DOI: http://doi.org/10.25728/cs.2021.4.5

ESTIMATING TIME CHARACTERISTICS OF CONTROL SYSTEMS WITH CYCLIC OPERATION: A NETWORK CALCULUS APPROACH[1]

V.G. Promyslov[2] and K.V. Semenkov[3]

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

[2]✉ v1925@mail.ru, [3]✉ semenkovk@mail.ru

Abstract. The practical validation of time characteristics of digital control systems is considered. The delay in information processing and transmission often has a probability distribution differing from the Gaussian one. Therefore, the confidence intervals calculated under the Gaussian distribution assumption will be incorrect for such systems. The idea is to estimate the time characteristics of a control system using non-statistical time parameter estimation methods. As one of such methods, Network Calculus is considered. The practical implementation of Network Calculus to estimate the parameters of control systems, particularly its features, is discussed. One of the main features is imposing special restrictions on data flows and system performance, determined by flow envelopes and maximum (minimum) service curves. Generally, these characteristics are unknown in advance. Mathematical methods are proposed to estimate these characteristics under known input and output data flows in the system. As shown below, the calculation of characteristics is significantly simplified for systems with cyclic data processing algorithms, and the data transfer rate over the network is much higher than that on the computing components of the system. Simulations are carried out, and the system's time parameters estimated by Network Calculus are compared with the results of classical statistical estimation methods. As an illustrative example, the time parameters of one component of a real nuclear power plant instrumentation and control system are estimated using Network Calculus.

Keywords: system performance, time characteristics, digital control systems, Network Calculus, non-statistical estimation methods.

INTRODUCTION

In most cases, modern control systems (CSs) for industrial plants are implemented as a computing environment distributed by functions and means. Its components are a set of hardware and software means for acquisition, accumulation, asynchronous processing, representation, and transmission of information. CS components can be distributed both spatially and functionally.

Dynamic validation methods of system parameters have been developed to confirm the CS operability. They vary depending on the industry: for example, the IAEA methodology is used for CSs of nuclear power plants (NPPs) [1]. Network simulators based on discrete mathematics, such as OMNeT ++ [1–3] and OPNET [4, 5], are used to estimate the dynamic characteristics of CSs. However, statistical methods are necessarily applied to confirm the time characteristics of a real system. In the latter case, the samples of measurements of the system's parameters are analyzed under the common assumption that the distribution law of the measured characteristics is close to Gaussian [1]. In most cases, this assumption is true for physical signals. As shown below, it can be false for the parameters describing the digital CS itself, e.g., signal transmission and processing times.

Network Calculus [6], a non-statistical analysis method for deterministic systems, is an alternative approach to estimating the characteristics of data flows between the components of computer networks. This method is based on the min-plus algebra and is attractive: in many cases, it allows considering linear systems that are nonlinear in the conventional algebra.

[1]This work was supported in part by the Russian Foundation for Basic Research, project no. 19-29-06044 (Sections 2 and 3).

The method involves no assumptions about the type of distribution for the measured process.

A feature of Network Calculus is using specific functions – the envelopes of the input and output data flows and service curves – o calculate the system characteristics, which primarily include the data transmission delay and buffering parameters.

Network Calculus was developed for analyzing flow systems without information losses during processing (e.g., for calculating the throughput of a network segment or determining the parameters of video streaming over Ethernet networks). Generally speaking, control systems do not belong to this class of systems due to the following characteristics:

- parallel processing of several tasks on one computing resource,
- a significant change in the volume of information at the component's input and output (the output flow can be either greater or smaller than the input one, e.g., when compressing information),
- heterogeneous information in CSs, unlike information transmission systems (here, information heterogeneity means that each element (bit) has a specific value and can be processed according to a particular algorithm).

No doubt, these features were considered within Network Calculus. The authors [7] extended the method to systems with cyclic dependencies between the input and output flows of components. The papers [8, 9] presented Network Calculus approaches for systems with a significant change in the input-output flow ratio. The publications [10, 11] considered various methods for describing joint processing disciplines of several tasks on one computing resource.

The approaches mentioned above have common drawbacks. First of all, their application requires accurate knowledge of the internal features of system operation: bound to them, the approaches become sensitive to any change in the operating modes of the system. In addition, when used for complex systems, the approaches lose the "transparency" of results and their simple correlation with other characteristics (the input data rate, data unevenness, and the computing power of the component).

In view of these drawbacks, when constructing a system model below, we will try preserving the generality and transparency of the results (on the one hand) and reflecting the unevenness of input and output data flows and the dependence of the data processing algorithm on the information contained in them (on the other hand).

Also, we will validate Network Calculus by comparing the delays yielded by statistical estimation methods and the former method. This comparison is of particular interest for correlating the results of Network Calculus with those of statistical methods, which is an underinvestigated problem.

We will solve the problem on model examples of control systems and a real NPP instrumentation and control system [11]. The CS under study is of a rather general type, and the problems considered below are common for the developers of industrial CSs. Therefore, this experience may be of interest to other researchers and engineers of industrial CSs.

1. THE STRUCTURE OF A TYPICAL PROCESS CONTROL SYSTEM

1.1. Typical interaction of control system components

This paper considers a typical CS for an industrial plant, further referred to as the CS. A similar CS structure arises in various applications for real plants; see [11, 13].

The typical structure of the CS is shown in Fig. 1. There are three levels:

– programmable logic controllers (PLC) and gateways (level *G* or level 1),

– the servers of primary processing and data storage (level *S* or level 2), and

– the components providing the human-machine interface (level *Z* or level 3).

The CS architecture under consideration has the following properties:

– One or more components of levels 1 and 3 can be connected to the server.

– Each communication channel (C1–C4) between different-level components can be redundant (redundancy is not shown in the diagram).

– Components *G*, *S*, and *Z* have cyclic information processing algorithms.

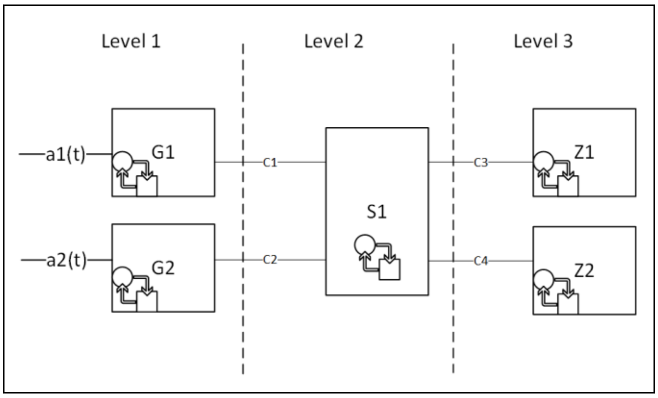

Fig. 1. The structural diagram of control system.

1.2. Data processing features

According to the operating conditions of the CS (see Fig. 1), different-level elements of the system use cyclic processing algorithms for the data transmitted from the gateway (*G*) at the level of connection to the server (*S*) and from the server to the workstation (*Z*). The data are the signals representing the plant's state and the control system itself.

Definition 1. A system element implements a cyclic data processing algorithm if the algorithm has the following properties:

– The element's initial state is waiting for the data arrival.

– The sequentially incoming data packets are processed in a deterministic uniform way, after which the system returns to the initial state.

