Fast Computation of Maximum Time Interval Error by Binary Decomposition

Stefano Bregni*, *Member, IEEE*, and Stefano Maccabruni**

*CEFRIEL, Via Fucini 2, 20133 Milano MI, ITALY
Ph.: +39-02-23954.265, Fax: +39-02-23954.254, E-mail: bregni@cefriel.it, WWW: http://www.cefriel.it/~bregni
**Politecnico di Milano, Dipartimento di Elettronica e Informazione, ITALY

Abstract — **Maximum Time Interval Error (MTIE) is historically one of the main time-domain quantities for the specification of clock stability requirements in telecommunications standards. Nevertheless, plain computation of the MTIE standard estimator proves cumbersome in most cases of practical interest, due to its heavy computational weight. In this paper, MTIE is first introduced according to its standard definition. Then, a fast algorithm based on binary decomposition to compute the MTIE standard estimator is described. The computational weight of the binary decomposition algorithm is compared to that of the estimator plain calculation, showing that the number of operations needed is reduced to a term proportional to $N\log_2 N$ instead of N^2. A heavy computational saving is therefore achieved, thus making feasible MTIE evaluation based on even long sequences of Time Error (TE) samples. The algorithm proposed is finally applied to TE sequences generated by simulation of all the types of power-law noise, in order to check its effectiveness and correctness.**

Index Terms — **Clocks, digital communication, jitter, SONET, synchronization, synchronous digital hierarchy, time domain measurements.**

I. Introduction

A major topic of discussion in standard bodies dealing with network synchronization [1]—[4] is clock noise characterization and measurement. Among the quantities considered in international standards for specification of phase and frequency stability requirements, the Maximum Time Interval Error (MTIE) has played historically a major role for characterizing time and frequency performance in digital telecommunications networks [5]—[12], as specifications in terms of MTIE are well suited to support the design of equipment buffer size.

In this paper, MTIE is first introduced according to its formal definition. Then, the main issue of its experimental measurement is pointed out: the heavy computational weight in most cases of practical interest, due to the number of operations nested in the direct, plain calculation of the MTIE standard estimator. Therefore, a fast algorithm to compute the MTIE standard estimator is described, thus making feasible MTIE evaluation based on even long sequences of Time Error (TE) samples. The computational weight of this fast algorithm is compared to that of the estimator plain calculation. Finally, the algorithm proposed is applied to TE sequences generated by simulation of all the types of power-law noise, in order to check its effectiveness and correctness.

II. Definition of MTIE

A thorough treatment of MTIE and of its properties can be found in [13]. Further specific analyses are reported in [14][15]. Here, solely the main definitions are summarized for the sake of understanding and to provide the reader with the background concepts.

A general expression describing a pseudo-periodic waveform which models the timing signal $s(t)$ at the clock output is given by [16]—[18]

$$s(t) = A \sin \Phi(t) \tag{1}$$

where A is the peak amplitude and $\Phi(t)$ is the *total instantaneous phase*, expressing the ideal linear phase increasing with t and any frequency drift or random phase fluctuation.

The generated *Time* function $\mathrm{T}(t)$ of a clock is defined, in terms of its total instantaneous phase, as

$$\mathrm{T}(t) = \frac{\Phi(t)}{2\pi \nu_{\mathrm{nom}}} \tag{2}$$

where ν_{nom} represents the oscillator nominal frequency. It is worthwhile noticing that for an ideal clock $\mathrm{T}_{\mathrm{id}}(t)=t$ holds, as expected. For a given clock, the *Time Error* function $\mathrm{TE}(t)$ (in standards also called $x(t)$) between its time $\mathrm{T}(t)$ and a reference time $\mathrm{T}_{\mathrm{ref}}(t)$ is defined as

$$x(t) \equiv \mathrm{TE}(t) = \mathrm{T}(t)\text{-}\mathrm{T}_{\mathrm{ref}}(t) \tag{3}$$

