Andrzej MAJCHER
Institute for Sustainable Technologies – National Research Institute, Radom
andrzej.majcher@itee.radom.pl

ASSESSMENT OF DELAYS IN THE APPLICATION LAYER OF AN EVENT-DRIVEN NETWORKED CONTROL SYSTEM

Key words

Networked control system, event driven, batch control, communication protocol.

Abstract

Problems of networked control systems (NCSs) are associated with inherent time delays. In such systems, particularly those applied to supervise batch processes, tasks with flexible time frames are also performed. However, even in such cases, the knowledge of border delay values is necessary. The article presents a simulation model for an event-driven NCS in which the events are initiated in the main controller and executed in local controllers. The author discusses the results of tests on the efficiency of the model, assessed in the application layer, and the results of verification tests with a real PLC and a Modbus TCP protocol. The structure of the control system and the method for the analysis of its efficiency can be used in the design of process control systems, simulators with virtual objects, and hardware-in-the-loop test systems.

Introduction

A networked control system (NCS) is a kind of a distributed control system in which signals are transmitted in the regulatory control loop through a data transmission network [1, 2]. The structure of an NCS (Fig. 1) is composed of a controller with a data transmission interface and a transmission protocol, actuators through which excitations *U(t)* are fed to the controlled object, sensors that measure output variables *Y(t)* of the object, and elements of a communication network that depend on the kind of the communication protocol used.

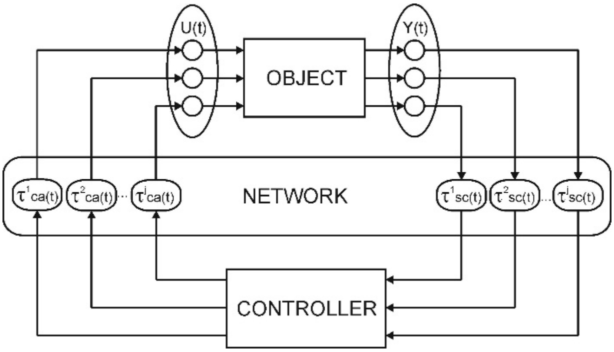


Fig. 1. General structure of an NCS system

The constantly growing areas of the possible application of NCSs result from some essential advantages of the networked architecture, where both sensors and actuators can be “plug and play” systems that do not require the adjustment of their settings and can be easily exchanged in the case of a system failure. The majority of configurations of NCSs reduce the length of wires connecting the controller with sensor and actuator modules, which makes the installation of the system easier, and reduces its costs. The possibility to use the network node redundancy also helps one to achieve better efficiency of the system and resistance to damage. The natural modularity of an NCS improves system diagnostics and management.

One of the most important problems encountered when building NCSs concerns the compensation of the effects of a delay introduced by the network. Network communication protocols lead to the occurrence of delays τ_{ca} and τ_{sc} (Fig.1), which are constant for deterministic protocols (e.g., Profibus, DeviceNet, CanPlus), or stochastic for an Ethernet network. In the case of deterministic protocols, constant delays are understood as the ability to determine the maximum transmission time for a given, specific, and completely defined communication parameters.

The most common computer networking technology for Local Area Networks (LANs), i.e. the Ethernet with the TCP/IP protocol, was not designed for real time systems. However, the great availability of network equipment means that this network is frequently used in real time systems, particularly in applications to monitor the variables of the status of control objects.

Among the NCS structures used, there is an event handling method that is particularly applied for the remote control over robots [3, 4, 5]. This method is based on the reference to the end of a given event not to the real time, which, through changeable delays induced in the network, prevents system destabilization. This method is also advantageous for systems controlling sets of research and test devices, as well as technological processes, where synchronisation of discrete events in real time is an important matter, and where adjustment of single state variables can be executed in local controllers. In batch control systems for technological processes, for example, process devices are controlled through information included in technological recipes that describe individual steps of the process. The recipes are connected with a model of the process, and not with hardware units of the technological stand [6, 7]. The sequence of the steps of the technological recipe performed by given hardware units and in a defined order, leads to the production of an end product. The steps can be treated as real time tasks executed in hierarchical order by the main controller (Fig. 2a). In such a structure, a control object can be represented by a model of a fast dynamic (linear/non-linear) or slow dynamic (linear/non-linear) process.

