

Article

# Non-Statistical Method for Validation the Time Characteristics of Digital Control Systems with a Cyclic Processing Algorithm

Vitaly Promyslov \* and Kirill Semenov

V.A. Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences, 117797 Moscow, Russia; semenkovk@mail.ru

\* Correspondence: vitalionics@gmail.com

**Abstract:** The paper discusses the problem of performance and timing parameters with respect to the validation of digital instrumentation and control systems (I&C). Statistical methods often implicitly assume that the probability distribution law of the estimated parameters is close to normal. Thus, the confidence intervals for the parameter are determined on the grounds of this assumption. However, we encountered cases when the delay distribution law in I&C is not normal. In these cases, we used the non-statistical network calculus method for time parameters estimation. The network calculus method is well elaborated for lossless digital system models with seamless processing algorithm depending only on data volume. We consider the extension of the method to the case of I&C systems with considerable changes in the data flow and content-dependent processing disciplines. The model is restricted to systems with cyclic processing algorithms and fast network connections. Network calculus describes the data flow and system parameters in terms of flow envelopes and service curves that are generally unknown in advance. In this paper, we define equations that allow the calculation of these characteristics from experimental data. The correspondence of the Network Calculus and classical statistical estimation methods is discussed. Additionally, we give an example of model application to a real I&C system.

**Keywords:** network calculus; analytical modeling of distributed systems and networks; control system; system validation; performance evaluation; computer and network control and management



**Citation:** Promyslov, V.; Semenov, K. Non-Statistical Method for Validation the Time Characteristics of Digital Control Systems with a Cyclic Processing Algorithm. *Mathematics* **2021**, *9*, 1732. <https://doi.org/10.3390/math9151732>

Academic Editor: Vladimir M. Vishnevsky

Received: 29 April 2021  
Accepted: 20 July 2021  
Published: 22 July 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Modern instrumental and control systems (I&C) for industrial facilities are implemented in most cases as a distributed digital environment. A number of approaches for the validation of the system's timing and performance characteristics have been developed, and they vary depending on the available input data and experts' theoretical backgrounds. Common techniques combine methods of statistical analysis and discrete mathematics [1–5]. The statistical methods usually assume that the distribution law of the measured parameters is close to normal [1]. In most cases, this assumption is valid for signals having a physical nature, but, as we will show, it can be false for signals describing the digital I&C itself, for example, data communication and information processing delays.

The network calculus method is one of the alternatives for evaluating computer network performance characteristics [6]. It is the non-statistical method of analyzing deterministic queuing systems based on mini-plus algebra and is attractive since it does not use assumptions about the probability distribution law of the values. The distinctive feature of the network calculus method is the use of specific functions to calculate delay and buffering parameters (service curves and envelopes of input and output data flow). Initially, network calculus was developed for the analysis of lossless digital streaming systems. The term lossless means that there are no data sources and sinks inside the system. Generally speaking, I&C systems do not belong to those systems since the following features characterize them:

- parallel processing of multiple tasks on same computing resource;
- significant change in the amount of information at the input and output of a component (for example, when a component compresses information);
- heterogeneity of data in a digital control system, in contrast to the information in streaming systems; it means that each element of data (bit) has its own value and can be processed according to a proper algorithm.

It is not to say that these features are not considered in the context of network calculus. The work [7] extends the method to systems with cyclic dependencies between input and output flows. Several works [8,9] give approaches for using network calculus in systems with significant differences between the input and output flow. Several papers [10,11] deal with various disciplines of joint processing of multiple tasks on a shared computing resource.

These approaches have common drawbacks. First, their application requires precise knowledge of the system's internal features and, being tied to it, they are sensitive to any change in the system's operating modes. Second, for complex systems, they lost the "transparency" of the results. Thus, a simple correlation with other characteristics (input data rate, burstiness, computational power of the component) becomes awkward.

In this paper, having analyzed these drawbacks, we develop a mathematical model of the digital I&C system. The model retains the generality and computational transparency with a possibility to take into account the uneven long-term relation between the input and output flows and heterogeneity of data. In the framework of the model, we investigated two subproblems that are of separate interest:

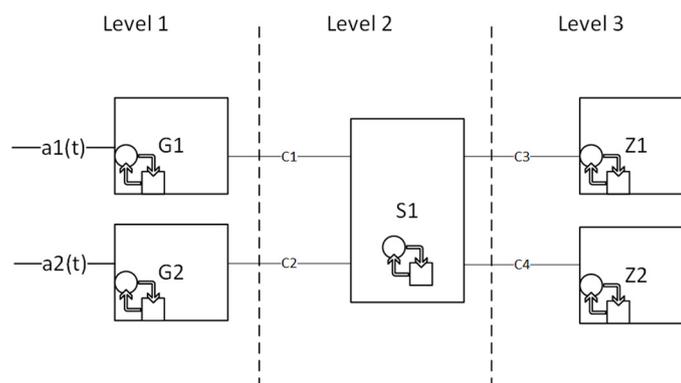
- comparing the network calculus computed delays with delays obtained by the statistical methods;
- calculation of the flow envelope and service curves from experimental data.

Both problems have not been adequately presented in the literature yet.

The model has been checked on simulated data and data obtained in a real I&C system [12].

## 2. The Structure of a Typical Industrial Automation Control System

The work considers a typical I&C system for an industrial facility (from now on, referred to as Control System—CS). The architecture of the CS has three levels (Figure 1):



**Figure 1.** Block diagram of the control system under study.

- Equipment embedded controllers and low-level data gateways—level G (1);
- Data storage and data processing servers—level S (2);
- Human-machine interface components—level Z (3).

The  $a(t)$  and  $a^*(t)$  with numeric index are the input and output in a particular channel. A similar CS structure is used in various applications for real objects [12,13].

In accordance with the practical operating conditions of the CS at the facility, elements  $(G, S, Z)$  of the system implement a cyclic data processing algorithm.

**Definition 1.** A cyclic data processing algorithm is an algorithm with the following properties. At the start, the system element is in a state of waiting for the data to arrive. Sequentially incoming data packets are processed in a determined uniform way, and the system returns to its original state after the last packets in the sequence are processed.

For the cyclic algorithm, the total processing time of the data packet  $D_C$  can be represented as the sum of two values:

$$D_C = T_E + T_S,$$

where  $T_E$  is network delay and  $T_S$  is data processing time within system elements.

### 3. Application of Network Calculus to Evaluate Control System Time Characteristics

Let us consider the aspects of applying the network calculus method for computing time characteristics of a CS.

The network calculus [6] method was invented by Cruz [14,15] and is based on min-plus algebra (see [16]). The basic terms and notations of the method can be found in [6] or [17]. In this section, we present only the most important definitions.

**Definition 2.** A function  $\beta$  is the (minimum) service curve of a network element (or system) with the input flow  $A$  if  $\beta$  is a causal flow function and the element (system) output flow  $A^*$  satisfies:

$$A^* \geq A \otimes \beta, \quad (1)$$

where  $\otimes$  is mini plus convolution operator.