For a cyclic algorithm, the total processing time of a data packet can be written as the sum of two values:

$$D_C = T_E + T_S ,$$

where T_E denotes the network delay, and T_S is the processing time on the CS element.

2. NETWORK CALCULUS FOR ESTIMATING TIME CHARACTERISTICS OF CONTROL SYSTEMS

2.1. Foundations of Network Calculus

Network Calculus [6] is based on rather new methods of applied mathematics introduced by Cruz [14, 15]. They involve the min-plus algebra; see the monograph [16]. The main application area of Network Calculus is the studies of queuing systems.

Let us briefly describe the method, following the book [6].

Definition 2. A flow (also called a cumulative flow) is a nondecreasing function of time such that

$$\begin{cases} A(t) \le A(s), \forall t < s, \\ A(t) \in \mathbb{R}_+ \cup \{+\infty\}, \\ t \in \mathbb{R}. \end{cases}$$

A flow is said to be causal if $A(t) = 0,\ \forall t \le 0$. The set of all causal functions is often denoted by *F*. For such functions, we introduce the operations of convolution and deconvolution.

Definition 1. Let *A* and β be causal flows. Their min-plus convolution, denoted by $A^* = A \otimes \beta$, has the form

$$A^*(t) = \inf_{0 \le s \le t} \{\beta(t-s) + A(s)\}. \quad (1)$$

Whenever no confusion occurs, we will omit the variable t in the expressions below. Obviously, $A^*(t) = 0,\ \forall t < 0$, and A^* is nonnegative since both A and β are nonnegative causal functions.

Definition 4. Consider functions A and β, where β is causal. Their mini-plus deconvolution, denoted by $H = A \oslash \beta$, has the form

$$H = \sup_{u \ge 0} \{A(t+u) - \beta(u)\}. \quad (2)$$

Note that the deconvolution of flows A and β, where β is causal, is a flow as well.

Definition 5. Let *A* and β be causal functions. Their max-plus convolution, denoted by $A^* = A \bar{\otimes} \beta$, has the form

$$A^*(t) = \sup_{0 \le s \le t} \{\beta(t-s) + A(s)\}. \quad (3)$$

Definition 6. Let A and β be flows, where β is causal. Their maxi-plus deconvolution, denoted by $H = A \bar{\oslash} \beta$, has the form

$$H = \inf_{u \ge 0} \{A(t+u) - \beta(u)\}. \quad (4)$$

Definition 7. A function β is the (minimum) service curve of a network element (or a system) with an input flow A if β is a causal flow and the output flow A^* of the element (system) satisfies the relation

$$A^* \ge A \otimes \beta. \quad (5)$$

Definition 8. A function γ is the (maximum) service curve of a network element (or a system) with an input flow A if γ is a causal flow and the output flow A^* of the element (system) satisfies the relation

$$A^* \le A \otimes \gamma. \quad (6)$$

Definition 9. A function a is called an envelope of a flow A if $A \le A \otimes a$ or, equivalently,

$$a \ge A \oslash A. \quad (7)$$

Incoming and outgoing flows are determined by the total volume of data observed at the input and output over a certain period. Therefore, the data pass through the system in a time defined as the horizontal deviation between these functions, $d(t)$.

Definition 10 (maximum delay in system). For linear systems with an input flow A, an output flow A^*, $A(t) \ge A^*(t)$, the maximum delay D_{max} is the maximum horizontal distance between the input and output flows:

$$D_{\max} = h(A, A^*) = \sup_{t \ge 0} \left\{ \inf \{ d \ge 0 : A(t) \le A^*(t+d) \} \right\}.$$

A fundamental result of Network Calculus is the possibility of determining delays using flow envelopes and service curves instead of cumulative flows:

$$D_{\max} = h(a, \beta). \quad (8)$$

It was proved in the book [6].

3. APPROACHES TO ESTIMATING CHARACTERISTIC CURVES OF NETWORK CALCULUS

Consider the problem of determining the flow envelope, the system's maximum and minimum service curves, and their linear approximations based on experimentally measured flows.

3.1. Calculating flow envelope based on experimental data

Equation (7) gives a direct way to calculate the envelope of a cumulative flow *A*. For facilitating calculations, it is convenient to operate the piecewise-linear approximation of the envelope, reduced in some cases to the affine function $y = kx + b$. The piecewise-linear representation allows adopting efficient computational algorithms for data processing. Operating the affine function, we can quickly ("on the fly") analyze the system and assess its behavior in quantitative terms.

Piecewise-linear approximation is traditionally used in the analysis of complex systems. Due to simplicity, this model is indispensable for Network Calculus, where both specially developed algorithms [10] and mathematical methods of optimal control and system identification are applied; for example, see [17] or [18].

The approximation of flow envelopes by the affine curve within Network Calculus was considered in [19]. The methods for calculating a one-component linear flow envelope suggested therein were based on support vector algorithms [20].

3.2. Calculating maximum and minimum service curves based on experimental data

Formula (7) allows directly calculating the flow envelope: the problem reduces to searching for effective analytical and computer methods of linear approximation. Determining the parameters of the service curve is much more difficult.

Theoretically, an exact estimate of the service curve can be obtained from formulas (1) and (2) using a specially selected test flow and the fact that the zero element δ_0 of the convolution functions is absorbed by the operator $\otimes$; see the book [6], p. 111. However, such an experiment is impracticable: it requires generating an infinitely large flow, which exceeds the capabilities of any real system.

The second approach is based on the property of the min-plus algebra described in [6]:

$$C \geq B \oslash A \iff B \leq A \otimes C. \tag{9}$$

Using this relation and formula (2), we obtain a lower bound for the maximum service curve:

$$\gamma' \leq A^* \oslash A, \tag{10}$$

where A and A^* are the input and output cumulative flows, respectively.

But the maximum service curve is often not enough to analyze the system. For example, the minimum service curve (5) is required to calculate the maximum system delay and the maximum buffer size. Algorithms for calculating the minimum service curve of general systems are unknown to the authors.

The approach to calculating the minimum service curve proposed below involves the following "weaker" property of the min-plus convolution and deconvolution.

Property. *If* $C \leq B \bar{\oslash} A$, *then*

$$B \geq A \otimes C. \tag{11}$$

P r o o f.

Let $C(s) \leq (B \bar{\oslash} A)(s)$ for $s \in \mathbb{R}$. This means that for any $v \geq 0$,

$$B(s+v) - A(v) \geq \inf_{u \geq 0} \big(B(s+u) - A(u)\big) \geq C(s),$$

i.e.,

$$B(s+v) \geq C(s) + A(v). \tag{12}$$

Introducing the notation

$$t = s + v,$$

we rewrite inequality (12) as

$$B(t) \geq A(t-s) + C(s). \tag{13}$$

Inequality (13) holds for any s such that $t \geq s \geq 0$. Hence, it will be true in the limit case (for the lower bound of its right-hand side):

$$B \geq A \otimes C, \ , \forall t \geq 0.$$

The proof of this property is complete. ♦

Now we estimate the minimum service curve. Let A and A^* be input and output cumulative flows, respectively. Due to the property established above, the function

$$\beta' = A^* \bar{\oslash} A \tag{14}$$

satisfies the inequality $A^* \geq A \otimes \beta'$. In other words, the estimate β' is the minimum service curve.

Since the property (11) is only a necessary condition, the estimate of the minimum service curve given by (14) can lie above or below the real minimum service curve of the system. Comparing the expressions (9) and (11), we also note that $\beta \leq \gamma'$. In other words, the minimum service curve is bounded above by the maximum service curve.