The *Maximum Time Interval Error* function $\mathrm{MTIE}(\tau,T)$ is the maximum peak-to-peak variation of TE in all the possible observation intervals τ (in former standards [5][6] denoted as S) within a measurement period T (see Fig. 1) and is defined as

$$\mathrm{MTIE}(\tau,T) = \max_{0\le t_0 \le T-\tau} \left\{ \max_{t_0 \le t \le t_0+\tau} [\mathrm{TE}(t)] - \min_{t_0 \le t \le t_0+\tau} [\mathrm{TE}(t)] \right\} \tag{4}$$

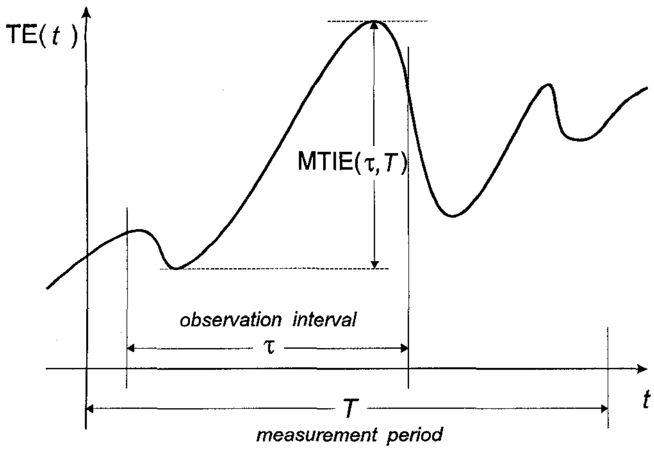

Fig. 1. Definition of MTIE(τ, T).

It should be noted, however, that the standards in force specify the MTIE limits simply as a function of τ (or S), thus implicitly assuming

$$\mathrm{MTIE}(\tau) = \lim_{T\to\infty} \mathrm{MTIE}(\tau, T) \qquad (5).$$

III. Measuring MTIE

MTIE measurement is usually based on the time-domain measurement of the TE process $x(t)$ between the output of the Clock Under Test (CUT) and a reference timing signal, which may be its input if the CUT is a slave clock (*synchronized clocks configuration*), or the output of a second Reference Clock if the CUT is a free-running clock (*independent clocks configuration*) [7][11]. Sequences of TE samples $\{x_i\}$, defined as

$$x_i = x\big(t_0 + (i-1)\tau_0\big) \quad i = 1,2,3,\ldots \qquad (6)$$

where t_0 is the initial observation time and τ_0 is the sampling period, are measured using digital counters and stored for numerical post-processing over a total measurement period T [12][13]. The samples x_i are typically measured between two corresponding zero-crossings of the timing signals involved.

Starting from the sequence $\{x_i\}$ of TE samples measured, the definition (4) may be applied directly to compute MTIE(τ,T). Letting N_T=T/τ_0+1 be the total number of available samples and N_τ=τ/τ_0+1 be the number of samples available in a window (observation interval) of span τ, for *each* single value MTIE(τ,T) the following expression has to be computed

$$\mathrm{MTIE}(\tau, T) = \max_{j=1}^{N_T - N_\tau + 1}\left[\max_{i=j}^{N_\tau + j - 1}(x_i) - \min_{i=j}^{N_\tau + j - 1}(x_i)\right] \qquad (7).$$

The above is the MTIE standard estimator recommended in [7][11].

MTIE masks currently specified in standards span over a wide range of τ: four decades, namely from 10^{-1} s up to 10^3 s. For a long time this range was even wider, from a few milliseconds up to 10^5 s. Furthermore, more specific studies may require investigation over different wide ranges.

As pointed out in [13], the number of samples N_T to process gets easily to the order of 10^5 in most cases of practical interest, if we are interested in a somehow accurate characterization of the clock noise. It obviously appears that the plain computation of the estimator (7) is unadvisable and quickly tends to be unmanageable, due to the number of operations nested in evaluation loops. Hence the need of contriving a suitable algorithm effective in cutting down the computational weight of a plain implementation of the estimator (7).

IV. MTIE Computation by Binary Decomposition

The fast algorithm proposed in this paper is based on a binary decomposition of a TE sequence $\{x_i\}$ made of N_T=$2^{k_{\mathrm{MAX}}}$ samples in nested windows made of N_τ=2^k samples (k=1, 2, 3,..., k_{MAX}). MTIE can be then evaluated recursively for each window size 2^k.