**Definition 3.** A function  $\gamma$  is the (maximum) service curve of a network element (or system) with the input flow  $A$  if  $\gamma$  is a causal flow function and the element (system) output flow  $A^*$  satisfies:

$$A^* \leq A \otimes \gamma. \quad (2)$$

**Definition 4.** A function  $a$  is the envelope of flow  $A$  if  $A \leq A \otimes a$  or, that is the same

$$a \geq A \oslash A, \quad (3)$$

where  $\oslash$  is mini plus deconvolution operator.

The input and output flows are determined by the total amount of data registered at the input and output over a certain period. Therefore, the time of data passage through the system will accordingly be defined as the horizontal deviation  $d(t)$  between these functions.

**Definition 5.** (The maximal delay in the system): For linear systems with input flow  $A$ , the output flow  $A^*$ ,  $A(t) \geq A^*(t)$ , the maximal delay  $D_{\max}$  expressed as the maximal horizontal distance between input and output flow curves is:

$$D_{\max} = h(A, A^*) = \sup_{t \geq 0} \{ \inf \{ d \geq 0 : A(t) \leq A^*(t + d) \} \}, \quad (4)$$

One of the crucial points of network calculus is that the maximum value of delay  $D_{\max}$  can be obtained using bounds rather than real cumulative functions, see the proof in [6]:

$$D_{\max} = h(a, \beta), \quad (5)$$

where  $h$  is the maximal horizontal distance between two curves.

#### 4. Network Calculus Main Curves Estimation Approaches

Let us consider the problem of determining the flow envelope, the minimum and maximum service curves of the system, and their linear approximations based on the flow data measured during the experiment.

##### 4.1. Calculation of the Flow Envelope Based on Experimental Data

Equation (3) defines a direct method for calculating the envelope of the cumulative flow  $A$ . It is convenient to deal with a piecewise linear approximation of the envelope, which is reduced to the affine function  $y = kx + b$  in some cases. The piecewise linear approximation allows using effective computational algorithms of data processing [18,19]. The affine function approximation allows to quickly perform system analysis and make numerical estimations of the system behavior [10]. The flow envelope approximation in the form of an affine function in the Network Calculus method was considered in [20]. The work [21] considers methods for calculating a one-component linear flow envelope based on support vector machine algorithms.

##### 4.2. Maximum and Minimum Service Curves Calculation Approaches Based on the Experimental Data

Determining the parameters of the service curve is not as easy as for envelope. Theoretically, we could get service curves as a strict bound using specially designed test flow and taking into account the convolution property when zero element for  $\wedge$  is absorbing for mini plus convolution operator  $\otimes$  [6]. However, the experiment is unrealistic because it would require a flow described by  $\delta(t)$ , that would overload any real system.

The second approach is using a relation between mini deconvolution and convolution operators [6]:

$$C \geq B \oslash A \Leftrightarrow B \leq A \otimes C, \tag{6}$$

Using (6) and the definition of min plus deconvolution [6], we obtain a lower bound for the maximum service curve:

$$\gamma' = A^* \oslash A, \tag{7}$$

where  $A$  and  $A^*$  are input and output flow, respectively.

Meanwhile, the estimation of only the maximum service curve is often not sufficient for system analysis. For example, the calculation of maximal system delay and buffer size requires the minimum service curve (1). We are not aware of any satisfactory methods of experimental minimum service curve calculation.

Below, we will introduce an approach to the calculation of the minimum service curve. It is similar to the approach of maximum service curve calculation, but uses a “weak” property that we are going to prove.

**Proposition 1.** Let  $A, B, C \in \mathcal{F}$  and if  $C \leq B \overline{\oslash} A$ , then

$$B \geq A \otimes C, \tag{8}$$

**Proof.** Let  $C(s) \leq (B \overline{\oslash} A)(s)$  for  $s \in \mathbb{R}$ . Let us write explicitly for some  $s, v \geq 0$ :

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

or

$$B(s + v) \geq C(s) + A(v), \tag{9}$$

$$t = s + v$$

Rewrite (9) as

$$B(t) \geq A(t - s) + C(s), \tag{10}$$

The inequality (10) is valid for any  $s, t \geq s \geq 0$ ; it is also valid for some  $s$  where the expression gains the low boundary, that is

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

The proved proposition allows us to estimate the minimum service curve. Let  $A$  be cumulative input flow,  $A^*$  be cumulative output flow. Then, by virtue of the proved proposition for the curve:

$$\beta' = A^* \overline{\otimes} A, \quad (11)$$

The inequality  $A^* \geq A \otimes \beta'$  is satisfied; it means, in turn, that  $\beta'$  is the estimated minimum service curve  $\beta' \leq \beta$ .

Since property (8) is only a necessary condition, the estimation of the minimum service curve obtained by Equation (11) can lie both above and below the real minimum system service curve. Comparing Equations (6) and (8), we also note that  $\beta' \leq \gamma'$ , i.e., the minimum service curve is bounded from above with the maximum service curve. Note, partially the result of the Equation (16) might be negative. Then we consider for a physically realizable system only the positive part:  $\max(0, \beta')$ .

In a particular case, when a system does not have a maximum service curve (i.e., when there is a mode of "instant" processing of the input data), it is possible to obtain an exact value for the minimum service curve by using the envelopes instead of the input and output cumulative flow. To do this, let us assume that  $\alpha, \alpha^*$  are envelopes of the input and output flows, respectively. It is known that  $\alpha^* = (\alpha \otimes \gamma) \otimes \beta$  (see [6] p. 34). Then the equation  $\gamma(t) = \delta(t)$  can be rewritten as:

$$\alpha^* = \alpha \otimes \beta, \quad (12)$$

and by the property of the operator  $\otimes$  (see [6] p. 123) and the minimum service curve  $\beta$ :

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

Using the commutativity of the operator  $\otimes$  and applying the same property in the reverse direction, we obtain the estimation of the minimum service curve:

$$\beta' = \alpha^* \otimes \alpha, \quad (13)$$

If the service curves can be described by affine functions, then there are fast convolution and deconvolution algorithms for them, which are necessary for calculating system parameters [11]. As shown in [22], the service curves can be approximated by the affine functions similar to the flow envelope and with the same algorithms based on the support vector machines algorithm.

## 5. I&C System Model and CS Time Characteristics Validation

Let us consider a problem of modeling a typical CS presented in Figure 1 using the network calculus framework. Additionally, let us suppose the CS has excessive computing resources. This allows us to decompose the system and consider each logical channel of the system separately. If this condition is not met, it is necessary to take into account the discipline of resource sharing (see, for example, one of the of the task scheduler models [11]). Figure 2 shows a separate channel of a CS. Each component in the channel has its own maximum and minimum service curves.

All main conclusions and equations in the section will be given for the minimum service curve. As follows from the definitions of the maximum and minimum (1, 2) service curves, the conclusions and equations for the maximum service curve will be similar and can be written by simply renaming the variables and replacing the inequality signs in the relations.