Consider a special case when the system has no maximum service curve: there is an "instantaneous" processing mode for input flows. In this case, an exact value of the minimum service curve can be obtained by replacing the input and output cumulative flows with their envelopes. To do this, assume that α and α^* are the envelopes of the input and output flows, respectively. As is known, $\alpha^* = (\alpha \otimes \gamma) \oslash \beta$; see the book [6], p. 34. For $\gamma(t) = \delta_0$, this equation can be written as

$$\alpha^* = \alpha \oslash \beta .$$

Due to the property of the operator $\oslash$ (see [6], p. 123) and the minimum service curve β,

$$\alpha = \beta \otimes \alpha^* .$$

Using the commutativity of the operator $\otimes$ and the same property reversely, we arrive at the following estimate of the minimum service curve:

$$\beta' = \alpha^* \oslash \alpha .$$

If the service curves can be described by affine functions, then there exist fast convolution and deconvolution algorithms necessary for calculating the system's parameters [11]. As shown in [21], the service curves can be approximated by affine functions like the flow envelope using similar support vector algorithms.

4. MODELING AND ESTIMATING TIME CHARACTERISTICS OF CONTROL SYSTEMS

Let us describe the typical CS (Fig. 1) using the Network Calculus model. In addition, assume that the CS has redundant computing power. Under this assumption, the system can be decomposed, and each logical channel can be considered separately. Otherwise, it is necessary to examine the mutual influence of different processing channels, e.g., using a task scheduler model [11].

In the CS shown in Fig. 1, consider a separate control channel (Fig. 2). Each CS component in the channel model is characterized by its maximum and minimum service curves. According to the definitions of the minimum (5) and maximum (6) service curves, the derivation and final equations for the maximum service curve of the CS will be similar to those for the minimum service one. They can be written by simply renaming the variables and reverting the inequality signs in the relations. Therefore, all the main conclusions and considerations in this section will concern the minimum service curve β. To indicate a particular component, we will add an appropriate alphanumeric subscript to β according to the notation in Fig. 2. Level 1 of the system receives an input flow designated by an uppercase letter with an index. The input and output flows of each component will be denoted by A and A^*, respectively.

Due to the definition of the minimum service curve (5), for each element of the linear system, we have the expression

$$A^* \geq A \otimes \beta .$$

However, in practice, the characteristics of all CS elements (except communication channels) are nonlinear: the scale of the flow between the input and output changes. For example, one alarm signal at the component's input can cause a whole avalanche of related signals in the algorithms of information protection and display in the control system. As a result, the information at the component's output will increase. To incorporate the flow scale changes into the model shown in Fig. 2, we introduce a scaling function M and its inverse M^{-1}. They implement the transformations $M : A^* \to A$ and $M^{-1} : A \to A^*$, respectively, [8]. In this case, the service curve β_{Si} of system channel i with the scaling functions takes the form

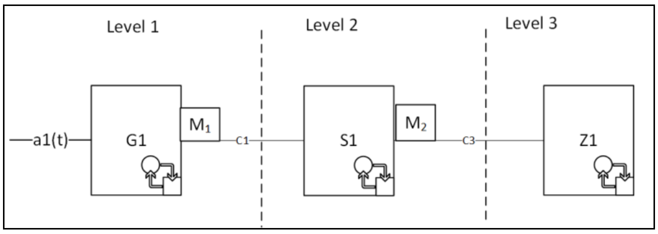

Fig. 2. Logical transmission channel *i*1 separated in CS.

$$\beta_{Si} = \beta_{Gk} \otimes M_1^{-1}(\beta_{Cn1} \otimes \beta_{Sl} \otimes M_2^{-1}(\beta_{Cn2} \otimes \beta_{Zm})), \quad (25)$$

where: $i, k, l, m \in \mathrm{N}$ are the numbers of serially connected components in the logical data processing channel at each CS level; $n1, n1 \in \mathrm{N}$ are the numbers of the communication channels used for data transmission between the components in channel $i1$; finally, M_1 and M_2 are the scaling functions of the corresponding components. The service curves β_{Cn1} and β_{Cn2} reflect the network data transmission delay T_E; the others, the data processing delay T_S in the component.

If the service curves and scaling functions can be calculated for each component, equation (15) yields bounds on the data processing delay for the entire system, depending on the input flow characteristics $ai(t)$, $i \in \mathrm{N}$. However, calculating the scaling functions M of a real system is a difficult problem not necessarily solved in practice.

To avoid difficulties with determining the scaling functions, we apply the following technique for the systems with a cyclic data processing algorithm: redefine the input and output flows and pass from the real flows to the virtual ones.

Suppose that all data received at the beginning of each cycle will be processed and transmitted to the output by the end of the cycle. Consider the function

$$\begin{cases} q(j) = \tau_j, \, j \in N, \\ q(0) = 0, \end{cases}$$

where j denotes the cycle number, and τ_j is the duration of cycle j. On the interval $[0, +\infty)$, we introduce the function

$$Q(x)=\sum_{l=0}^{j} q(l),\ j\le x<j+1.$$

Obviously, the step function $Q(x)$ is a flow by Definition 2.

For such a component, the output flow Q^* can be obtained from the input flow by the single-cycle shift:

$$Q^*(x)=\begin{cases} Q(x-1), x\in[1,+\infty),\\ 0, x\in[0,1). \end{cases}$$

Figure 3 shows the structural diagram (Fig. 2) redefined for the virtual flows in the components of types *G*, *S*, and *Z*.

For the channel with such virtual flows for the components *G*, *S*, and *Z*, we redefine the minimum and maximum service curves β and γ:

$$Q^*\ge Q\otimes\beta,$$
$$Q^*\le Q\otimes\gamma.$$

Also, we introduce the direct mappings M_0', M_1', and M_2' for the transformation $M': Q^*\to A$ and the inverse mappings for the transformation $M^{-1'}: A^*\to Q$. Then the service curve for the system in Fig. 3 takes the form

$$\beta_{Si}=\beta_{Gk}\otimes M_1'\otimes\beta_{Cn1}\otimes M_1^{-1'}\otimes \otimes\beta_{Sl}\otimes M_2'\beta_{Cn2}\otimes M_2^{-1'}\otimes\beta_{Zm}. \quad (16)$$

In turn, equation (16) can be reduced to a convenient form; see subsection 5.1 of the paper [8]. For this purpose, the scaling functions M' are transferred from the component's input to the output, and the pair ($M', M^{-1'}$) at the component's output is canceled:

$$\beta_{Si}=\beta_{Gk}\otimes M_1^{-1'}(\beta_{Cn1})\otimes\beta_{Sl}\otimes \otimes M_2^{-1'}(\beta_{Cn2})\otimes\beta_{Zm}. \quad (17)$$

The partial transition from the data flows A to the cycles Q in equations (16) and (17) does not simplify the operation of scaling functions. However, under the condition

$$\beta_{Ci}\gg\beta_{\{G,S,Z\}i}, \quad (18)$$

the curve β_{Ci} can be replaced by the function

$$\delta(t)=\begin{cases} 0, t=0,\\ +\infty, t>0. \end{cases}$$

It is neutral with respect to min-convolution and has the property $f=\delta\otimes f$. (For example, see the book [6].)