As first step (k=1), all the possible 2-points windows (τ=τ_0) are analyzed in the TE sequence: for each of them, the maximum and minimum values are stored. Their difference is the MTIE(τ_0) measured in that window and the maximum of the MTIE values of all the 2-points windows is the resulting MTIE(τ_0,T) of the whole sequence. At this first step, there is no computational saving yet compared to the plain computation of the standard estimator.

Then, as second step (k=2), all the possible 4-points windows (τ=3τ_0) are considered. The maximum and minimum values of each of these windows can be obtained by comparing the maximum and minimum values of the two 2-points windows in which the 4-points window can be split. The difference between the maximum of the two maxima and the minimum of the two minima is the MTIE(3τ_0) measured in that 4-point window. The maximum of the MTIE values of all the 4-point windows is the resulting MTIE(3τ_0,T) of the whole sequence.

The next step (k=3) is to consider all the possible 8-points windows (τ=7τ_0), split in two 4-points windows. Then so on for increasing integer values of k. The computational saving of this algorithm, compared to the plain computation of the standard estimator, lies in avoiding the comparison of all the samples in the windows of size larger then 2. The price to pay is that we have to limit the evaluation of MTIE(τ,T) just to the $\log_2 N_T$ values corresponding to the windows made of N_τ=2^k samples (this corresponds to a bit more than three MTIE values per decade on the τ axis, which may be considered sufficient in most practical applications).

More formally, starting from the TE sequence vector **x** made of N_T=$2^{k_{\mathrm{MAX}}}$ TE samples x_i, two matrices $\mathbf{A}_{\mathrm{M}}$ and $\mathbf{A}_{\mathrm{m}}$ are built. Matrices are made of N_T-1 columns (indexed by i) and $\log_2 N_T$ rows, indexed by k. The first N_T-2^k+1 elements of each k-th row of the matrix $\mathbf{A}_\mathrm{M}$ contain the maximum values of all the possible 2^k-points windows sliding from left to right along the TE sequence $\{x_i\}$. The matrix $\mathbf{A}_\mathrm{m}$ contains, in an analogous fashion, the corresponding minimum values of the 2^k-points windows. Therefore, the set of all the possible 2^k-points windows in the whole TE sequence is completely described by the couple of vectors

$$\begin{aligned}\mathbf{a}_{\mathrm{M}/k} &= \{a_{\mathrm{M}/k,i}\}\\ \mathbf{a}_{\mathrm{m}/k} &= \{a_{\mathrm{m}/k,i}\}\end{aligned} \qquad i = 1,2,\ldots,N_T - 2^k + 1 \tag{8}$$

where $\mathbf{a}_{\mathrm{M}/k}$ and $\mathbf{a}_{\mathrm{m}/k}$ are the k-th rows taken from the matrices $\mathbf{A}_\mathrm{M}$ and $\mathbf{A}_\mathrm{m}$ respectively.

The first row (k=1) of matrices $\mathbf{A}_\mathrm{M}$ and $\mathbf{A}_\mathrm{m}$ is obtained directly by the TE sequence vector $\mathbf{x}$ as

$$\begin{aligned}a_{\mathrm{M}/1,i} &= \max(x_i, x_{i+1})\\ a_{\mathrm{m}/1,i} &= \min(x_i, x_{i+1})\end{aligned} \tag{9}$$

for i=1, 2,..., N_T-1. Next rows (k>1), instead, are obtained recursively as

$$\begin{aligned}a_{\mathrm{M}/k,i} &= \max(a_{k-1,i}, a_{k-1,i+p})\\ a_{\mathrm{m}/k,i} &= \min(a_{k-1,i}, a_{k-1,i+p})\end{aligned} \tag{10}$$

where $p=2^{k-1}$, for i=1, 2,..., N_T-2^k+1.

