Modeling Technique and a Simulation Tool for Analysis of Clock Synchronizaion in Communication Networks

M. A. Bueva and L. N. Schelovanov
Department of Digital Data Processing and Transmission
State University of Telecommunications
Moika 61, 191065, St. Petersburg, Russia
bueva@spb.lanck.net

G. S. Antonova
Research and Development Department
GE Consumer & Industrial
8525 Baxter Pl., Burnaby, BC V5A 4V7, Canada
galina.antonova@ge.com

A. Ciuffoletti
Department of Computer Science
University of Pisa
Largo B. Pontecorvo I56100, Pisa, Italy
augusto@di.unipi.it

Abstract— **Clock synchronization and timing continue to be an active area of research and development. This paper describes a modeling technique and a simulation tool for analysis of clock synchronization in communication networks. The tool supports analysis of stability, quality of control, time to converge, calculations of standard Maximum Time Interval Error (MTIE) and Time Deviation (TDEV) parameters, etc. Initially developed for clock synchronization in communication networks, the tool can be enhanced to address synchronization needs of other applications as well.**

I. Introduction

Timing and synchronization are important aspects in the design of many major networks, such as communication networks, computer and data networks. Emerging real-time applications in various utilities and in the military require accurate time-of-day information. These include power utilities, industrial automation, military, and test and measurement applications.

In communication networks with interconnections of digital switches and transmission lines oscillator frequencies at transmitter and receiver ends must be synchronized with sufficient accuracy to avoid slips, manifesting themselves in loss or repetition of user data. Proper synchronization is required to ensure that acceptable quality of network services is provided to the end users. In addition new real-time services, e.g. electronic client billing, require maintaining accurate time-of-day information through out the whole of the communication network.

According to international recommendations clock synchronization in communication networks should be organized in hierarchical levels with the master oscillator at the top of the tree topology [1, 2]. The preferred synchronization method is the "master-slave", although the mutual synchronization and plesiochronous mode are also applicable. The master oscillator, called the primary reference clock (PRC), is a very accurate frequency source with a long-term stability better than 10^{-11}. Normally atomic clocks provide the PRC. These may be autonomous or traceable to Universal Time Coordinated (UTC), using the Global Positioning System (GPS), the Global Navigation Satellite System (GLONASS) or the Galileo system. Most slave clocks contain a crystal oscillator as part of a phase locked loop (PLL) and can fall to a holdover mode when input synchronization signals are absent.

Basic synchronization signals are 2048/1544kHz signals; TI/E1 and SONET/SDH line rate signals. For synchronization of PDH equipment through a SONET/SDH network 2048/1544kHz signals are used, since tributary TI/E1 signals are affected by SONET/SDH pointer adjustments and are not recommended for purposes of clock synchronization.

Phase noises of master and slave oscillators, different conversions of information signals, SONET/SDH pointer adjustments, diurnal temperature variations and other random components cause a whole range of phase instabilities known as jitter and wander. The quality and number of master clocks, slave clocks, synchronization links and the amount of jitter and wander, are the main factors defining the quality of clock synchronization and the quality of services the information network offers.

To estimate phase instabilities existing within a network several special criteria were proposed by standards organizations. These include Maximum Time Interval Error (MTIE), Time Deviation (TDEV) etc with requirements specified in the form of standard limitation masks [1, 2].

Computer and data communication network applications also require maintaining accurate time-of-day information. For networks built using TCP/IP protocol suite Network Time

A. Ciuffoletti has been supported by INFN/CNAF - Bologna in the frame of the European CoreGRID project.

Protocol (NTP) or Simple Network Time Protocol (SNTP) can be used to perform synchronization and timing functions. Distributed NTP servers with different timing accuracy are organized in a hierarchical structure for the Internet and intranets. Timing information is distributed from servers to clients via specialized message exchange and stored by clients in a well-known format [3].

Global trends towards real-time services, improvements in voice over IP technologies, integration of traditional voice and data networks and the increasing popularity of high speed Ethernet networks in Local and Metropolitan areas bring new challenges to network design and in particular to the organization of clock synchronization. In compliance with ongoing work led by the Metropolitan Ethernet Forum (MEF) communication network infrastructure, traditionally built on legacy SONET/SDH equipment can now be replaced by Ethernet switches at much lower cost and still ensure delivery of carrier-class services required by the end user [15].

In response to emerging market requirements for clock synchronization with sub-microsecond timing accuracy, initially primarily driven by new test and measurement applications, a new protocol for synchronization called the Precision Time Protocol (PTP) was developed in 2002 [4]. In addition to specialized message exchange between master and slave clocks PTP specifies a hardware time stamping mechanism to bypass delays in protocol stacks and correct for non-deterministic Ethernet delays. PTP Version 2, with expanded functionality, is currently being developed with strong support from communication, industrial automation, power, military, and test and measurement industries. Development of the NTP Version 4, with additional features of security, etc., is also on its way.