The monotonic scaling function of the network component satisfies the relation

$$M^{-1'}(\delta(t))\to\delta(n), n\in\mathrm{N}.$$

For such a function, we may ignore β_{Ci} in equation (17), thereby eliminating the scaling functions. Physically, the assumption (18) means that the processing cycle time in the network stack corresponding to the information transmission time over the system network is negligible compared to the information processing time on a computing resource. This assumption mainly holds for modern digital control systems, where the transmitted information has a relatively small volume compared to the throughput of communication channels.

In this case, the general service curve for chain *i* of the secondary flow (see equation (17)) reduces to

$$\beta_{Si}=\beta_{Gk}\otimes\beta_{Sl}\otimes\beta_{Zm}.$$

Here $k, l, m\in\mathrm{N}$ are the numbers of the serially connected components that process data at the CS levels.

Although Network Calculus is quite "transparent" in the sense of results interpretation, it involves nonstandard characteristics of the studied objects: the flow envelope and the service curve. These characteristics are not directly measured but result from calculations. Obviously, the methods used to calculate them will also affect the reliability of the final result. Therefore, we will focus on the practical aspects of calculating the flow envelope and the service curve.

5. VALIDATION OF NETWORK CALCULUS FOR ESTIMATING TIME CHARACTERISTICS

5.1. Reference data and validation procedure

Before calculating the characteristics of the typical CS (Fig. 1), we have to validate Network Calculus on data with known statistical parameters. A test program

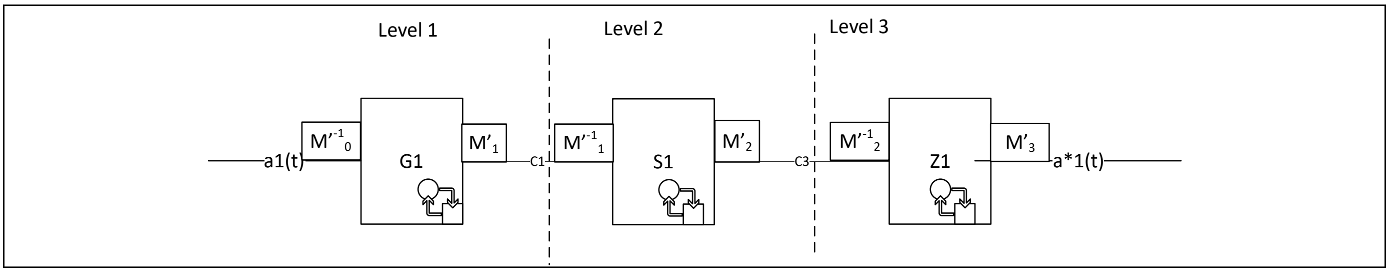

Fig. 3. The structural diagram of CS with cyclic virtual flows.

was created to simulate the CS component with a cyclic operation algorithm; the network delay *TE* and the cycle time *TS* were random variables with a given distribution. The obtained data were processed using the Network Calculus library of Matlab [10].

The subject of study was the ratio of the maximum delay determined by Network Calculus and the maximum measured delay in the sample as well as the dependence of the maximum calculated delay on the sample size and distribution.

In addition, a technique was developed to carry out measurements and obtain samples with network packet sizes and cycle times on a test program simulating the cyclic algorithm of a real system. Figure 4 shows the pseudocode of this test program. The file containing the parameters of the delay distribution function is read line by line in the loop; a random delay with a given distribution is inserted after each read operation. The duration of each cycle is recorded in the output file.

```
printTimeAndVal(InData, OutData)
 // subroutine print timestamp and input/output data counter

main
begin
   insum=0;  //  input  data flow counter
   outsum=0; ;  //  output  data flow counter

while((nch =read(par1, par2, …, parN, Distr)) >0) //read file by lines
{

  insum += nch; // increase current input  flow counter
 printTimeAndVal(insum, outsum)); //write data to file

  randn(par1, par2, …, parN, Distr);  // Distr – statistical distribution
// delay execution on random value with selected distribution
// par1, par2, …, parN - parameters of the distribution
    write(outdata);
    outsum += nch; // increase current output data   flow counter
    printTimeAndVal(fpi, insum, outsum);
}
end;
```

Fig. 4. Test program's pseudocode for generating output flows with a given delay distribution.

5.2. Testing technique and results

For each sample, three delays were calculated:

– the maximum experimental delay D_x over the entire dataset,

– the delay determined by Network Calculus according to formula (8) with the minimum service curve β estimated by formula (14),

– the delay determined by Network Calculus according to formula (8) with the lower bound (10) of the maximum service curve γ.

The calculations were carried out for samples of different sizes *L*. The data in the samples had different distributions, including those close to Gaussian and heavy-tailed ones. The numerical results presented below are rounded within 1%.

The initial test data are combined in the table available for download [22]. The data obtained with the test program are indicated by the asterisk (*).

Figure 5 shows the ratio D/D_x depending on the sample size for different distributions, where *D* is the maximum delay determined by Network Calculus with the service curve (14).

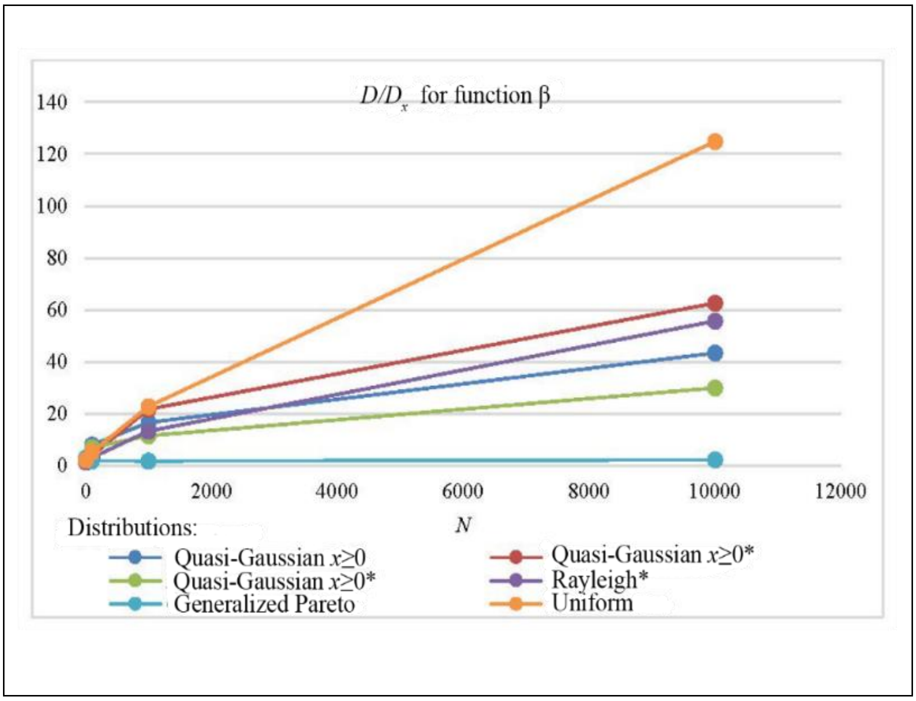

Fig. 5. Maximum delay estimate depending on sample size for β.

Figure 6 shows the ratio D/D_x depending on the sample size for different distributions, where D is the maximum delay determined by Network Calculus with the service curve (10).

The testing results allow drawing the following conclusions.