Finally, the value MTIE(τ,T) for $\tau=(N_\tau-1)\tau_0$ and $N_\tau=2^k$ (here denoted as MTIE$_k$ for the sake of brevity) can be evaluated from the k-th rows of the matrices $\mathbf{A}_\mathrm{M}$ and $\mathbf{A}_\mathrm{m}$ as

$$\mathrm{MTIE}_k = \max_{i=1,\ldots,N_T-2^k+1} \left(a_{\mathrm{M}/k,i} - a_{\mathrm{m}/k,i}\right) \tag{11}.$$

An example of binary decomposition tree, applied on a TE sequence $\{x_i\}$ made of N_T=16 samples (k_MAX=4), is shown in Fig. 2, which depicts the four couples of vectors $\mathbf{a}_{\mathrm{M}/k}$ and $\mathbf{a}_{\mathrm{m}/k}$ (for k=1, 2, 3, 4) built recursively starting from the TE vector $\mathbf{x}$.

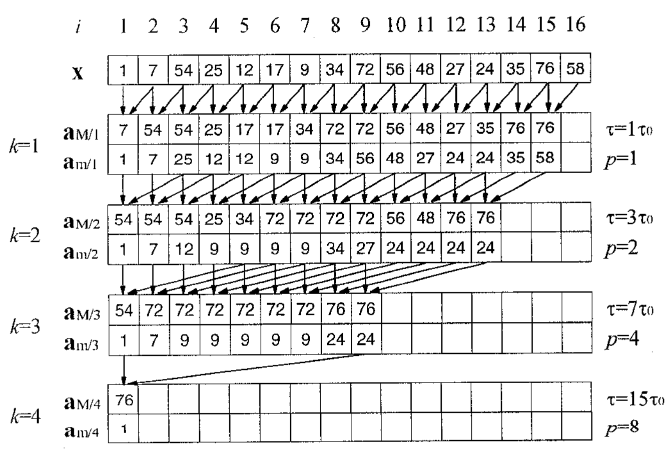

Fig. 2. Example of execution of the binary decomposition algorithm (N_T=16).

V. Computational Saving

The number of operations involved in the estimator plain computation and in the binary decomposition algorithm has been evaluated, in order to assess the resulting computational saving.

A. Plain Computation of the Estimator

As far as a plain computation of the estimator (7) is concerned, three nested loops can be identified:

1) an external loop increasing the observation interval τ, executed one time per each single value MTIE(τ,T) to compute;
2) a first internal loop executed, given τ, for each N_τ-points sliding window (the external max[·] function in (7)), i.e. $N_T-N_\tau+1$ times;
3) the most internal loop to find the maximum and minimum value in a set of N_τ samples, thus involving $2(N_\tau-1)$ comparison test branches and a variable number of assignments according to the particular TE sequence (we neglect here the possibility to use a more efficient algorithm to extract the maximum and minimum values).

If we limit MTIE computation to one value per octave on the τ axis, as in the binary decomposition algorithm, then the first loop is executed $k_\mathrm{MAX}=\log_2 N_T$ times, the second loop N_T-2^k+1 times (k=1, 2,..., k_MAX) and the third loop involves $2^{k+1}-2$ branches. Thus, the computational weight results approximately (from now on, N_T will be denoted simply as N for the sake of brevity):

$$\begin{array}{ll}\frac{4}{3}N^2 + \ldots & \text{comparison test branches}\\ 3(N\log_2 N - 2N) + \ldots & \text{assignments (best case)}\\ \frac{2}{3}N^2 + \ldots & \text{assignments (worst case)}\\ N\log_2 N - 2N + \ldots & \text{additions}\end{array} \tag{12}.$$

Comparison test branches are the most time-consuming operations.

It is worthwhile noticing that MTIE plain computation turned out to be a N^2-problem because we decided to limit MTIE computation to one value per octave on the τ axis. If MTIE is computed for all the possible N-1 values of τ, then the number of operations required gets proportional to N^3 instead.