Changes in the market place and standardization activities require that timing and synchronization continue to be a current and active area of research and development, which raises a need for tools and instruments to assist in the analysis of such systems. Modeling and simulation have been traditionally used to analyze processes in complex systems. Although time transfer networks were described mathematically [5] and selective analysis was performed using simulations, e.g. noise accumulation in chains on slave clocks [6, 7], a comprehensive tool with a wide spectrum of capabilities specifically aimed to analyze synchronization in communication networks would be very useful. Such tool is introduced in this paper, accompanied by a description of the underlying modeling technique, graphical user interface and selected simulation results. The tool supports analysis of stability, quality of control, time to converge, calculations of standard Maximum Time Interval Error (MTIE) and Time Deviation (TDEV) parameters, etc. Currently the tool offers an analysis of synchronization in telecommunication networks built using PDH and SONET/SDH technologies. Tool enhancements are required to support TCP/IP and Ethernet-based networks.

II. Modeling Technique

A general modeling technique described in this paper allows derivation of a discrete model of a simulated system from its transfer function. The technique is applicable to both linear and non-linear systems with any input stimulus. System complexity is limited only by the processing power of the computer.

The transformation from the continuous-time *p*-domain to the discrete-time *z*-domain is performed using the Boxer-Thaler method, which was shown to be more accurate than the commonly used bilinear method. For this transformation the **S**-Matrix method was used. General expressions for a system of any order and pre-calculated matrices for up to 32nd order were obtained to facilitate the simulation of complex systems.

A. General Algorithm and Model

A general algorithm used for modeling is as follows. For any *k*th-order system with a given continuous transfer function

$$W(p)=\frac{a_0+a_1\cdot p+a_2\cdot p^2+...+a_k\cdot p^k}{b_o+b_1\cdot p+b_2\cdot p^2+...+b_k\cdot p^k} \qquad (1)$$

a general discrete model can be derived in the form of a recursive equation:

$$Y[nT_c]=\frac{1}{B_k}\left\{\sum_{\nu=0}^{k}A_{k-\gamma}\cdot X[nT_c-\nu T_c]-\sum_{\nu=1}^{k}B_{k-\gamma}\cdot Y[nT_c-\nu T_c]\right\}, \qquad (2)$$

where $Y[nT_c]$ is the output variable,

$X[nT_c]$ is the input variable of arbitrary kind,

A_i, B_i are the elements of $\overline{\mathbf{A}}_\mathbf{k}, \overline{\mathbf{B}}_\mathbf{k}$ vectors,

k is the order of the transfer function,

T_c is the sampling period, and

n is the number of samples (n = 0, 1, 2…).

Coefficients A_i, B_i are determined as

$$\overline{\mathbf{A}}_\mathbf{k}=\mathbf{S}_\mathbf{k}\cdot\overline{\mathbf{T}}_\mathbf{k}\cdot\overline{\mathbf{a}}_\mathbf{k}, \qquad \overline{\mathbf{B}}_\mathbf{k}=\mathbf{S}_\mathbf{k}\cdot\overline{\mathbf{T}}_\mathbf{k}\cdot\overline{\mathbf{b}}_\mathbf{k}, \qquad (3)$$

where $\overline{\mathbf{A}}_\mathbf{k}, \overline{\mathbf{B}}_\mathbf{k}$ are the vectors of numerator and denominator coefficients in the general discrete model,

$\overline{\mathbf{a}}_\mathbf{k}, \overline{\mathbf{b}}_\mathbf{k}$ are the vectors of numerator and denominator coefficients of the transfer function,

$\mathbf{S}_\mathbf{k}$ is the pre-calculated *k*th-order matrix for a transformation from the *p*-domain to the *z*-domain, and

$\overline{\mathbf{T}}_\mathbf{k}$ is the *k*th-order vector of expressions for sampling period.

B. Transformation from the p-domain to the z- domain

To compute processes in discrete-time systems different transformation methods from the *p*-domain to the *z*-domain can be used. Among these methods, the bilinear or Tustin transformation [8], Madwed and Boxer-Thaler transformation [9, 10, 11], all offer different degrees of accuracy and computational complexity.

General expressions for the Boxer-Thaler transformation are shown in Table 1. Expressions for the bilinear transformation for higher powers of k are found simply by multiplying the expression for p^{-1} by itself k times.

The difference between these two transformations is obvious from the general expressions. It has been previously shown [9, 10] that better results can be obtained using the Boxer-Thaler transformation due to its more accurate nature.

C. The S-Matrix Method

Even for simple systems many complex calculations are required to obtain a discrete equivalent of continuous transfer functions using any transformation method. To facilitate this task the so-called **Q**-Matrix for the bilinear and Boxer-Thaler transformations were formulated and are widely used [10, 11].

Independently and unaware of that work on the other side of the "iron curtain" in place at that time, a similar **S**-Matrix method was proposed for the Boxer-Thaler transformation with a general expression formulated for a system of any order and **S** matrices pre-calculated up to order 32 [12]. After its formulation the **S**-Matrix method was extensively used to calculate processes in discrete-time systems and in simulations.

Q-Matrix expressions correspond to **S**-Matrix expressions, which increases the creditability of the correctness of their formulation. The modeling technique described in this paper uses the **S**-Matrix method.

III. Model of Clock Synchronization

Clock synchronization can be described as an automatic control system, composed of master and slave clocks, connected via synchronization links. Master clocks are generally good quality reference oscillators, while slave clocks normally contain phase locked loops, traceable to a higher accuracy clock. Many sources of jitter and wander are present both in the clocks themselves and in links between them.

A general block diagram of a clock synchronization model consisting of n elements is shown in Fig. 1, where

$\Delta\theta$ is the input synchronization signal,

$\Delta\varphi$ is the output synchronization signal,

ξ are the frequency and phase instabilities,

W are the clock transfer functions, and

M is the synchronization link transfer function.