The maximum delay D estimated using the service curve (10) is close to the experimental maximum delay D_x: as a rule, the former is somewhat smaller than the latter. The resulting estimate better correlates with the real maximum delay for large samples and distributions close to Gaussian [22].

At the same time, the estimate of the maximum delay using the minimum service curve (14) is more accurate for small samples and heavy-tailed distributions. The ratio D/D_x grows with increasing the sample size, although the delay's rate of change decreases with increasing the sample size. The ratio D/D_x can reach 10^2.

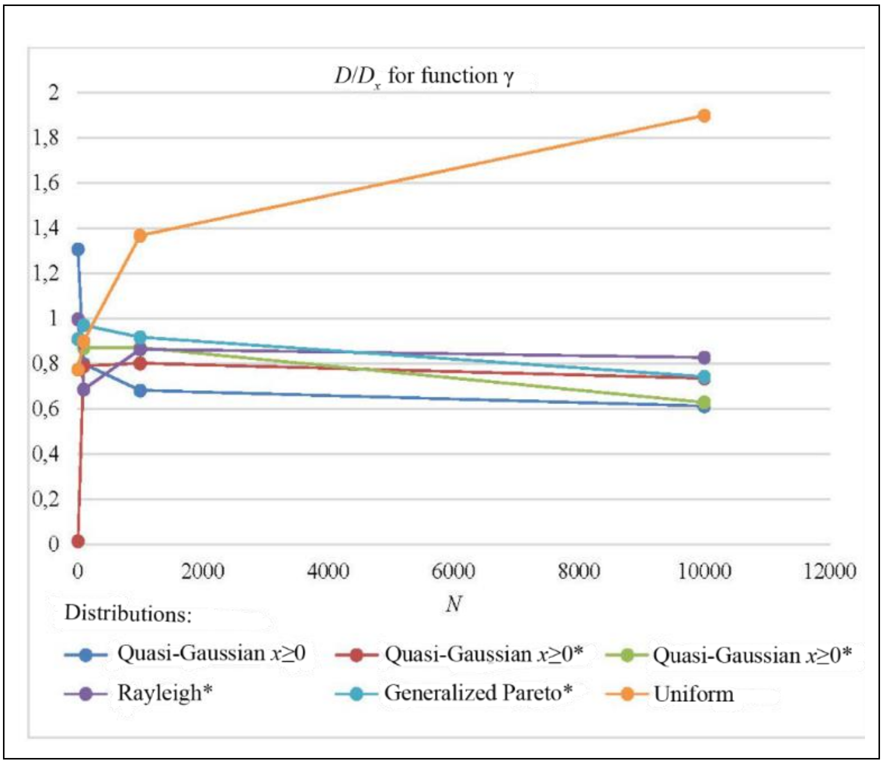

Fig. 6. Maximum delay estimate depending on sample size for γ.

Figure 7 presents the ratio D/D_x depending on the sample size and overshoot amplitude in the case of the minimum service curve.

The simulation shows that the maximum delay calculated using the service curve (10) characterizes the delay in normal operating conditions; see Fig. 6. The estimated delay is close by absolute value to the maximum delay in the sample and weakly depends on the sample size for sufficiently large samples.

Figure 8 shows typical curves yielded by Network Calculus based on the experimental data. The sample data have the Rayleigh distribution with $\sigma = 1$; however, single overshoots of 300 σ were added to the sample. For clarity, the small-sample data are given here. The upper horizontal line corresponds to the maximum delay calculated for the service curve (10). This delay is close to the maximum sample delay. The lower

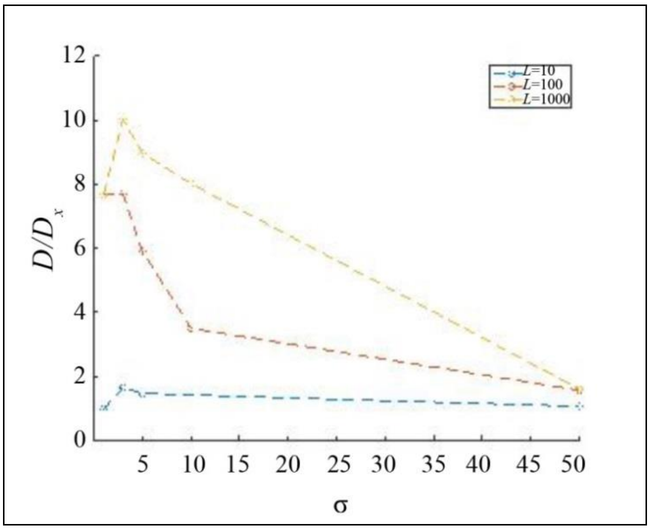

Fig. 7. The ratio of measured and calculated delays for the Rayleigh distribution with μ = 0 and σ = 300 bytes depending on single overshoot in σ and sample size *L*.

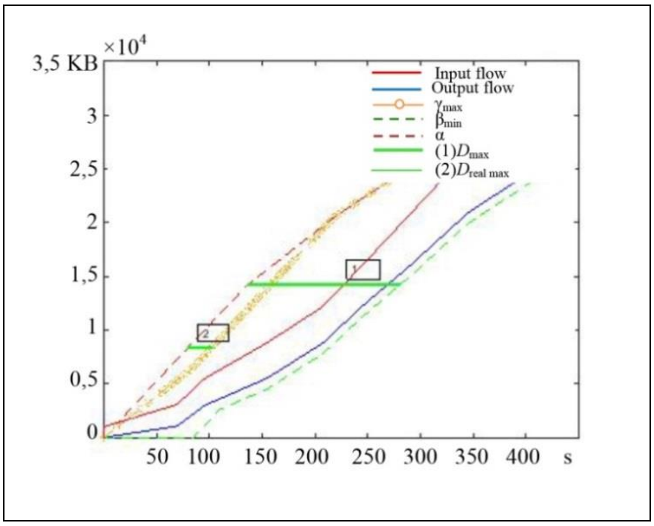

Fig. 8. Network Calculus-based experimental curves for a sample of size *L* = 10 with the Rayleigh distribution and a single overshoot of 300 σ.

horizontal line corresponds to the maximum delay estimated for the service curve (14). Clearly, the input flow envelope limits all curves on the graph from above; the estimate of the minimum service curve (14), from below.

For the samples obtained on the test example (not generated), the results turned out to be somewhat less stable. However, the differences in the order of magnitude from the same distributions obtained by direct data generation do not exceed 20%. The data acquisition procedure described above can be recommended for measurements of real systems.

5.3. Comparison of results: Network Calculus vs. classical statistical methods

According to the simulation results, the ratio D/D_x and the maximum delay depend on the distribution function of processing times, the sample size, and the number and amplitude of single overshoots in the data.

This dependence is complex due to the nonlinear formulas describing the basic Network Calculus operations (1)–(4) to find the maximum delay. According to these formulas, the flow envelope and the service curve include segments of close values arranged in descending order for the flow envelope and the maximum service curve and ascending order for the minimum service curve. (See the book [6], p. 113.)

Thus, for larger samples, the flow will contain many segments with a considerable slope of the curve; therefore, the flow envelope (7) and the service curves calculated by formulas (10) and (14) will change.

The maximum service curve (10) estimated on the sample will be similar to the flow envelope. The minimum service curve has the opposite trend (see Fig. 8). Hence, the estimate (10) depends less on variations of the input data and sample size.