B. Binary Decomposition Algorithm

As far as the binary decomposition algorithm is concerned, on the other hand, the following loops can be identified:

1) a first loop initializing the first row (k=1) of matrices $\mathbf{A}_\mathrm{M}$ and $\mathbf{A}_\mathrm{m}$ and then computing MTIE$_1$, involving among the rest $2(N-1)$ comparison test branches;
2) a second main loop increasing the row index k (k>1), executed $\log_2 N$-1 times;

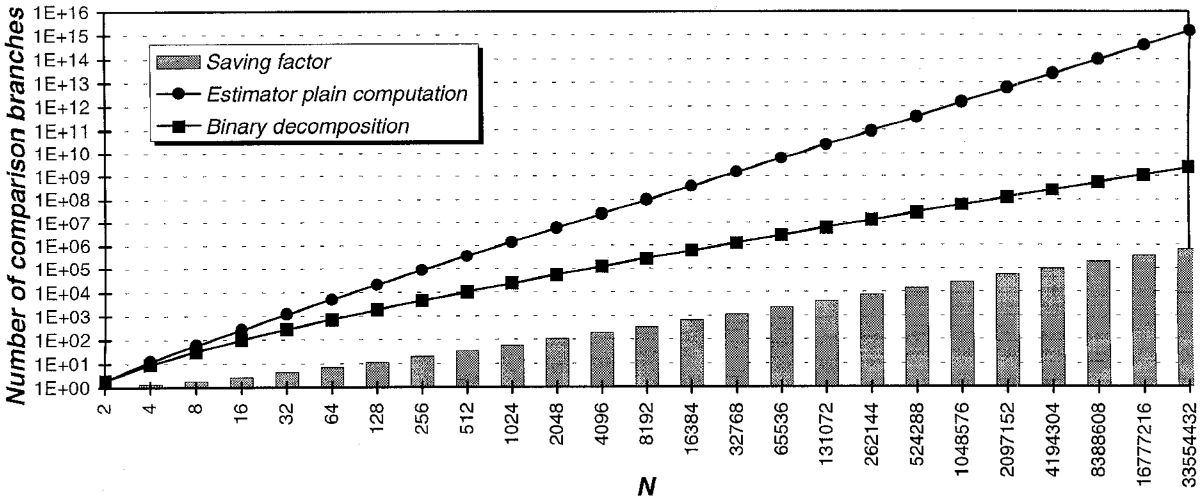

Fig. 3. Number of comparison test branches required by the MTIE estimator plain computation and the binary decomposition algorithm as a function of the total number of TE samples N.

3) a loop, internal to the previous one, to compute the next rows (k>1) of matrices $\mathbf{A}_M$ and $\mathbf{A}_m$ and to evaluate the corresponding MTIE$_k$, involving among the rest $3(N-2^k+1)$ comparison test branches.

Thus, the computational weight results approximately:

$$\begin{array}{ll} 3N\log_2 N - 7N + \ldots & \text{comparison test branches} \\ 2N + \ldots & \text{assignments (best case)} \\ 3(N\log_2 N - 2N) + \ldots & \text{assignments (worst case)} \\ N\log_2 N - 2N + \ldots & \text{additions} \end{array} \qquad (13).$$

C. *Comparison in Terms of Computational Weight*

It is worthwhile noticing that, in the binary decomposition algorithm, the number of comparison test branches and worst-case assignments needed has been reduced to a term proportional to $N\log_2 N$ instead of the N^2 involved in the plain computation of the estimator (7).

The graph of Fig. 3 compares, on a logarithmic scale, the number of comparison test branches needed by the two algorithms considered as a function of the total number of available TE samples N, for $2^1 \le N \le 2^{25}$ (to build this graph, all the lower-order terms not shown in (12) and (13) have been taken into account). Moreover, the ratio between the two numbers (i.e., the computational saving factor) is plotted as well for ease of comparison. It may be noticed that in the most common range $2^{14} \le N \le 2^{19}$ (i.e., $16.384 \le N \le 524.288$) the saving factor turns out to be in the remarkable order of $10^3 \div 10^4$.