The synchronization signals and frequency instabilities are modeled in terms of the phase of their signals.

To obtain a complete model of a clock synchronization system, models of a slave clock, a master clock and a synchronization link were developed.

TABLE I. General Expressions for the Boxer-Thaler Method

p^{-k}	$F_k\left(z^{-1}\right)$	$F_k\left(V^{-1}\right)$
p^{-1}	$\frac{T_c}{2}\left(\frac{1+z^{-1}}{1-z^{-1}}\right)$	$\frac{T_c}{2}V^{-1}$, $V^{-1}=\frac{1+z^{-1}}{1-z^{-1}}$
p^{-2}	$\left(\frac{T_c}{2}\right)^2\left(\frac{1/3+z^{-1}\cdot 10/3+z^{-2}/3}{\left(1-z^{-1}\right)^2}\right)$	$\left(\frac{T_c}{2}\right)^2\left(V^{-2}-2/3\right)$
p^{-3}	$\left(\frac{T_c}{2}\right)^3\left(\frac{4z^{-1}+4z^{-2}}{\left(1-z^{-1}\right)^3}\right)$	$\left(\frac{T_c}{2}\right)^3\left(V^{-3}-V^{-1}\right)$

Connectivity is represented by a **C** matrix of connections incorporated into an adder block at the input of each clock:

$$C=\begin{bmatrix} 0 & c_{12} & c_{13} & \dots & c_{1n} \\ c_{21} & 0 & c_{23} & \dots & c_{2n} \\ c_{31} & c_{32} & 0 & \dots & c_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & c_{n3} & \dots & 0 \end{bmatrix}, \tag{4}$$

where c_{ij} represents the connection and gives the weighting coefficient for a synchronization link from clock i to clock j.

If slave clock x uses a single synchronization input from clock y then coefficient c_{yx} would be equal to 1 and all other coefficients in the row x would be equal to 0. If the synchronization link from clock y fails and clock x can synchronize from clock z then coefficient c_{yx} will be cleared and coefficient c_{zx} will be set to 1.

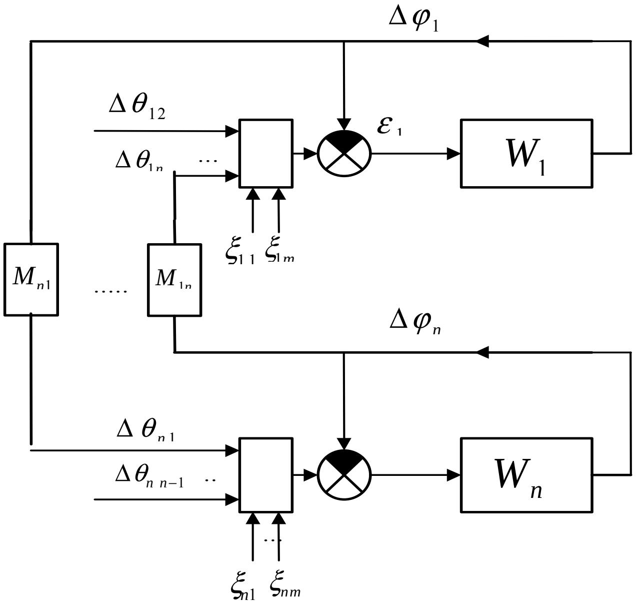


Figure 1. General block diagram of clock synchronization system with n elements.

For clocks with multiple synchronization inputs **C** matrix elements can be used as weighting coefficients that reflect the quality of input synchronization signal and quality of synchronization link from a particular source.

Master clocks with the highest accuracy may have no synchronization inputs; so all coefficients c_{ij} in its row will be equal to 0. The same is true for a slave clock when it falls into holdover mode in response to losing all synchronization inputs.

A. Slave Clock Model

A slave clock normally contains a phase locked loop, traceable to a higher accuracy clock. The general block diagram, Fig. 2, shows the main PLL components: phase detector (PD), phase detector filter (PDF), low-pass filter (LPF) and voltage-controlled oscillator (VCO).

Applying the general modeling technique described in Section II the model of a slave clock was derived as follows.

The first equation in the slave clock model is the equation of a closed loop system, characterizing phase error $\varepsilon\ [nT_c]$ as follows:

$$\varepsilon\ [nT_c] = \Delta\Theta\,[nT_c] - \Delta\varphi\,[nT_c], \tag{5}$$

where $\Delta\Theta\ [nT_c]$ and $\Delta\varphi\ [nT_c]$ are phases of input and output signals in radians.

One of the key questions in the implementation of a PLL is the choice of a phase detector; this defines the relationship between the phase difference and the control voltage. Different PD characteristics were modeled, including those using pulse amplitude modulation (PAM) and pulse duration modulation (PDM).

The simplest PAM characteristics are the ideal linear characteristic $V[nT_c] = K_{PD} \cdot \varepsilon\ [nT_c]$ with no time limitations and an arbitrary constant in volts/radians K_{PD}, sine and cosine characteristics. For the PDM method PD characteristics based on zero-crossing (ZC) and flip-flop (FF) techniques were simulated.