Heavy-tailed distributions are characterized by some overshoots strongly differing from the rest of the values. For distributions close to Gaussian, the appearance of such overshoots in the sample is less probable, but they are characterized by a sufficient volume of data within the confidence interval. Accordingly, the general slope trend for the envelope and service curves will differ depending on the distribution of the measured parameter. The samples with single large overshoots will be characterized by the curves with a large slope value at the beginning and its subsequent sharp decrease; the samples without large overshoots, by the curves with a smooth decrease in slope (Fig. 9).

Some examples of the service curves for different sample sizes were given in the supplementary material; see Fig. 1 in [23].

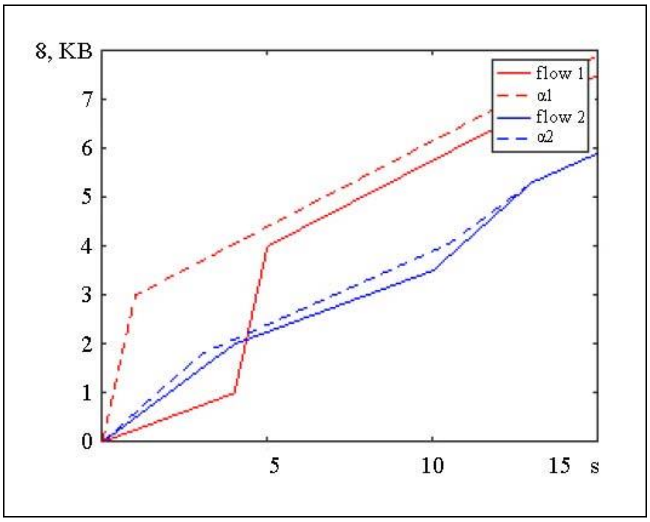

Fig. 9. Example of two cumulative flows and envelopes (α1, α2) for them. Flow 1 has overshoots in data (point 4 on axis X). Flow 2 has no overshoots in data.

These considerations explain the relationship between the delays estimated by Network Calculus and classical statistical methods. (For example, see [23].)

As is known [25], the results yielded by Network Calculus assume the worst combination of information processing conditions in the system. Graphically, this means that the segments with the greatest changes in the input flow are concentrated at the beginning of the envelope curve (the worst scenario predicted based on the observed data). For the delay based on the minimum service curve, the worst scenario is the largest data packet arriving when the server is busy and has low performance. When calculating the delay with the maximum service curve, the maximum data size corresponds to the maximum service characteristic: the maximum volume of data is accompanied by the maximum system performance, which is typical for normal operating conditions.

In both cases, the delay estimated by Network Calculus corresponds to the delay calculated using statistical methods in the scenarios described above. The probability that the delay will reach this value in a real sample corresponds to the experimental probability of this scenario. During simulations, we calculated the probability that the real delay would be less than that yielded by Network Calculus. This probability is close to 1 for the delay determined using the minimum service curve; see [22].

6. CALCULATING CONTROL SYSTEM DELAY: AN EXAMPLE

We estimated the time characteristics of the real control system described above (Fig. 1). Note that the network data transmission delays between the components were also measured to validate the simplified formula (23) for calculating the service curve of the entire system.

Empirical distributions were calculated for the measured values, and spectral characteristics were additionally analyzed for the network delay.

Consider the maximum data processing time in a component with a cyclic operation algorithm estimated by Network Calculus.

The measurements were carried out for the components of level *Z* (Fig. 2). The volume of cyclic data processed is rather stable in normal operating conditions and has an average rate of change. However, under special conditions (actuation of equipment protection and lockouts or transition between modes), the volume of data and the algorithm (rate) of data processing can vary significantly.

The empirical distribution of the cycle time T_S (Fig. 10) differs from the Gaussian or Poisson and is multimode. In accordance with the algorithm of operation, each mode corresponds to a typical processing cycle for a certain class of data.

For the given sample, the estimate of maximum delay using the service curves (10), (14) was made. The following table presents the results obtained.

Simulation results for component *Z*

$L \sim 10^3$							
D_x	$p(D_x)$	D	$p(D)$	D/D_x	D'	$p(D')$	D'/D_x
0.37	~ 1	5.1	~ 1	4.9	0.32	0.87	0.3

Consider the parameters of the network data transmission delay T_E between the system components shown in Fig. 2. As an example, the data on network packets passing between the components of levels *S* and *Z* are presented. The components in this example exchange data via the TCP/IP protocol. The data transmission characteristics between other components of the system are similar.

In the experiment, the standard tcpdump utility of the OS was used for measuring the round-trip time (RTT) of a TCP packet, i.e., the period between sending the packet by the component *S* and receiving confirmation [26] from the component Z. The RTT measurements are shown in Fig. 11.

The typical round-trip time of a packet is tens of microseconds. However, the RTT distribution significantly differs from the Gaussian distribution inherent in physical processes or the Poisson distribution widely used in queuing theory (Fig. 12).

The distribution in Fig. 12 has three distinct periods. At the same time, no long-term periods were revealed during the experiment. The RTT spectrum analysis in Fig. 13 confirms this fact: the spectrum is noisy.

The maximum transit time of network packets is approximately 10^3 times smaller than the processing time of information in cycles. Hence, for this CS, the service curve can be calculated using the simplified formula (18).

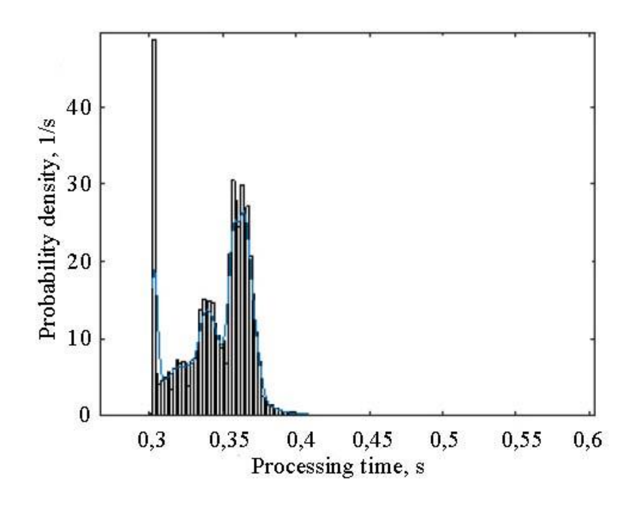

Fig. 10. The empirical probability density of cycle time for component Z. The firm line shows a smoothened envelope of the distribution.

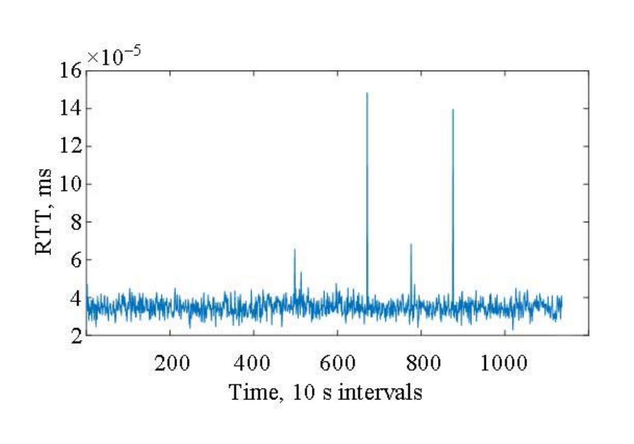

Fig. 11. RTT between the components S and Z of the real CS. The data are averaged on 10 s intervals.