VI. Example of Algorithm Execution on Power-Law Noise Time Error Sequences

In order to check the effectiveness and correctness of the proposed algorithm, both the estimator plain computation and the binary decomposition algorithm have been applied to TE sequences generated by simulation of the so-called *power-law* noise [17][18], the model most frequently used to represent clock output phase noise in the frequency domain. In terms of the one-sided Power Spectral Density (PSD) of $x(t)$, such model is expressed by

$$S_x(f) = \begin{cases} \dfrac{1}{(2\pi)^2} \displaystyle\sum_{\alpha=-4}^{0} h_{\alpha+2} f^{\alpha} & 0 \le f \le f_h \\ 0 & f > f_h \end{cases} \qquad (14)$$

where the h_{-2}, h_{-1}, h_0, h_{+1} and h_{+2} coefficients are device-dependent parameters[1] and f_h is an upper cut-off frequency, mainly depending on low-pass filtering in the oscillator and in its output buffer amplifier. The noise types of the model (14) are: White Phase Modulation (WPM) for α=0, Flicker Phase Modulation (FPM) for α=-1, White Frequency Modulation (WFM) for α=-2, Flicker Frequency Modulation (FFM) for α=-3 and Random Walk Frequency Modulation (RWFM) for α=-4.

First, in order to simulate WPM (α=0) noise, two white and uniformly distributed pseudo-random sequences of length $N=2^{17}=131072$ were generated. Then, applying a well-known transformation formula [19][20], one white Gaussian pseudo-random sequence of the same length was obtained, thus approximating a WPM noise. Spectral shaping was accomplished by filtering in the Fourier domain the WPM (α=0) noise sequence through integrators of fractional order $-\alpha/2$ [21], having transfer function $H_{-\alpha/2}(f)=K(j2\pi f)^{\alpha/2}$, to generate the FPM ($\alpha$=-1), WFM ($\alpha$=-2), FFM ($\alpha$=-3) and

1 The reason of the subscript α+2 (α=-4,-3,-2,-1,0) is that, historically, the coefficients h_{-2}, h_{-1}, h_0, h_{+1}, h_{+2} have been used in the power-law model definition in terms of the PSD $S_y(f)$ of the random fractional frequency deviation $y(t)=dx(t)/dt$. The relationship $S_y(f)=(2\pi f)^2 S_x(f)$ holds [17].

RWFM (α=-4) noise sequences of the same length according to the power-law model (14).

The MTIE values computed with the two algorithms, starting from the five TE sequences generated as above, are plotted in Fig. 4. As expected, MTIE values computed through the estimator plain computation and the binary decomposition algorithm are the same. Therefore, actually just one curve has been plotted per each type of noise instead of two (one per algorithm).

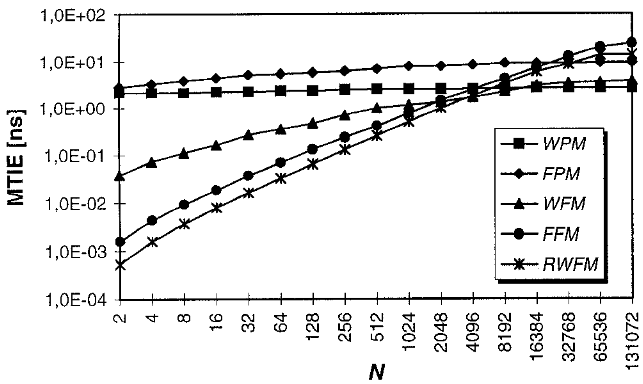

Fig. 4. MTIE values computed through the estimator plain computation and the binary decomposition algorithm on power-law noise simulated TE sequences (N=2^{17}=131072).

VII. Conclusions

In this paper, a fast algorithm based on binary decomposition to calculate the MTIE standard estimator was proposed. The computational weight of the binary decomposition algorithm was compared to that of the estimator plain computation. Moreover, the algorithm proposed was applied to TE sequences generated by simulation of power-law noise, in order to check its effectiveness and correctness.

The proposed algorithm proved effective in achieving a strong computational saving, by reducing the number of comparison test branches and worst-case assignments needed to a term proportional to $N\log_2 N$ instead of N^2 (see the graph in Fig. 3). Therefore, this binary decomposition algorithm makes feasible MTIE evaluation based on even long sequences of TE samples and may be successfully applied by telecommunications engineers involved in time-domain measurement of clock stability.

Acknowledgements

The Authors are indebted with G. Zampetti (with Telecom Solutions, S. José, CA, USA) who has been lavish of helpful suggestions on MTIE computation. Moreover, the Authors wish to thank Prof. Elio Bava (Politecnico di Milano) for his support.