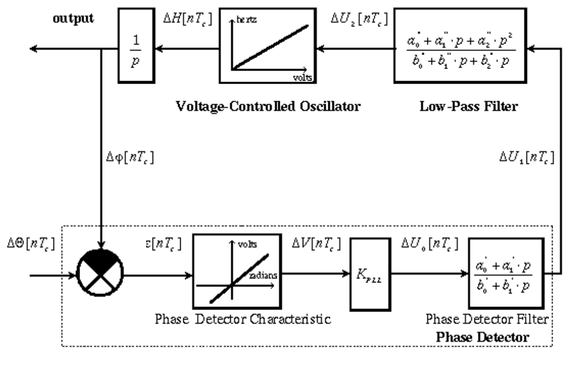


Figure 2. General block diagram of a PLL.

As shown in Fig. 2, the control signal then passes through a phase detector filter. For a first-order PDF the recursive equation (2) can be written as:

$$\Delta U_1[nT_c] = \frac{1}{B_1^{'}} \cdot \{A_1^{'} \cdot \Delta U_0[nT_c] + A_0^{'} \cdot \Delta U_O[nT_c - T_c] - B_0^{'} \cdot \Delta U_1[nT_c - T_c]\}, \tag{6}$$

where $\Delta U_0[nT_c],\ \Delta U_1[nT_c]$ are the input and output, respectively, at time nT_c, and

$\Delta U_0[nT_c - T_c],\ \Delta U_1[nT_c - T_c]$ are the input and output at time $nT_c - T_c$.

Coefficients $A_0^{'}, A_1^{'}, B_0^{'}, B_1^{'}$ are determined from the coefficients of a PDF transfer function using the first-order matrix S_1.

After leaving the PD the control signal goes through the low-pass filter. For a second-order LPF the recursive equation (2) can be written as:

$$\begin{gathered}\Delta U_2[nT_c] = \frac{1}{B_2^{''}} \cdot \{A_0^{''} \cdot \Delta U_1[nT_c] + A_1^{''} \cdot \Delta U_1[nT_c - T_c] + \\ A_2^{''} \cdot \Delta U_1[nT_c - 2T_c] - B_0^{''} \cdot \Delta U_2[nT_c - T_c] - B_1^{''} \cdot \Delta U_2[nT_c - 2T_c]\}.\end{gathered} \tag{7}$$

Coefficients $A_0^{''}, A_1^{''}, A_2^{''}, B_0^{''}, B_1^{''} B_2^{''}$ are calculated from the coefficients of a LPF transfer function using the second order matrix S_2.

After passing through the low-pass filter, the control signal is fed to the voltage-controlled oscillator to tune its frequency. The relationship between the control voltage and the oscillator frequency is defined by the oscillator characteristic. For a linear oscillator characteristic with a correlation coefficient K_{osc}:

$$\Delta\omega = K_{osc}\Delta u_2,$$

and in terms of phase

$$\frac{d\Delta\varphi(t)}{dt} = K_{osc} \cdot \Delta u_2, \tag{8}$$

where $\Delta\omega$ is the oscillator frequency variation,

$\Delta\varphi(t)$ is the oscillator phase variation, and

Δu_2 is the control voltage variation.

In the *p*-domain (8) can be written as:

$$K_{osc} \cdot \Delta U_2(p) = p \cdot \Delta\Psi(p). \tag{9}$$

Using (9) the VCO model can be derived as

$$\Delta\varphi[nT_c] = \frac{1}{B_1'''}\left\{A_1''' \cdot \Delta U_2[nT_c] + A_0''' \cdot \Delta U_2[nT_c - T_c] - B_1''' \cdot \Delta\varphi[nT_c - T_c]\right\}. \quad (10)$$

Coefficients $A_0''', A_1''', B_0''', B_1'''$ are determined using the matrix S_1.

Various sources of PLL noises, such as PD noise, VCO noise, quantization noise can optionally be added to the slave clock model. A good description of noise sources in PLLs is given in [13].

B. Master Clock Model

A master clock model includes its frequency stability, long-term and short-term accuracy. Stability is modeled as a systematic frequency variation with given maximum offset $\Delta f_{\max}$ and accuracy is modeled as additive white Gaussian noise with a given maximum variation a_{ac}.

The master clock model for the period of simulation in days T_{sim} is follows:

$$o_1[nT_c] = o_{st}[nT_c] + o_{ac}[nT_c] = o_{st}[nT_c - T_c] + \frac{T_{sim}}{T_c} \cdot 2\pi \cdot (r_{st}[nT_c] + r_{ac}[nT_c])$$

where $r_{st}[nT_c]$ is the uniformly distributed variable with values in the interval $[0, \quad \Delta f_{\max}]$, and

$r_{ac}[nT_c]$ is the uniformly distributed variable with values in the interval $[-a_{ac}/2, \quad a_{ac}/2]$.

C. Synchronization Link Model

A synchronization link model that takes into account the propagation delay τ is obtained from the transfer function of a delay block:

$$W(p) = \frac{Y(p)}{X(p)} = e^{-p\tau}.$$

Using (2) the model can be written as follows:

$$\Delta\varphi^*[nT_c] = \left(1 + \frac{2\tau}{T_c}\right) \cdot \Delta\varphi[nT_c] + \left(1 - \frac{2\tau}{T_c}\right) \cdot \Delta\varphi[nT_c - T_c] - \Delta\varphi^*[nT_c - T_c] \cdot$$

Coefficients $A_0'''', A_1'''', B_0'''', B_1''''$ are determined using the matrix S_1.