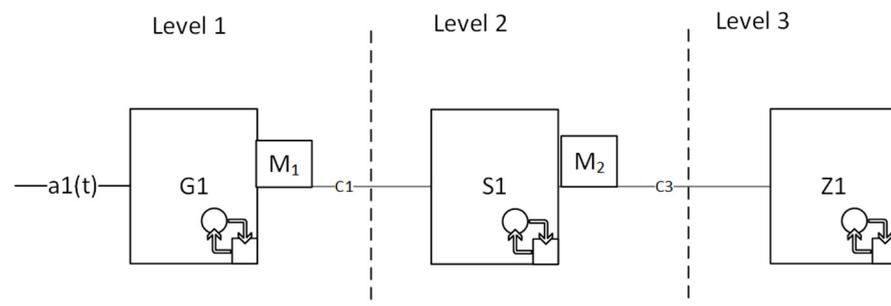


Figure 2. Dedicated logical transmission channel in a control system.

To designate a specific component, we will add a lower alphanumeric index to  $\beta$  following the notation in Figure 2. Cumulative input and output flows at each component will be denoted as  $A, A^*$ .

Then, in accordance with the definition of minimum service curve (2), for every system element, we can write an expression in the following form:

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

However, in practice, all CS elements' characteristics, except the communication conduits, are not linear; the scale of the flow changes between the input and output. For example, a single alarm signal at the component input can cause an avalanche of related signals, which will lead to an increase in information at the output of the component. To describe the change in the flow scale, the scaling function  $M$  and its inverse function  $M^{-1}$  are introduced into the model (Figure 2). The scaling function provides the transformation  $M : A^* \rightarrow A$  and  $M^{-1} : A \rightarrow A^*$  [8]. So, the service curve of the  $\beta_{Si}$  system for the  $i$ -th channel with regard to the scaling functions is expressed as:

$$\beta_{Si} = \beta_{Gk} \otimes M_1^{-1}(\beta_{Cn1} \otimes \beta_{Sl} \otimes M_2^{-1}(\beta_{Cn2} \otimes \beta_{Zm})) \tag{14}$$

where  $i, k, l, m, \in \mathbb{N}$  are indexes of connected in sequence components in a data processing logical channel at each CS level;  $n1, n1 \in \mathbb{N}$  are the indexes of communication channels used in data transmission between the components in the channel  $i1$ ; and  $M_1, M_2$  are the scaling functions of the corresponding components. The service curves  $\beta_{Cn1}, \beta_{Cn2}$  reflect the network data transmission delay  $T_E$ , and the rest refers to the data processing delay  $T_S$  in the component.

Provided that service curves calculation and scaling functions for each of the components is possible, Equation (14) allows to obtain the bounds for data processing delay in the entire system, depending on the characteristics of the input flows  $ai(t), i \in \mathbb{N}$ . However, in practice, calculating the scaling functions  $M$  of a real system is a complicated and not always solvable problem.

To avoid difficulties with the definition of scaling functions, we will redefine the input and output flows and move from real flow to virtual one for systems with a cyclical data processing algorithm.

Let us suppose all data received by the system at the beginning of each cycle will be processed and transmitted to the output by the end of the cycle, and consider the following function:

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

where  $j$  is the number of the cycle,  $\tau_j$  the cycle, and  $j$  duration. Next, let introduce on an interval  $[0, +\infty)$  a step function  $Q(x)$ :

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

The step function  $Q(x)$  is by definition a flow function.

The output flow  $Q^*$  for a component can be obtained from the input flow by shifting it to one cycle:

$$Q^*(x) = \begin{cases} Q(x - 1), x \in [1, +\infty) \\ 0, x \in [0, 1) \end{cases} \tag{15}$$

Note: the definition of the maximal delay for virtual flows must be redefined as:

$$D_{\max} = v(Q, Q^*) = \sup_{t \geq 0} \{Q(t) - Q^*(t + d)\}.$$

The structural diagram (Figure 2), redefined for virtual flows in components of the G, S, Z type, is shown in Figure 3.

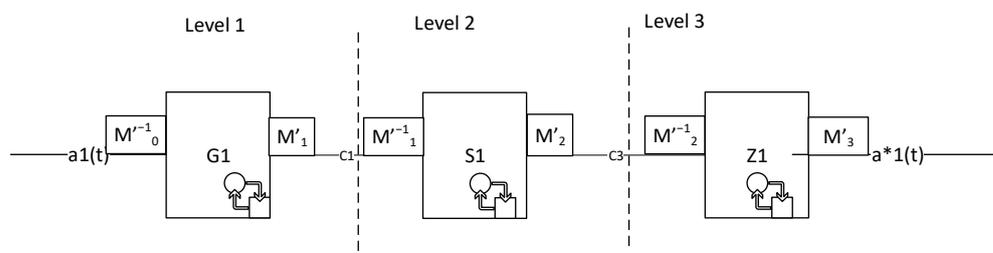


Figure 3. Dedicated logical transmission channel i1 in the control system.

For a channel with “virtual” flows for components G, S, Z, we redefine the meaning of service curves  $\beta$  and  $\gamma$  as:

$$\begin{aligned} Q^* &\geq Q \otimes \beta \\ Q^* &\leq Q \otimes \gamma \end{aligned}$$

and introduce mapping operators:  $M_0', M_1', M_2'$  providing the transformation  $M' : Q^* \rightarrow A$ , and the inverse mapping operators providing the transformation  $M^{-1'} : A^* \rightarrow Q$ . Then the service curve for the system shown in Figure 3 will look like this:

$$\beta_{Si} = \beta_{Gk} \otimes M_1' \otimes \beta_{Cn1} \otimes M_1^{-1'} \otimes \beta_{S1} \otimes M_2' \otimes \beta_{Cn2} \otimes M_2^{-1'} \otimes \beta_{Zm}, \tag{16}$$

In turn, Equation (16) can be reduced (see [8] Section 5.1) to a more convenient form by transferring the scaling functions  $M'$  from the input to the output of the component and omitting the pair  $M', M^{-1'}$  by component output:

$$\beta_{Si} = \beta_{Gk} \otimes M_1^{-1'}(\beta_{Cn1}) \otimes \beta_{S1} \otimes M_2^{-1'}(\beta_{Cn2}) \otimes \beta_{Zm}, \tag{17}$$

The partial transition from  $A$  to  $Q$  in Equations (16) and (17) does not generally simplify the work with the scaling functions. However, if:

$$\beta_{Ci} \gg \beta_{\{G,S,Z\}i} \tag{18}$$

one can replace  $\beta_{Ci}$  by a function that is neutral with respect to the mini-convolution operator  $\delta(t)$ :

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

with the following property:  $f = \delta \otimes f$  (see, for example, [6]).

For a monotonic scaling function:

$$M^{-1}(\delta(t)) \rightarrow \delta(n), n \in \mathbb{N}$$

it becomes possible to omit  $\beta_{Ci}$  from Equation (17) and, accordingly, to get rid of the scaling functions. Physically, the Assumption (18) means that the processing cycle time in the network stack and the time of information transmission over the system network are negligible compared to the time of information processing on a computing resource. This assumption is mainly fulfilled in modern CS, where the transmitted information has a relatively small volume compared to the bandwidth of communication channels.