References

[1] W. C. Lindsey, F. Ghazvinian, W. C. Hagmann and K. Dessouky, "Network synchronization", *Proceedings of the IEEE*, vol. 73, no. 10, Oct. 1985, pp. 1445-1467.

[2] P. Kartaschoff, "Synchronization in digital communications networks", *Proceedings of the IEEE*, vol. 79, no. 7, July 1991, pp. 1019-1028.

[3] J. C. Bellamy, "Digital network synchronization", *IEEE Communications Magazine*, vol. 33, no. 4, Apr. 1995, pp. 70-83.

[4] S. Bregni, "A historical perspective on network synchronization", *IEEE Communications Magazine*, vol. 36, no. 6, June 1998.

[5] ITU-T Recs. G.810 *Considerations on Timing and Synchronization Issues*, G.811 *Timing Requirements at the Outputs of Primary Reference Clocks Suitable for Plesiochronous Operation of International Digital Links*, G.812 *Timing Requirements at the Outputs of Slave Clocks Suitable for Plesiochronous Operation of International Digital Links*, Blue Book, 1988.

[6] ANSI T1.101-1994 *Synchronization Interface Standard*.

[7] ITU-T Rec. G.810 *Definitions and Terminology for Synchronisation Networks*. Geneva: Aug. 1996.

[8] ITU-T Rec. G.811 *Timing Characteristics of Primary Reference Clocks*. Geneva: Sep. 1997.

[9] ITU-T Rec. G.812 *Timing Requirements of Slave Clocks Suitable for Use as Node Clocks in Synchronization Networks*. Geneva: June 1998.

[10] ITU-T Rec. G.813 *Timing Characteristics of SDH Equipment Slave Clocks (SEC)*. Geneva: Aug. 1996.

[11] ETSI Draft ETS 300 462 *Transmission and Multiplexing (TM); Generic Requirements for Synchronisation Networks*. 1997.

[12] S. Bregni, "Clock stability characterization and measurement in telecommunications", *IEEE Trans Instrum. Meas.*, vol. 46, no. 6, Dec. 1997.

[13] S. Bregni, "Measurement of Maximum Time Interval Error for telecommunications clock stability characterization", *IEEE Trans. Instrum. Meas.*, vol. 45, no. 5, Oct. 1996.

[14] S. Bregni and P. Tavella, "Estimation of the percentile Maximum Time Interval Error of Gaussian white phase noise", *Proceedings of IEEE ICC '97*, Montréal, Québec, Canada, June 1997.

[15] S. Bregni and F. Setti, "Impact of the anti-aliasing pre-filtering on the measurement of Maximum Time Interval Error", *Proceedings of IEEE GLOBECOM '97*, Phoenix, AZ, USA, November 1997.

[16] J. A. Barnes, A. R. Chi, L. S. Cutler, D. J. Healey, D. B. Leeson, T. E. McGunigal, J. A. Mullen Jr., W. L. Smith, R. L. Sydnor, R. F. C. Vessot and G. M. R. Winkler, "Characterization of frequency stability", *IEEE Trans. Instrum. Meas.*, vol. 20, no. 2, May 1971.

[17] J. Rutman and F. L. Walls, "Characterization of frequency stability in precision frequency sources", *Proceedings of the IEEE*, vol. 79, no. 7, pp. 952-960, July 1991.

[18] M. Carbonelli, D. De Seta and D. Perucchini, "Characterization of timing signals and clocks", *European Trans. on Telecommunications*, vol. 7, no. 1, Jan.-Feb. 1996.

[19] S. Bregni, L. Valtriani and F. Veghini, "Simulation of clock noise and AU-4 pointer action in SDH equipment", *Proceedings of IEEE GLOBECOM '95*, Singapore, Nov. 1995.

[20] D. E. Knuth, *The Art of Computer Programming*, vol. 2, p. 118. London: Addison-Wesley, 1981.

[21] J. A. Barnes, D. W. Allan, "A statistical model of flicker noise", *Proceedings of the IEEE,* vol. 54, no. 2, pp. 176-178, Feb. 1966.