Simulations assume a constant propagation delay for a given synchronization link. Model enhancements are required to include variable and asymmetrical delays, which are present in real-life networks.

Phase jumps, frequency offsets, jitter and diurnal wander, which are typical network impairments, are optional additions to the synchronization link model.

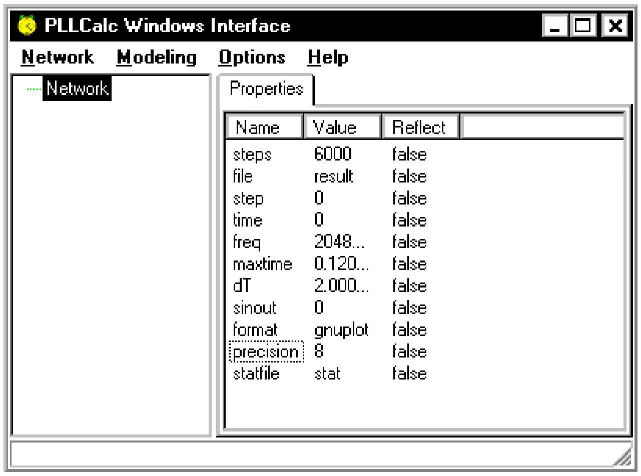
IV. Simulation Tool

Using the described modeling technique a simulation tool was developed by students at the St. Petersburg State University of Telecommunications, Russia. The tool is targeted for use in a MS Windows environment with the computational part written in the standard ANSI C++ programming language and graphical user interface (GUI) written in XML language.

Input parameters for the tool include, but are not limited to, the number and type of clocks, their connectivity, parameters of slave clocks and sources of jitter and wander. A choice of various input stimulus is offered, with phase jump and frequency jump being among the possible choices. Observation intervals for the MTIE and the TDEV calculations are also user configurable.

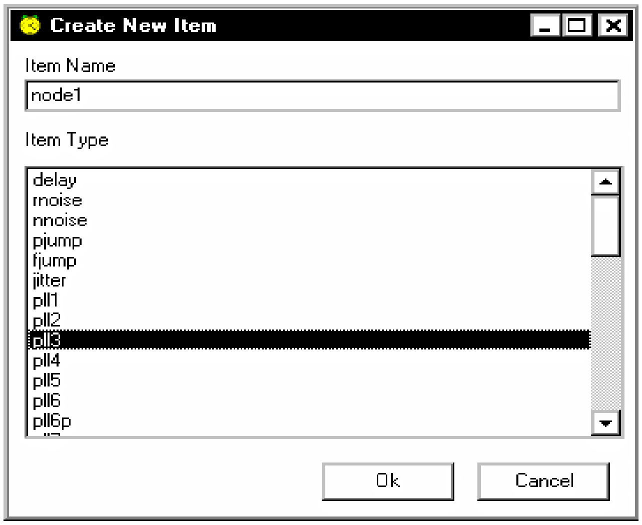
From the main program window, Fig. 3, a user can create a clock synchronization system by selecting clocks from a list of predefined models or by creating his own system from a list of clock components as shown in Fig. 4. Clock parameters, including a type of phase detector, noise sources and input stimulus, are configured in the node's property menu, Fig. 5. Once all nodes are created and configured, they can be connected using links menu, as shown in Fig. 6.

This tool has been widely used since it was created for analysis of clock synchronization elements and complete network synchronization systems. Due to its modular structure and scalability any network topologies can be analyzed with various clocks and links included in simulations. For example, chains of slave clocks represented by SDH equipment clocks (SECs), with or without stand-alone synchronization equipment (SASE), can be modeled. Various fault tolerance scenarios can be considered, including switching to a standby synchronization source, restoration of an original synchronization source, fall back into holdover mode upon losing all synchronization sources, etc.



Name	Value	Reflect
steps	6000	false
file	result	false
step	0	false
time	0	false
freq	2048...	false
maxtime	0.120...	false
dT	2.000...	false
sinout	0	false
format	gnuplot	false
precision	8	false
statfile	stat	false

Figure 3. The main program window.



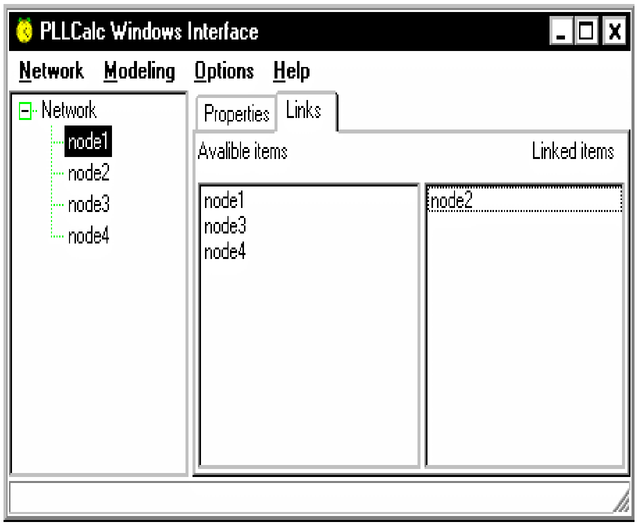
Item Name

node1

Item Type

delay
rnoise
nnoise
pjump
fjump
jitter
pll1
pll2
pll3
pll4
pll5
pll6
pll6p

Figure 4. Creating a new synchronization node.