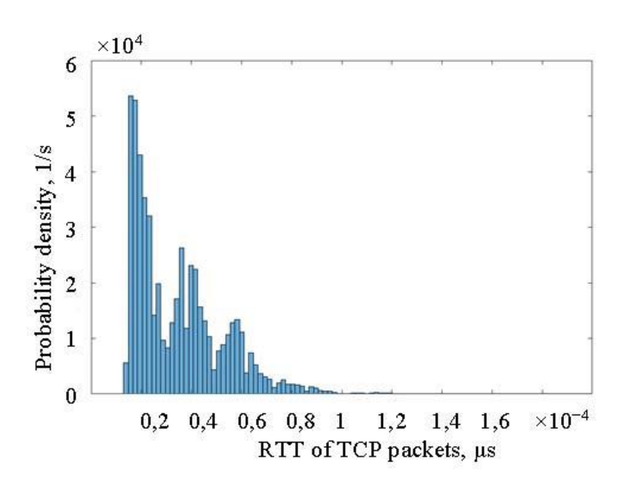

Fig. 12. The empirical probability density of RTT of TCP packets.

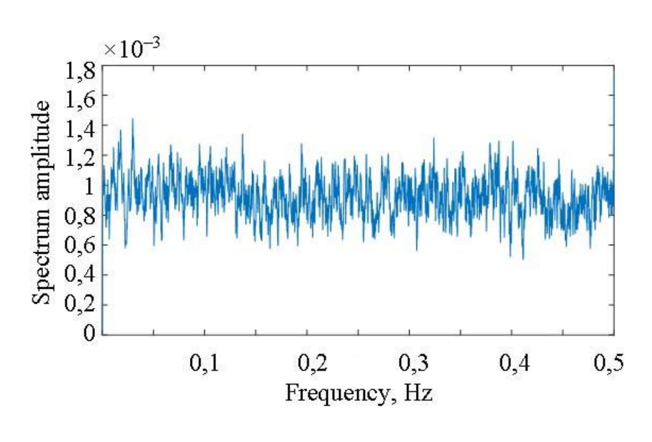

Fig. 13. The amplitude of RTT spectrum. The zero harmonic corresponding to the mean value is cut out. The data are reduced to a uniform scale with 2 s intervals between points.

In the course of measurements on the real CS, we verified that the empirical distributions (Figs. 10 and 12) have a heavy tail. The algorithm for recognizing heavy-tailed distributions [27] was employed for this purpose. It demonstrated better results than the tests based on the Kolmogorov–Smirnov criterion.

According to the real system test, the distributions of delays in the network components and the information processing components belong to the class of heavy-tailed distributions.

CONCLUSIONS. DISCUSSION OF THE RESULTS

This paper has considered the problem of validating the time characteristics of digital control systems (CSs) during their testing. CS requirements often include restrictions on the processing time of individual CS components and the information transmission time between CS components.

Constraints can be imposed on both average and maximum (limit) values of these characteristics. They are expressed either in statistical form (confidence intervals) or in standard form (admissible ranges of the absolute values) [27].

Estimating a random variable on a sample is a classical problem of mathematical statistics: it has long been developed and well described in the literature (for example, see [23]). However, the interpretation of the resulting estimates, which extends the experience of operating "ordinary" measurements of physical quantities with almost Gaussian distributions to estimating the time characteristics of digital control systems, may lead to incorrect conclusions. Let us formulate the main problems.

The procedure for validating the requirements during tests is primarily based on calculating the sample mean and sample variance (e.g., the IAEA methodology [1]). If a random variable has a finite mean and variance, the sample mean is a consistent unbiased estimate of the theoretical mean and does not depend on the type of distribution. A known disadvantage of this method is low robustness under extraneous overshoots in the sample [29]. However, sample variance, both biased and unbiased, is a consistent estimate of the variable's theoretical variance.

When interpreting the resulting estimates of the mean and variance, engineers implicitly assume that the delays obey the Gaussian law and intuitively transfer the estimated confidence intervals for the Gaussian variable to control system delays. Indeed, if a random variable has the Gaussian distribution, the sample mean and variance can be used to estimate the confidence interval for the validated parameter. However, the distribution function of delays in CSs is generally non-Gaussian.

The physical nature of the measured quantity (time) restricts the form of its distribution: at least, it is bounded on the left. If the technical requirements specify the maximum absolute value (e.g., the signal transit time between the CS components should not exceed a given threshold), this condition implies that the distribution of the random variable is also bounded on the right. Therefore, the restrictions dictate that the distribution is not, in the strict sense, the distribution of a Gaussian random variable.

According to the study of a real CS (see above), the distribution of delays in the network components and the components processing information significantly differs from the Gaussian one: it often has a multimode nature and belongs to the class of heavy-tailed distributions.

In the general case, the Chebyshev inequality yields a (very rough) estimate for the probability that a random variable will exceed a given threshold. Therefore, when estimating time characteristics, it is necessary to obtain an appropriate distribution and then apply statistical estimation methods for this class of distributions or, as an alternative, use non-statistical methods.

This paper has considered a non-statistical approach to estimating the time delay in control systems based on Network Calculus. This method is not completely new; however, it is still underinvestigated by researchers. When applying it to computer systems analysis, we should consider some of the features of the method. One feature is insufficient transparency in correlating the results yielded by Network Calculus with those yielded by classical statistical methods for estimating time parameters of control systems. In addition, the system's input data necessary for this method are generally not specified as "passport parameters" of the system and the information processed by it. Such data include flow envelopes, service curves, scaling functions for uneven data flows, etc. The technical difficulties of Network Calculus are well known, and separate approaches have been developed to resolve them partially; for example, see [8, 10, 11, 19, 30]. However, these solutions also require initial data about the system, which are unavailable for the user or are poorly formalized. Moreover, there is no general methodology for estimating the minimum service curve, an important parameter of Network Calculus.

Therefore, this paper has proposed methods for estimating the minimum service curve using the input and output cumulative data flows. For a special case of a control system with a cyclic information processing algorithm, a simplified formula without scaling functions has been presented for calculating the system service curve.

We have investigated the correlation between the maximum delay estimated by Network Calculus with the results obtained using the statistical analysis of time delay samples. In particular, it has been established that the maximum delay in the data sample and the maximum delay estimated by Network Calculus are closest if the distribution of the sample data has single large overshoots. This property is inherent in heavy-tailed distributions. It has been hypothesized that the maximum delay relates to the probability of a rare event – the sequential arrival of a significant volume of data – under a low server performance for the minimum service curve.

The research presented above allows developing non-statistical estimation methods for time characteristics of digital control systems considering the peculiarities of their operation. Also, it significantly expands the application area of Network Calculus for estimating the parameters of control systems.

The problem of describing closed-loop paths, characteristic of control systems, goes beyond the scope of this paper. A corresponding mathematical apparatus has been developed within Network Calculus; see [6, 31]. However, it has not been properly validated for real systems.