In this case, the overall system service curve in Equation (17) for the  $i$ -th chain for the “secondary” virtual flow is simplified:

$$\beta_{Si} = \beta_{Gk} \otimes \beta_{Sl} \otimes \beta_{Zm}, \quad (19)$$

where  $k, l, m \in \mathbb{N}$ —are the indexes of sequentially connected components.

## 6. Network Calculus Method Application Verification for System’s Time Characteristics Estimation

### 6.1. Reference Data and Verification Procedure

The delay is calculated using an input flow envelope and service curve, which are not measured directly but are the result of calculations. It is clear that the methods used to compute them will also affect the trustworthiness of the final result. Therefore, let us focus on the practical aspects of the delay computation and compare the results of network calculus and statistical analysis.

The network calculus method was verified using test data with known statistical parameters. The test program simulates the CS component with a cyclic operation algorithm. The network delay  $T_E$  and cycle time  $T_S$  were random variables distributed according to a certain law. For data analysis, we use the free network calculus library [10].

### 6.2. Comparing the Network Calculus and Statistical Results

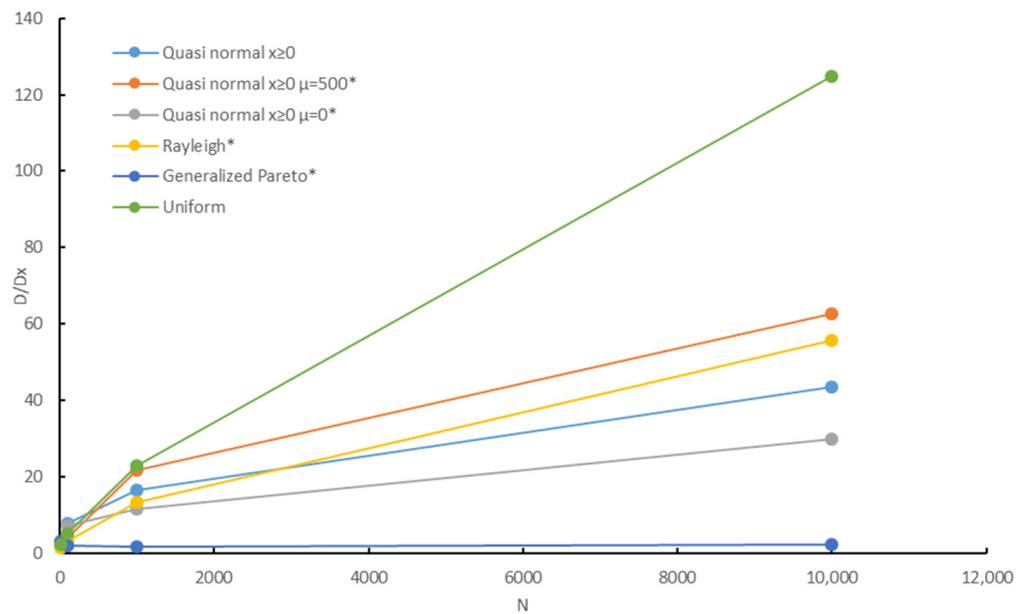
For each sample, we compute:

- delay ( $D$ ) calculated by the network calculus (5) using experimental maximum and minimum service curves;
- maximum measured delay in the sample ( $D_x$ );
- ratio of  $D/D_x$ ;
- dependence of the maximum calculated delay on the sample size ( $L$ ) and distribution.

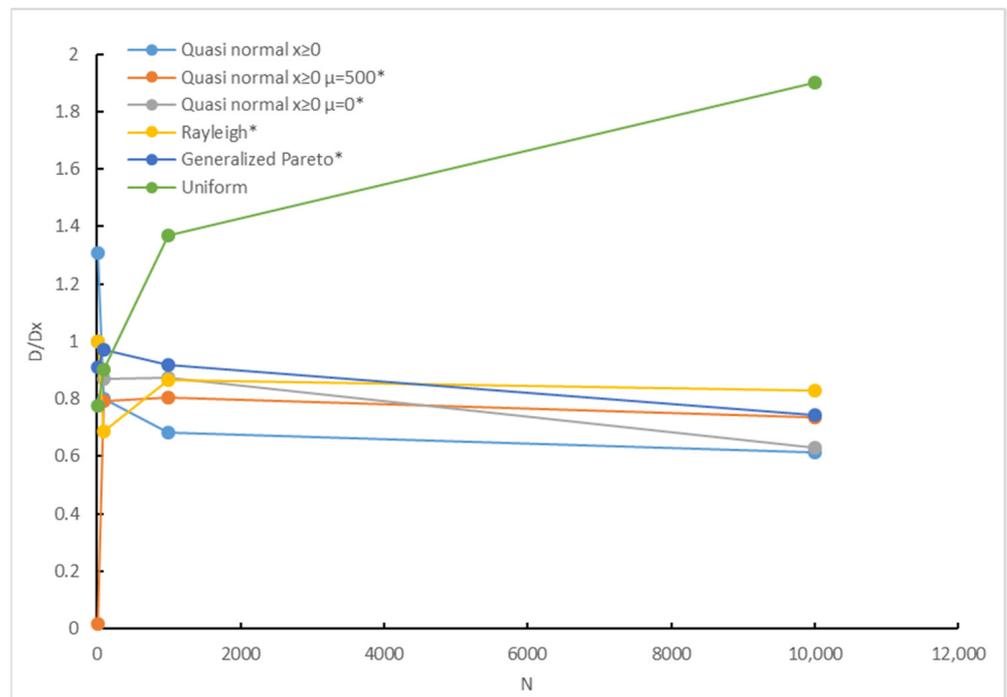
The data in the samples have different distributions, including distributions that were close to normal and those with heavy tails. In the experiments we assume that the data was completely processed:  $\exists t_e \geq 0, A(t_e) = A^*(t_e)$ . All data presented have been rounded to ~1%. When using the distribution law with possible negative values, the negative data were discarded.

Figure 4 shows the dependence of the ratio  $D/D_x$  on the sample size for different probability distribution functions, where  $D$  is the maximum delay in the network calculus method with the service curve calculated by Equation (13).

Figure 5 shows the dependence of the ratio  $D/D_x$ , where  $D$  is the calculated maximum delay in the network calculus method with the service curve calculated with the use of Equation (7) for different probability distribution functions and sample sizes.