Network
- node1
- node2
- node3
- node4

Avalible items	Linked items
node1 node3 node4	node2

Figure 6. Connecting synchronization nodes.

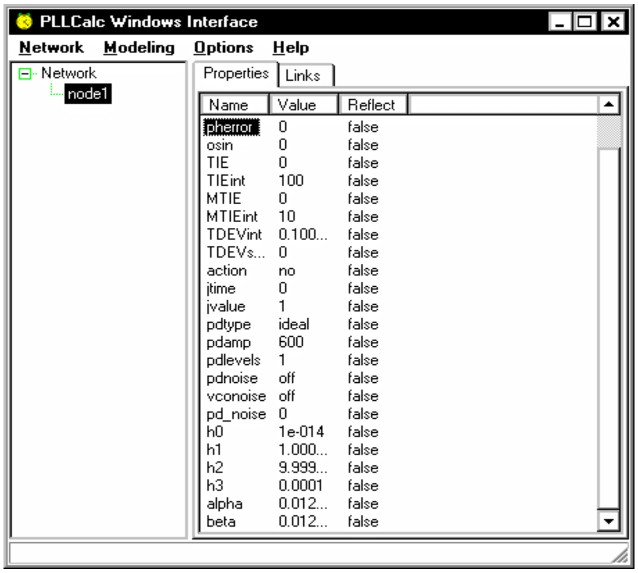
V. Selected Simulation Results

To illustrate the tool capabilities let us demonstrate selected simulation results obtained for the analysis of different synchronization systems.

System stability is generally a very important requirement enforced on automatic control systems. Depending on system and slave clock parameters the whole system may or may not return to a normal mode of operation after an unexpected disturbing stimulus is applied. Stability analysis for synchronization systems with given parameters can be performed using the so-called D-splitting method. The method allows defining margins for system and clock parameters to ensure its stable operation.

The quality of control characteristics, e.g. a response to a phase or frequency jump, can be analyzed for stand-alone slave clocks and for a complete system. Different types of slave clocks were evaluated using a phase jump as an input stimulus. Results were compared to standard masks for devices of appropriate types. On network level chains with a various number of SEC slave clocks were modeled.

Fig. 7 and Fig. 8 exhibit simulation results when a phase jump to 0.75 π radians and frequency jump of 6 radians/s were applied to the first slave clock in the chain. Both results demonstrate expected delays in propagation of the disturbing stimulus and a return to a normal stable mode of operation.



Network
- node1

Name	Value	Reflect
pherror	0	false
osin	0	false
TIE	0	false
TIEint	100	false
MTIE	0	false
MTIEint	10	false
TDEVint	0.100...	false
TDEVs...	0	false
action	no	false
jtime	0	false
jvalue	1	false
pdtype	ideal	false
pdamp	600	false
pdlevels	1	false
pdnoise	off	false
vconoise	off	false
pd_noise	0	false
h0	1e-014	false
h1	1.000...	false
h2	9.999...	false
h3	0.0001	false
alpha	0.012...	false
beta	0.012...	false

Figure 5. Configuring node's parameters.

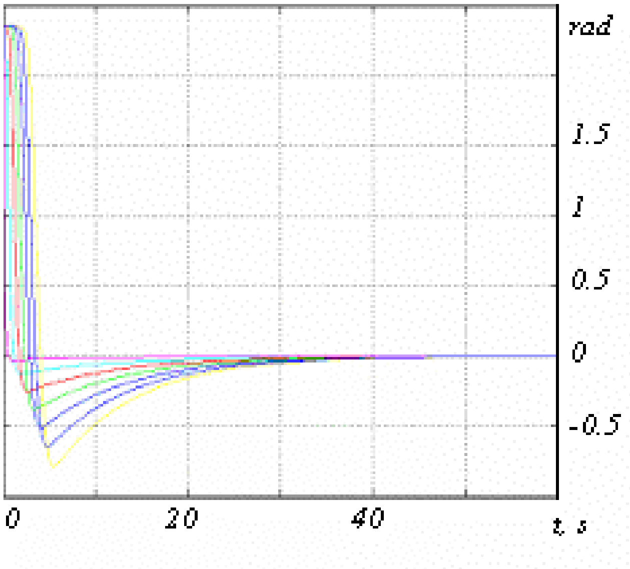


Figure 7. Phase errors at the outputs of a chain of SEC slave clocks after a phase jump is applied to the 1st clock.

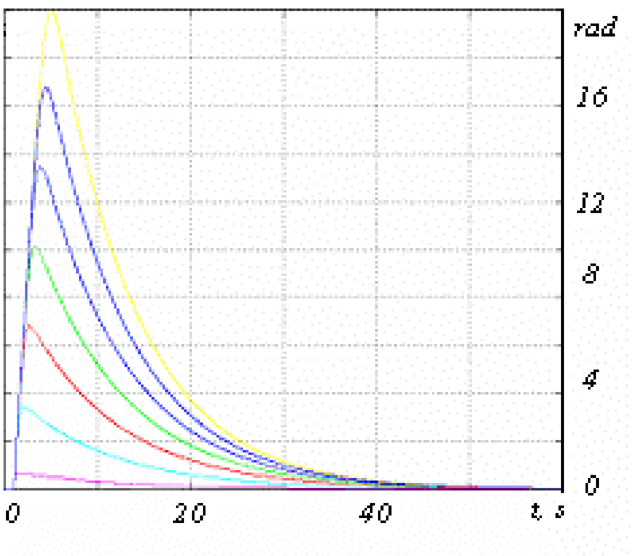


Figure 8. Phase errors at the outputs of a chain of SEC slave clocks after a frequency jump is applied to the 1st clock.

