker Jr., "Global Positioning System: Theory and Applications", Volume I. American Institute of Aeronautics and Astronautics (AIAA), 1996.

[3] "NAVSTAR Global Positioning System - Interface Specification IS-GPS-200". NAVSTAR GPS Joint Program Office, 2006.

[4] "Global Navigation Satellite System - GLONASS - Interface Control Document", English edition, version 5.1. Russian Institute of Space Device Engineering, 2008.

[5] C. L. Liu and J. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment". Journal of the ACM Journal of the ACM (JACM), ACM New York, NY, USA, January 1973, vol. 20, issue 1, pp. 46-61.

[6] M. Kokkonen and S. Pietila, "A New Bit Synchronization Method for a GPS Receiver". In: Position Location and Navigation Symposium, 2002 IEEE, Palm Springs, California, April 2002, pp. 85-90.

[7] Tao Z. and Gannan Y., "A New Bit Synchronization Method for an Ultra-Tightly Integrated GPS Receiver". In: Second International Conference on Information and Computing Science (ICIC '09), Manchester, England, May 2009, vol. 1, pp. 239-242.

[8] Sichao L., Jinhai S., Jinhai L., and Yuepeng Y., "A modified histogram bit synchronization algorithm for GNSS receivers". In: 2nd International Conference on Information Science and Engineering (ICISE), Hangzhou, China, December 2010, pp. 1720-1723.

[9] M. Anghileri, T. Pany, Jong-Hoon W., and G. W. Hein, "An Algorithm for Bit Synchronization and Signal Tracking in Software GNSS Receivers". In: the 19th International Technical Meeting of the Satellite Division (ION GNSS 2006), Institute of Navigation, Fort Worth, Texas, USA, September 2006, pp. 1836-1848.

[10] N. I. Ziedan and J. L. Garrison, "Bit Synchronization and Doppler Frequency Removal at Very Low Carrier to Noise Ratio Using a Combination of the Viterbi Algorithm with an Extended Kalman Filter". In: Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003), Portland, Oregon, USA, September 2003, pp. 616-627.

[11] G. L. Puga, P. A. Roncagliolo, and J. G. Garcia, "Low Computational Cost GNSS GPS/GLONASS Maximum Likelihood Bit Synchronization Method". In: Proceedings of the 13th Argentine Symposium on Technology (JAIIO/AST 2012), La Plata, Buenos Aires, Argentina, August 2012, pp. 204-215.