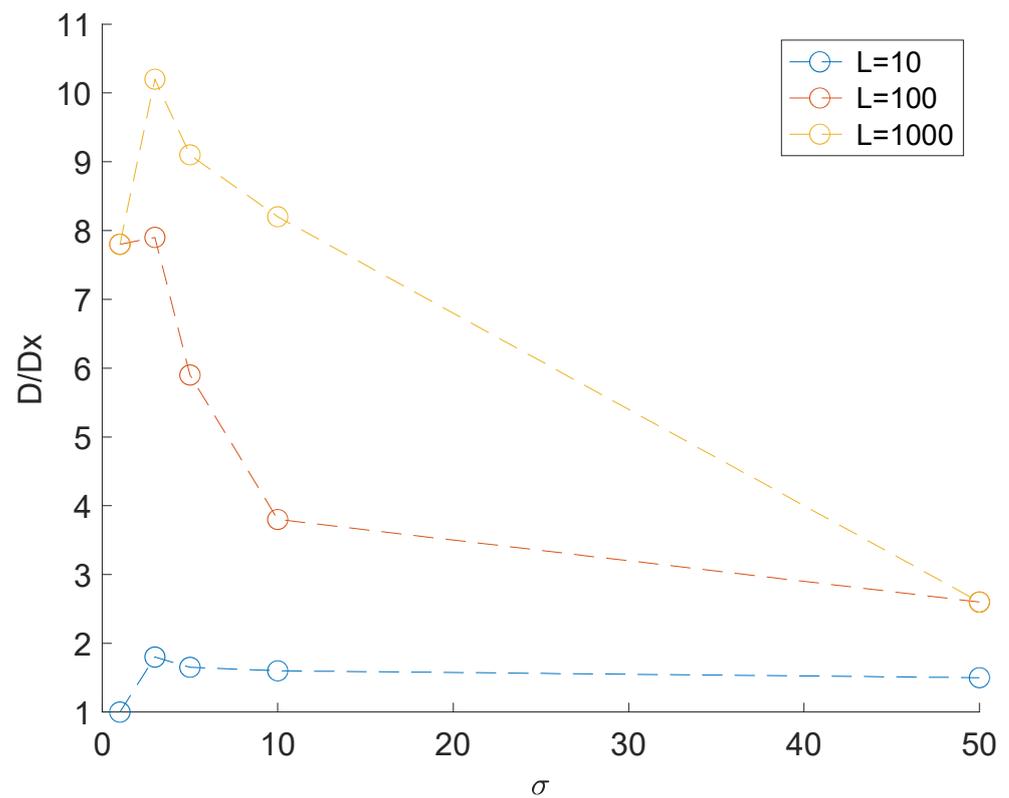


**Figure 4.** Dependence of the maximum delay estimate on the sample size for minimum service curve  $\beta'$ ,  $N$  is a sample size. The symbol \* means the data is measured in a real computational environment; the data without \* are generated with a random number generator.



**Figure 5.** Dependence of the maximum delay estimation on the sample size for the maximum service curve  $\gamma'$ ,  $N$  is a sample size. The symbol \* means the data is measured in a real computational environment; the data without \* are generated with a random number generator.

The dependence of  $\frac{D}{D_x}$  on the sample size and outlier amplitude using the minimum service curve is shown in Figure 6.



**Figure 6.** The ratio of the measured and calculated delay for the Rayleigh distribution with parameters  $\mu = 0, \sigma = 300 B$  depending on the magnitude of a single outlier in  $\sigma$  for several fixed sets of sample lengths.

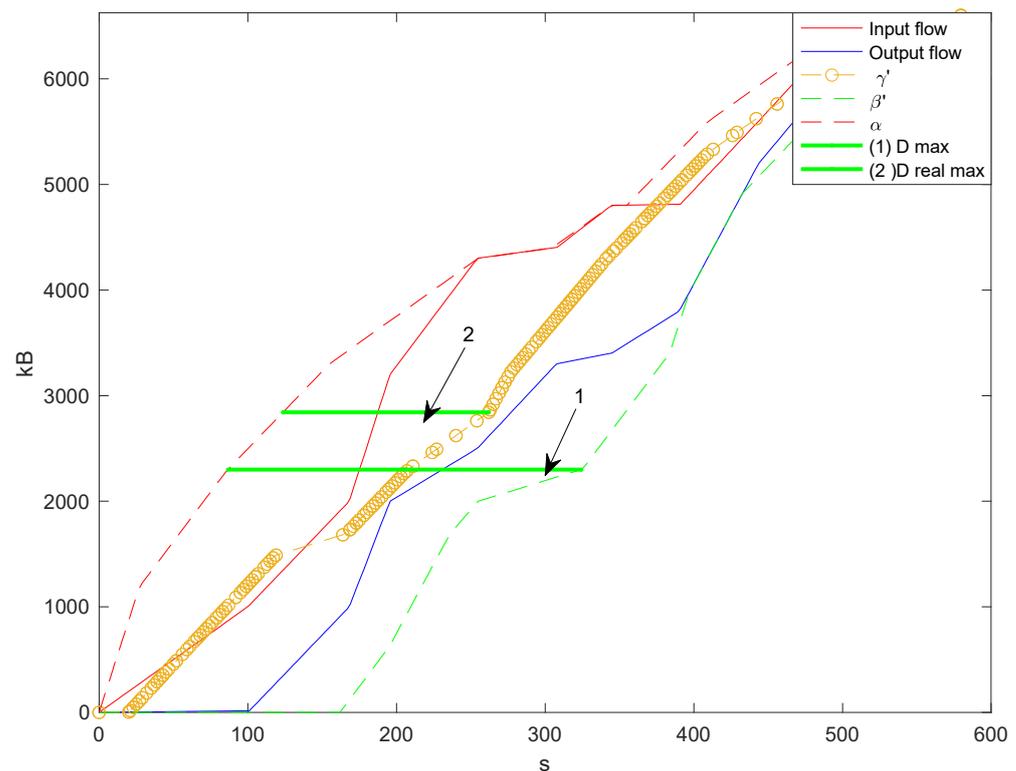
The experiments allow us to make the following observations and conclusions about the relation of the statistical and network calculus results.

The maximum delay estimation with the use of a minimum service curve (13) is more accurate for short samples (Figure 6) and heavy-tailed distributions (Figure 4). The  $\frac{D}{D_x}$  ratio increases with increasing sample size, although the rate of delay changes decreases with the increasing sample. The  $\frac{D}{D_x}$  ratio can be as high as  $10^2$ .

Meanwhile, the simulation shows that the maximum delay calculated with the use of service curve (7) characterizes the delay in the normal operating mode (Figure 5). The delay is close to the maximum delay in the sample and depends weakly on the sample size for sufficiently large samples.

The maximum delay estimation with the use of service curve (7) is close to the experimental maximum delay  $D_x$  but  $D$  is commonly somewhat less than  $D_x$ . The resulting estimate better correlates with the real maximum delay during an increase in the sample size and for distributions close to normal (Figure 5).

Figure 7 shows a typical shape of the network calculus curves obtained from the experimental network data. The sample data follow quasi-normal distribution with  $\mu = 500, \sigma = 100$ . For clarity, the data of a short size are given. The lower horizontal line corresponds to the maximum delay calculated for the maximum service curve (7). This delay is close to the maximum sample delay. The upper horizontal line corresponds to the estimation of the maximum delay for the maximum service curve (11). The figure clearly shows that the input flow envelope limits all curves on the figure from above, and an estimation of the minimum service curve limits them from below.