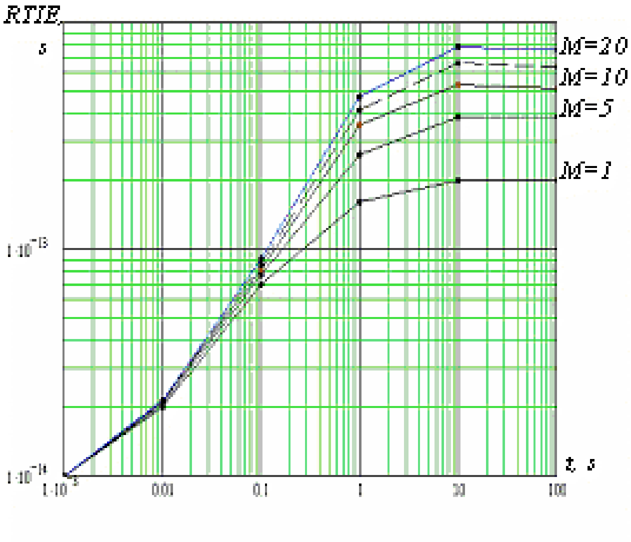


Figure 10. RTIE for chains of M SEC slave clocks, caused by noise of the phase detector.

The time to acquire a synchronous mode of operation when slave clocks are controlled by higher accuracy master clocks is an important parameter for many systems. Transient processes happen at the time of initial system startup, switching to an alternative synchronization source, restoring a preferable synchronization connection and falling in and out of a holdover mode. The time to converge was evaluated for systems with a different number of synchronization nodes. As shown in Fig. 9 the number of nodes affects the time to converge much more in systems with a smaller number of nodes.

Noise accumulation in a long chain of slave clocks has been an area of concern for many synchronization systems. To address this issue simulation were performed for chains of SEC slave clocks with and without SASE clock.

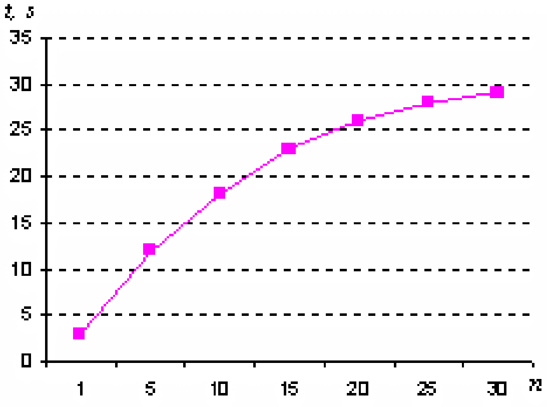


Figure 9. Time to converge as a function of the number of nodes in a synchronization system.

The relative time interval error (RTIE) was evaluated for chains with up to 20 SEC clocks. For these experiments only noise of phase detector was included into SEC slave clock model. Simulation results, depicted in Fig.10, have relatively low RTIE values, showing that the effect of phase detector noise can be insignificant.

The MTIE results were obtained for chains of SEC slave clocks with the noise of their oscillators included in the model. The plots in Fig. 11 show that adding a SASE clock into a chain can dramatically improve the noise characteristics of the whole synchronization system.

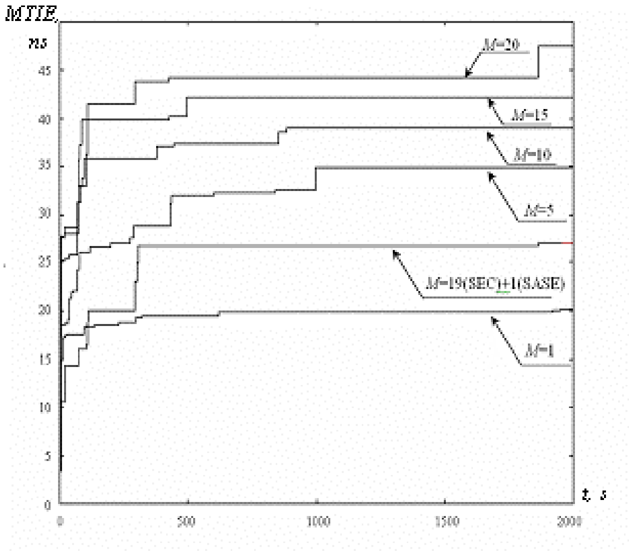


Figure 11. MTIE for chains of M SEC slave clocks with and without a SASE clock.

The developed tool was compared with the commercially available simulator MATLAB 6.5, which has a specialized library for analysis of synchronization systems. Experiments show that simulation results obtained using these programs are in agreement, with variance of less than 1%, but the time required to run the simulations is significantly less when the developed package is used. Analyzing identical systems with observation time τ = 100s, simulations using MATLAB take 2 times longer and with observation time τ = 1000s 3.5 times longer than simulations using the developed package. Fig. 12 illustrates simulation time as a function of observation interval for simulations of three linear systems, performed using the developed tool and MATLAB 6.5.