REFERENCES

1. On-line Monitoring of Instrumentation in Research Reactors, *IAEA TECDOC Series*, no. 1830, IAEA: Vienna, 2017.
2. Golshani, G., Taylor, G., and Pisica, I., Simulation of Power System Substation Communications Architecture Based on IEC 61850 Standard, *Proceedings of the 2014 49th International Universities Power Engineering Conference* (*UPEC*), 2014, pp. 1–6. DOI: 10.1109/UPEC.2014.6934745.
3. Ahmad, Z. and Durad, M.H., Development of SCADA Simulator Using Omnet++, *Proceedings of the 2019 16th International Bhurban Conference on Applied Sciences and Technology* (*IBCAST*), 2019, pp. 676–680. DOI: 10.1109/IBCAST.2019.8667158.
4. Chen, L., Zhang, K., Xia, Y., and Hu, G., Scheme Design and Real-Time Performance Analysis of Information Communication Network Used in Substation Area Backup Protection, *Proceedings of the 2012 Power Engineering and Automation Conference*, 2012, pp. 1–4. DOI: 10.1109/PEAM.2012.6612461.
5. Zhang, Z., Huang, X., Keune, B., et al., Modeling and Simulation of Data Flow for VLAN-Based Communication in Substations, *IEEE Systems Journal*, 2017, no. 4, pp. 2467–2478. DOI: 10.1109/JSYST.2015.2428058.
6. Le Boudec, J.-Y. and Thiran, P., *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, Springer Verlag, 2019.
7. Schioler. H., Dalsgaard. J., Larsen. K., and Jessen, J., CyNC – a Method for Real Time Analysis of Systems with Cyclic Data Flows, *J. Embed. Comput.*, 2006, vol. 2, no. 3-4, pp. 347–360.
8. Fidler, M. and Schmitt, J., On the Way to a Distributed Systems Calculus: An End-to-End Network Calculus with Data Scaling, *SIGMETRICS Perform. Eval. Rev.*, 2006, vol. 34, no. 1, pp. 287–298. DOI: 10.1145/1140277.1140310.
9. Promyslov, V.G., Simulation of Computer System with a Flow Alternation between Components, *Control Sciences*, 2012, no. 1, pp. 62–70. (In Russian.)
10. Bouillard, A. and Thierry, É., An Algorithmic Toolbox for Network Calculus, *Discrete Event Dyn. Syst.*, 2008, vol. 18, pp. 3–49. DOI: 10.1007/s10626-007-0028-x.
11. Bouillard, A., Jouhet, L., and Thierry, E., Comparison of Different Classes of Service Curves in Network Calculus, *Proceedings of the 10th International Workshop on Discrete Event Systems* (*WODES*), 2010, pp. 316–321.
12. Byvaikov, M.E., Zharko, E.F., Mengazetdinov, N.E., et al., Experience from Design and Application of the Top-Level System of the Process Control System of Nuclear Power-Plant, *Automation and Remote Control*, 2006, vol. 67, no. 5, pp. 735–747.
13. Ibrahim, W.Z. and Sallam, H., Instrumentation and Controls Architectures in New NPPs, *Int. J. of Nuclear Knowledge Management*, 2014, vol. 6, pp. 283–302. DOI: 10.1504/IJNKM.2014.062830.
14. Cruz, R.L., A Calculus for Network Delay. Part I: Network Elements in Isolation, *IEEE Trans. on Information Theory*, 1991, vol. 37, pp. 114–131.
15. Cruz, R.L., A Calculus for Network Delay. Part II: Network Analysis Information Theory, *IEEE Trans. on Information Theory*, 1991, vol. 37, pp. 132–141.
16. Baccelli, F., Cohen, G., Olsder, G.J., and Quadrat, J.-P., *Synchronization and Linearity: An Algebra for Discrete Event Systems*, New York: John Wiley & Sons, 1992.
17. Ahmadi, H., Martí, J. R., and Moshref, A., Piecewise Linear Approximation of Generators Cost Functions Using Max-Affine Functions, *Proceedings of the 2013 IEEE Power Energy Society General Meeting*, 2013, pp. 1–5. DOI: 10.1109/PESMG.2013.6672353.
18. Camponogara, E. and Nazari, L., Models and Algorithms for Optimal Piecewise-Linear Function Approximation, *Mathematical Problems in Engineering*, 2015, vol. 2015, article no. 876862. DOI: 0.1155/2015/876862.
19. Baybulatov, A.A. and Promyslov, V.G., The Approximation of Envelope in Network Calculus Applications, *Control Sciences*, 2016, no. 6, pp. 59–64. (In Russian.)

20. Vapnik, V.N., *Statistical Learning Theory*, New York: John Wiley, 1998.
21. Baybulatov, A.A. and Promyslov, V.G., Control System Availability Assessment via Maximum Delay Calculation, *Proceedings of the 2019 International Conference on Industrial Engineering, Applications and Manufacturing* (*ICIEAM*), Sochi: IEEE, 2019, pp. 1–6. DOI: 10.1109/ICIEAM.2019.8743012.
22. URL: https://www.dropbox.com/s/zhohii8d9qfqrki/widetable2ru.xlsx?dl=0.
23. URL: https://www.dropbox.com/s/yurroijpayx6p38/Table2en.docx?dl=0
24. Cramér, H., *Mathematical Methods of Statistics*, Princeton: Princeton University Press, 1999. Originally published in 1946.
25. Fidler, M. and Rizk, A., A Guide to the Stochastic Network Calculus, *IEEE Communications Surveys & Tutorials*, 2015, vol. 17, no. 1. DOI: 10.1109/COMST.2014.2337060.
26. RFC 793. Transmission Control Protocol. Protocol Specification, 1981. https://tools.ietf.org/html/rfc793
27. Burnecki, K., Wylomanska, A., and Chechkin, A., Discriminating between Light- and Heavy-Tailed Distributions with Limit Theorem, *PLoS ONE*, 2015, vol. 10(12), article no. e0145604. DOI: 10.1371/journal.pone.0145604.
28. *GOST* (State Standard) *28195-89*: *Software Quality Assessment. General Regulations*, 1989.
29. Smith, M., *Statistical Analysis Handbook. A Comprehensive Handbook of Statistical Concepts, Techniques and Software Tools*, Edinburgh: The Winchelsea Press, 2018.
30. Promyslov, V., Using the Method of Network Calculus to Simulate the Computerized Control System with Non-Uniform Data Flow, *IFAC-PapersOnline*, 2012, vol. 45, no. 6, pp. 645–648. DOI: 10.3182/20120523-3-RO-2023.00273.
31. Baybulatov, A. and Promyslov, V., On a Deterministic Approach to Solving Industrial Control System Problems, *Proceedings of the 2020 International Russian Automation Conference* (*RusAutoCon*), Sochi, 2020, pp. 115–120. DOI: 10.1109/RusAutoCon49822.2020.9208149.

This paper was recommended for publication by V.M. Vishnevsky, a member of the Editorial Board.

Received March 23, 2021, and revised June 15, 2021.
Accepted June 22, 2021.

Author information

Promyslov, Vitalii Georgievich. Cand. Sci. (Phys.-Math.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia, ✉ v1925@mail.ru

Semenkov, Kirill Valer'evich. Cand. Sci. (Phys.-Math.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia, ✉ semenkovk@mail.ru

Cite this article

Promyslov, V.G. and Semenkov, K.V., Estimating Time Characteristics of Control Systems with Cyclic Operation: A Network Calculus Approach. *Control Sciences* **4,** 41–53 (2021). http://doi.org/10.25728/cs.2021.4.5