**Figure 7.** Experimental Network Calculus curves for a quasi-normal distributed sample with  $\mu = 500$ ,  $\sigma = 100$ ; size of  $L = 10$ . Arrow 1 points to the maximum delay for the service curve (7), arrow 2 points to the maximum delay estimation for the service curve (11).

### 6.3. Comparison of Network Calculus and Statistical Calculation Results

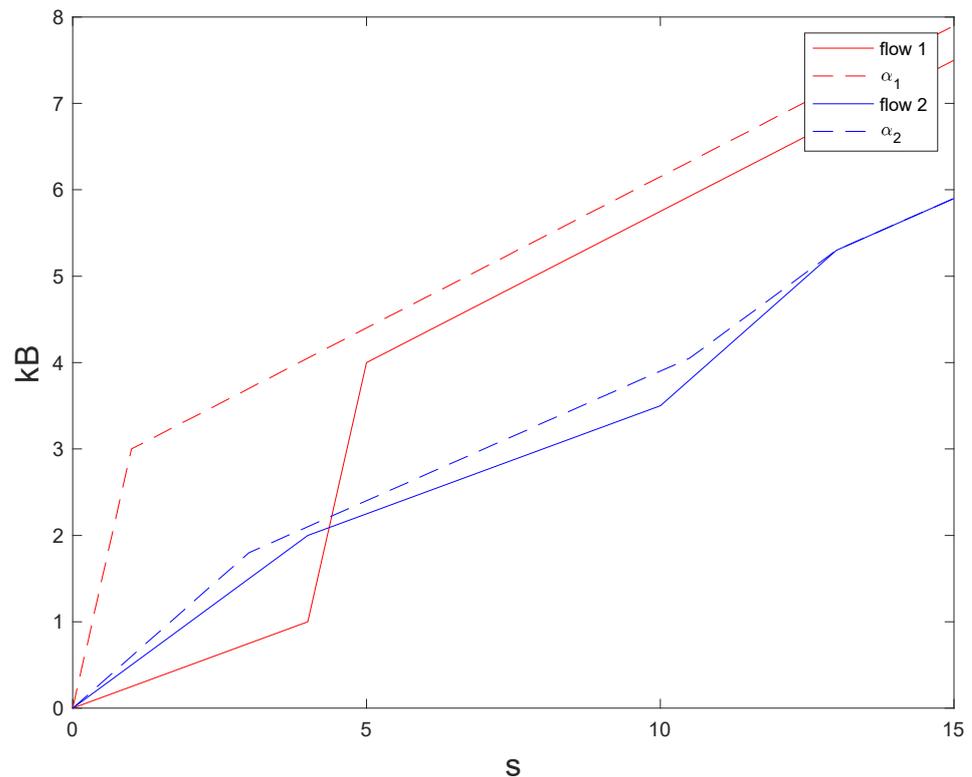
Our simulations have shown that the  $\frac{D}{D_x}$  ratio values and the maximum delay depend on the probability distribution law of processing time and on the sample length and the number and amplitude of individual outliers in the data.

The dependence is complex due to the non-linear nature of the formulas describing the basic operations. According to them, the flow envelope and service curve will have sections composed of data close in value, sorted in descending order for the flow envelope and the maximum service curve and ascending for the minimum service curve (CM. [6] p. 113).

Thus, with the sample length increase, the flow will contain a larger number of sections with a significant steepness. Therefore, both the flow envelope (8) and the service curves are calculated by Equations (7) and (11) will change.

The maximum service curve (7) estimation from the sample will have similar behavior with the flow envelope. The minimum service curve has an opposite tendency (Figure 7). Therefore, maximum delay estimation by Equation (7) is less dependent on changes in the input data and the length of the sample.

A heavy-tailed probability distribution law is characterized by the presence of a certain number of outliers, which are very different from the rest of the values. For distributions close to normal, the appearance of such outliers in the sample is less likely, but they are nonetheless characterized by the presence of a sufficient amount of data within the confidence interval. So, the overall curvature trend for the envelope and service curves will differ depending on data distribution. For samples with single large outliers, the curves will have large curvature at the beginning and a subsequent sharp decrease. For samples without large outliers, the curvature will decrease smoothly (Figure 8).



**Figure 8.** An example of two cumulative flows (flow 1, flow 2) and envelopes ( $\alpha_1, \alpha_2$ ) for them. Flow 1 has outliers in the data (point 4 on the X-axis), and flow 2 has no significant outliers.

This lets us understand the relationships between the delay estimation obtained from the network calculus and classical methods of statistical estimation (see, for example, [23]).

It is known [24] that network calculus implicitly assumes the worst combination of conditions for information processing in the system. Graphically, it means that the beginning of the envelope curve consists of areas with the most significant changes in the input flow (i.e., a worst-case scenario that can be predicted from the observed data). For the delay based on the minimum service curve, the worst-case scenario is also the arrival of the largest packet of data 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 performance (i.e., when the maximum amount of data accompanies the system's maximum performance), which is typical for the system's normal operation.

In both cases, the delay calculated using network calculus corresponds to the delay calculated for the two described scenarios for statistical methods of calculation. The probability that an actual delay will reach this value corresponds to the probability of this scenario being realized in the experiment.

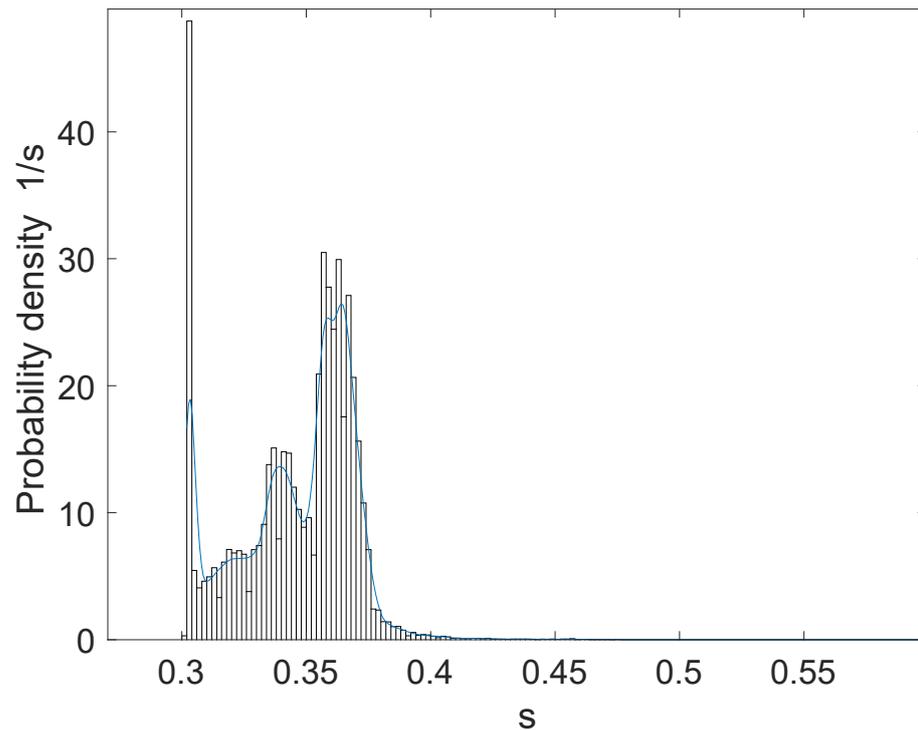
For verification, we calculated the probability that the real delay will be less than the delay calculated by the network calculus method. For the delay calculated using the minimum service curve, in most cases, as expected, this probability is close to the unity.