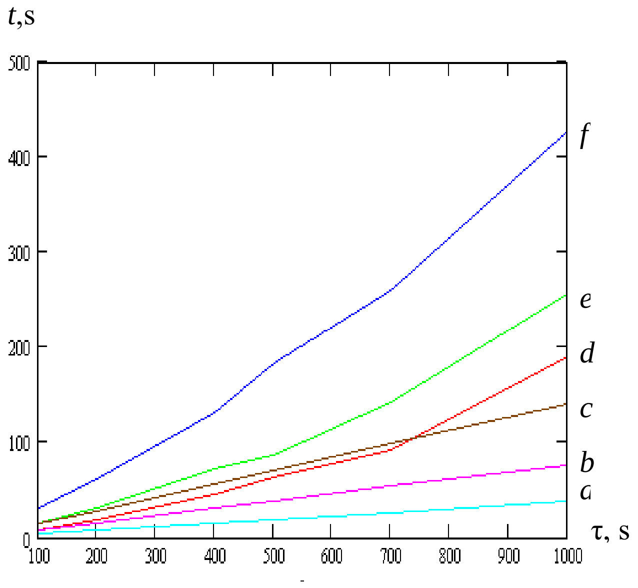


Figure 12. Simulation time *t* as a function of observation interval τ for simulation of linear synchronization systems with N nodes using the developed tool (graphs *a*,*b*,*c*) and MATLAB 6.5 (graphs *d*,*e*,*f*). N =5 for graphs *a* and *d*. N=10 for graphs *b* and *e*. N=20 for graphs *c* and *f*.

VI. Conclusion

Clock synchronization and timing continue to be a current and active area of research and development. The modeling technique and a simulation tool described in this paper were specifically developed to assist in the analysis and design of clock synchronization systems. The tool has a friendly easy-to-use graphical user interface and supports detailed analyses of synchronization system stability, characteristics of quality of control, time to converge, calculations of standard MTIE and TDEV parameters, etc. The modular structure of the underlying model allows for easy modifications, e.g. to include new clock types or to add calculations of a new quality criteria.

The tool can be enhanced by including an optional variable delay to the synchronization link model, adding support for analysis of clock synchronization in TCP/IP and Ethernet-based networks, etc. Comparison with the commercially available simulator MATLAB 6.5 has shown an agreement of simulation results, with variance of less than 1%, and that a significantly smaller simulation time is required using the tool, introduced in this paper.

References

[1] ITU-T Recommendations G.810-G.824, Geneva, Switzerland, 1997-2004.

[2] ETSI EN 300 421 Standards.

[3] D. L. Mills, “Network time protocol (Version 3) specification, implementation and analysis,” RFC 1305, March 1992.

[4] “IEEE standard for a precision clock synchronization protocol for networked measurement and control systems,” IEEE Std. 1588-2002, New York, USA, Nov. 2002.

[5] W. C. Lindsay, A. V. Kantak, “Mathematical models for time transfer networks,” in Proc. of the 1978 IEEE International Conference on Communications (ICC’78), Toronto, Canada, June 1978, pp. 16.1.1-16.1.4.

[6] G. M. Garner, “Accumulation of random noise in a chain of slave clocks,” in Proc. of the 1994 IEEE International Frequency Control Symposium, Boston, USA, June 1994, pp. 798-811.

[7] M. Carbonelli, D. De Seta, D. Perucchini, “Root mean square time interval error accumulation along slave clock chains,” in Proc. of the 1993 IEEE International Conference on Communications (ICC’93), Geneva, Switzerland, May 1993, pp. 816-820.

[8] F. D. Groutage, L. B. Volfson, and A. M. Schneider, “S-plane to Z-plane mapping using a simultaneous equation algorithm based on the bilinear transformation,” IEEE Trans. on Automatic Control, vol. AC-32, pp. 635-637, July 1987.

[9] G. S. Antonova, L. N. Schelovanov, E. M. Doronin, “Experimental proof of expediency of use of Boxer-Thaler method for digital simulation of complex linear units,” in Summaries of the 6th Technical Conference for Professors and Students, St. Petersburg State University of Telecommunications, June 1994 (in Russian).

[10] C. H. Wang, M. Y. Lin, C. C. Teng, “On the nature of the Boxer-Thaler and Madwed integrators and their applications in digitizing a continuous-time system,” IEEE Trans. on Automatic Control, vol. 35, pp. 1163-1167, Oct. 1990.

[11] M. N. Jackson, G. Hayward, “A Q-matrix implementation of the Boxer and Thaler Z-Form method,” IEEE Trans. on Circuit and Systems, vol. CAS-33, pp. 771-714, July 1986.

[12] L. N. Schelovanov, “Algorithmic method to calculate processes in unlocked and locked video systems with arbitrary inputs,” Questions of Radio Electronics, Moscow, 1970 (in Russian).

[13] V. F. Kroupa, “Noise properties of PLL systems,” IEEE Trans. on Communications, vol. COM-30, pp. 2244-2252, Oct. 1982.

[14] R. Boxer and S. Thaler, “A simplified method of solving linear and nonlinear systems,” in Proc. of the IRE, vol. pp. 89-101, Jan. 1956.

[15] Metropolitan Ethernet Forum documents and specifications: http://www.metroethernetforum.org/.