## 7. Practical Example of Calculating the Delay for a Real I&C System

This section presents the results of evaluating the time characteristics of the actual control system described above (Figure 1). In order to substantiate the possibility of using the simplified Formula (18) for calculating the service curve of the entire system, the network delays of data transmission between the components were also measured. For the measured values, empirical probability distributions are calculated, and for the network delay, spectral characteristics are additionally analyzed.

Measurements were carried out for elements of level Z (from Figure 2). The amount of cyclic data being processed is relatively stable under normal operating conditions and has some average speed. However, with some special (actuation of protections and equipment interlocks) or transient processes (transition from mode to mode), the amount of data and the algorithm (that is, speed) of processing can change significantly.

The empirical distribution of the cycle time  $T_S$  (Figure 9) is different from the normal or Poisson and is polymodal. In accordance with the functioning algorithm, each of the modes corresponds to a typical processing cycle for a particular type of data.



**Figure 9.** The empirical probability density function of the cycle time of the Z component. The solid line shows the smoothed approximation of the distribution.

For this sample, the maximum delay was estimated using service curves 7 and 11. The results for sample size  $L \sim 10^3$  are shown in Table 1.

**Table 1.** The experimental delay calculation results for I&C system component Z.

Parameter	Value
$D_x$	0.37
$p(D_x)$	$\sim 1$
$D$	5.1
$p(D)$	$\sim 1$
$D/D_x$	4.9
$D'$	0.32
$p(D')$	0.87
$D'/D_x$	0.3

In the experiment, we measured the network delay  $T_E$  using the tcpdump utility from the Linux OS. The round-trip time (RTT) of a TCP packet was measured, i.e., the time elapsed from the moment the S component sent packet until the confirmation (ACK, [25]) receipt from the Z component. The typical round trip time of a packet is tens of microseconds and the maximum transit time of network packets is  $\sim 10^3$  times less than the processing time of information in cycles. The characteristics of the process of transferring data between

other components of the system are similar. The condition that  $T_E \ll T_s$  allows using the simplified Formula (18) to calculate the service curve.

In the course of measurements on a real I&C system, we verified that the empirical distributions obtained from measurements (Figure 9) have a heavy tail. The check was carried out using the algorithm for recognizing distributions with heavy tails [26], which, on the author's tests, showed better results than tests based on the Kolmogorov–Smirnov criteria.

The test showed that a real system of distribution of time delays, both in the network components and in the components that process information, belong to a heavy-tailed distribution.

## 8. Results and Discussion

The paper considers estimating the temporal characteristics of digital I&C control systems (CS) during the system's validation. CS technical requirements often include constraints on data processing and communication delays. The constraints can be imposed on both average and maximum (absolute) values. They can be expressed either as statistical constraints with confidence intervals or as limiting absolute values [27].

Estimating a random variable from a sample is a classical problem of statistics. It is well described in the literature (see, for example, [28]). However, the interpretation of the measured characteristics of the digital I&C system with a presumption that the probability law of the values is close to normal may lead to incorrect conclusions. Let us formulate the main problems.

The procedure of technical requirements validation during tests is mainly based on calculating the sample mean and sample variance (for example, [1]). If a random variable has a finite expectation value and variance, the sample mean is an unbiased consistent estimation of the theoretical mean and does not depend on the type of distribution. A known disadvantage of this method is its low robustness in extraneous outliers in the sample [28]. However, sample variance, both biased and unbiased, is a consistent estimate of the theoretical variance of a quantity.

In practice, when interpreting the obtained estimates of the mean and variance, it is implicitly assumed that the time delays are distributed according to the normal law and intuitively transfer the estimates of the confidence intervals for the normally distributed quantity to the case of time delays in control systems. Indeed, if a random variable has a normal distribution then, having a sample mean and variance, it is easy to estimate the confidence interval for the parameter being confirmed. However, the probability function of the delay in the CS is generally not normal.

The physical nature of the measured value (time) imposes restrictions on the probability distribution function. At least, it is bounded on the left. If the technical requirements specify the maximum absolute value (for example, "the signal transit time between the CS components should not exceed a certain value") this form of the requirement implies that the random variable has a distribution function that is also bounded from the right. So, the absolute restrictions mean that the distribution function is not initially, in the strict sense, a distribution function of a normal random variable.

Our study of a real CS encountered that the distributions of the data processing and communication delays significantly differ from the normal, often have polymodal nature, and belong to heavy-tailed distributions. In a general sense, to estimate the probability of a random variable exceeding a particular value, one can use Samuelson's–Chebyshev's inequality. However, it gives a very rough estimate.

The paper considers a non-statistical approach for delay calculation in CS based on the network calculus method. The network calculus method is not entirely new, but it is still not well understood by testing specialists. When applying it to the analysis of computer systems, it is necessary to take into account some of the peculiarities of the method. Thus, the input data about the system, which are necessary for the calculation using the network calculus method, in the general case, are not specified as "logbook parameters" of the system. For example, such input data for the method are flow envelopes, service curves,

scaling functions in the case of an uneven data flow, etc. are *a priori* unknown. There is also the lack of transparency in corresponding the network calculus results with classical (statistical) methods. The technical difficulties of the method are known, and various approaches to partially resolve them have been developed, for example, [8–11,19,29]. However, these solutions also require initial data about the system, which are absent or poorly formalized in practice.

Within the frames of network calculus, we proposed a mathematical model for the description of computer systems with cyclic data processing algorithms that are common in CSs. The model allows one to take into account data heterogeneity and significantly simplifies delay calculations if the major delay is due to the data processing.

We got solutions for two subproblems. First, we proved a necessary condition of the minimum service curve existence that allows estimating the curve from data flows. Second, we performed simulations of the correspondence of network calculus and statistical results.

In particular, it is shown that the closest correspondence between the statistically calculated maximum delay and the calculation of the maximum delay by the Network Calculus method is obtained when the data distribution in the sample has single large outliers, which is typical for heavy-tailed distributions. It is assumed that the maximum delay is related to the probability of a rare event, a sequential arrival of a significant amount of data with low server performance for a minimum service curve.

The developed methods are verified on simulating examples and successfully applied to the real I&C system.

We have not managed to formulate a sufficient condition on the minimum service curve. It is possible that a sufficient condition for a general case does not exist. However, we hope to obtain a sufficient condition for the particular case of realizable service curves.

**Author Contributions:** Conceptualization, V.P. and K.S.; methodology, V.P.; formal analysis, K.S.; investigation, V.P. and K.S.; writing—original draft preparation, V.P. and K.S.; writing—review and editing, V.P. and K.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** The reported study was partially (Sections 3 and 4) funded by RFBR, project number 19-29-06044.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** We are grateful to A.A. Baybulatov for fruitful discussions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. IAEA. *On-Line Monitoring of Instrumentation in Research Reactors*; IAEA TECDOC Series, No. 1830; IAEA: Vienna, Austria, 2017; p. 80.
2. Golshani, G.; Taylor, G.; Pisica, I. Simulation of power system substation communications architecture based on IEC 61850 standard. In Proceedings of the 2014 49th International Universities Power Engineering Conference (UPEC), Cluj-Napoca, Romania, 2–5 September 2014; pp. 1–6. [\[CrossRef\]](#)
3. Caiko, J.; Kunicina, N.; Patlins, A.; Galkina, A. Analysis and simulation of hybrid network for industrial automation. In Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, Prague, Czech Republic, 15–17 September 2011; pp. 248–252. [\[CrossRef\]](#)
4. Morato, D.; Perez, C.; Magana, E.; Izal, M. Network Simulation in a TCP-Enabled Industrial Internet of Things Environment—Reproducibility Issues for Performance Evaluation. *IEEE Trans. Ind. Inform.* **2021**. [\[CrossRef\]](#)
5. Zhang, Z.; Huang, X.; Keune, B.; Cao, Y.; Li, Y. Modeling and Simulation of Data Flow for VLAN-Based Communication in Substations. *IEEE Syst. J.* **2017**, *11*, 2467–2478. [\[CrossRef\]](#)
6. Le Boudec, J.-Y.; Thiran, P. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*; Springer: Berlin/Heidelberg, Germany, 2019; p. 245, Online Version of the Book, LNCS.
7. Schioler, H.; Nielsen, J.D.; Larsen, K.; Jessen, J. CyNC: A method for Real Time Analysis of Systems with Cyclic Data Flows. *J. Embed. Comput.* **2006**, *2*, 347–360.

8. Fidler, M.; Schmitt, J. On the way to a distributed systems calculus: An end-to-end network calculus with data scaling. In Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems—SIGMETRICS '06/Performance '06, Saint Malo, France, 6–10 June 2006; ACM Press: New York, NY, USA, 2006; pp. 287–298. [\[CrossRef\]](#)
9. Promyslov, V. Using the Method of “Network Calculus” To Simulate the Computerized Control System with Non-Uniform Data Flow. *IFAC Proc. Vol.* **2012**, *45*, 645–648. [\[CrossRef\]](#)
10. Bouillard, A.; Thierry, É. An Algorithmic Toolbox for Network Calculus. *Discrete Event Dyn. Syst.* **2008**, *18*, 3–49. [\[CrossRef\]](#)
11. Bouillard, A.; Jouhet, L.; Thierry, E. Comparison of different classes of service curves in network calculus. *IFAC Proc. Vol.* **2010**, *43*, 306–311. [\[CrossRef\]](#)
12. Poletykin, A.; Jharko, E.; Mengazetdinov, N.; Promyslov, V. The new generation of upper levels systems and industry 4.0 conception in NPP APCS. In Proceedings of the 2017 Tenth International Conference Management of Large-Scale System Development (MLSD), Moscow, Russia, 2–4 October 2017; pp. 1–5. [\[CrossRef\]](#)
13. IAEA. *Approaches for Overall Instrumentation and Control Architectures of Nuclear Power Plants*; IAEA Nuclear Energy Series NP-T-2.11 English STI/PUB/1821 978-92-0-102718-4 2018; IAEA: Vienna, Austria, 2018.
14. Cruz, R.L. A Calculus for Network Delay. Part I: Network Elements in Isolation. *IEEE Trans. Inf. Theory* **1991**, *37*, 114–131. [\[CrossRef\]](#)
15. Cruz, R.L. A Calculus for Network Delay. Part II: Network Analysis Information Theory. *IEEE Trans. Inf. Theory* **1991**, *37*, 132–141. [\[CrossRef\]](#)
16. Baccelli, F.; Cohen, G.; Olsder, G.J.; Quadrat, J.-P. *Synchronization and Linearity—An Algebra for Discrete Event Systems*; John Wiley & Sons: New York, NY, USA, 1993; 514p.
17. Promyslov, V.; Semenov, K. The Estimation of Control System Parameters in “Network Calculus” from Experimental Data. In Proceedings of the 2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Sochi, Russia, 17–21 May 2021; pp. 818–822.
18. Ahmadi, H.; Martí, J.R.; Moshref, A. Piecewise linear approximation of generators cost functions using max-affine functions. In Proceedings of the 2013 IEEE Power Energy Society General Meeting, Vancouver, Canada, 21–25 July 2013; pp. 1–5. [\[CrossRef\]](#)
19. Camponogara, E.; Nazari, L.F. Models and Algorithms for Optimal Piecewise-Linear Function Approximation. *Math. Probl. Eng.* **2015**, *2015*, 1–9. [\[CrossRef\]](#)
20. Baybulatov, A.A.; Promyslov, V.G. A Technique for Envelope Regression in Network Calculus. In Proceedings of the 11th IEEE International Conference on Application of Information and Communication Technologies, Moscow, Russia, 20–22 September 2017; pp. 1–4. [\[CrossRef\]](#)
21. Vapnik, V.N. *Statistical Learning Theory*; John Wiley & Sons: New York, NY, USA, 1998; 768p.
22. Baybulatov, A.A.; Promyslov, V.G. Control System Availability Assessment via Maximum Delay Calculation. In Proceedings of the 2019 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Sochi, Russia, 25–29 March 2019; pp. 1–6. [\[CrossRef\]](#)
23. Mohr, D. *Statistical Methods*, 4rd ed.; Academic Press: Cambridge, MA, USA, 2021; 784p.
24. Fidler, M.; Rizk, A. A Guide to the Stochastic Network Calculus. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 92–105. [\[CrossRef\]](#)
25. RFC 793. Transmission Control Protocol. Protocol Specification. 1981. Available online: <https://tools.ietf.org/html/rfc793> (accessed on 28 April 2021).
26. Burnecki, K.; Wylomanska, A.; Chechkin, A. Discriminating between Light- and Heavy-Tailed Distributions with Limit Theorem. *PLoS ONE* **2015**, *10*, e0145604. [\[CrossRef\]](#) [\[PubMed\]](#)
27. ISO/IEC. *ISO/IEC 14756 Information Technology—Measurement and Rating of Performance of Computer-Based Software Systems*; ISO/IEC: Geneva, Switzerland, 1999.
28. Smith, M. *Statistical Analysis Handbook. A Comprehensive Handbook of Statistical Concepts, Techniques and Software Tools*; The Winchelsea Press: Edinburgh, UK, 2018; 638p.
29. Baybulatov, A.A.; Promyslov, V.G. On a Deterministic Approach to Solving Industrial Control System Problems. In Proceedings of the 2020 International Russian Automation Conference (RusAutoCon), Sochi, Russia, 6–12 September 2020; pp. 115–120. [\[CrossRef\]](#)