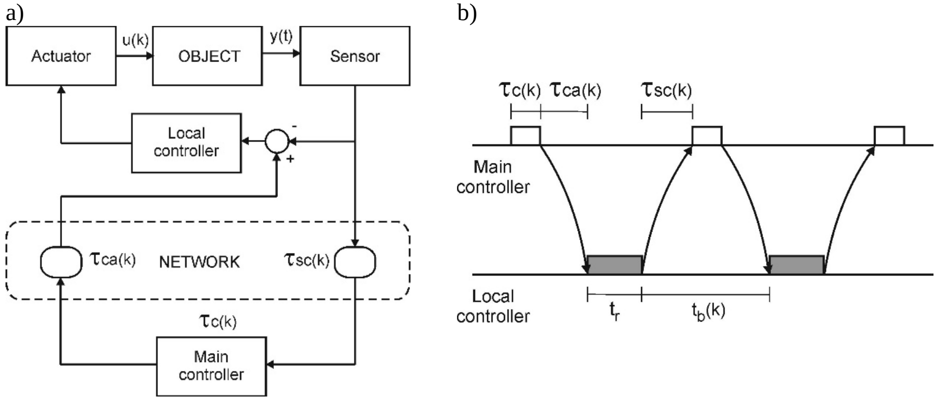


Fig. 2. The structure of a hierarchical NCS [8] (a) and time dependencies of technological recipes in this structure (b)

For a hierarchical structure, it is essential to define a minimum task execution time t_r and a minimum time between the execution of consecutive tasks t_b (Fig. 2b). The values of these times determine the quality of the technological process, and the knowledge of them is necessary when the proper execution of the process requires its consecutive phases to be performed in a quick manner.

Such technological requirements can be met through the application of deterministic communication protocols with well-known delays (e.g., the EtherCAT protocol [9, 10]). However, in this case, the application of solutions with a non-deterministic Ethernet Network, OPC servers, and a PC with the Windows operating system, which are more popular because they are cheaper, are problematic. A proper execution of the batch control depends on the delays in this kind of a system. This is less important in SCADA applications where, most commonly, slowly changing process parameters (e.g., temperature, pressure, level) are monitored, but it becomes crucial for the control over processes in which it is necessary to quickly perform each phase of the process. Event based synchronisation does not completely eliminate the impact of delays. The next event is handled when the controller receives confirmation that the previous one has been finished. The minimum time between sending the event handling commands and the minimum event handling time depends on total delays in information transmission and processing.

In the following sections of the article, the author presents a simulation model for a system with a hierarchical structure with an algorithm for the synchronisation of events introduced by the main controller, and performed in local controllers (Section 2). This is followed by the presentation of the results of delay measurements in a real system employing PLCs with the Modbus/TCP protocol as slave controllers, and the LabView environment for a PC with the MS Windows operating system used as a main controller (Section 3). Table 1 depicts symbols used in the article.

Table 1. Symbols

Notation	Description
τ_c	Delay introduced by the controller in the general model of the NCS
τ_{ca}	Delay on the controller – actuator path in the general model of the NCS
τ_{sc}	Delay on the sensor – controller path in the general model of the NCS
t_r	Event handling (execution of the recipe) time in the local controller
t_b	Time between consecutive events handled by the local controller
τ_{LM}	Delay on the local controller – main controller path in an NCS with a hierarchical structure
τ_{ML}	Delay on the main controller – local controller path in an NCS with a hierarchical structure
t_{rs}	Time between the setting of the task completion request s(t) and the confirmation r(t) as seen by the application of the client
t_{rM}	Event handling time as seen by the main controller
$t_{\Delta M}$	Sampling time of the main controller

1. Simulation model of an NCS

A simulation model of an NCS (Fig. 3a) contains the *n* number of local controllers which have their own data transmission servers, and which are connected through a network with a main controller *(PC)*. The main task of the main controller is, as assumed by the author, to handle events taking place in control objects *(Object 1, 2, …, n)*. The event simulated is the measurement of the time interval with a value set in the main controller. This corresponds to the control procedure for a batch process, where a main controller requests the execution of a given phase of the process in the local controller.

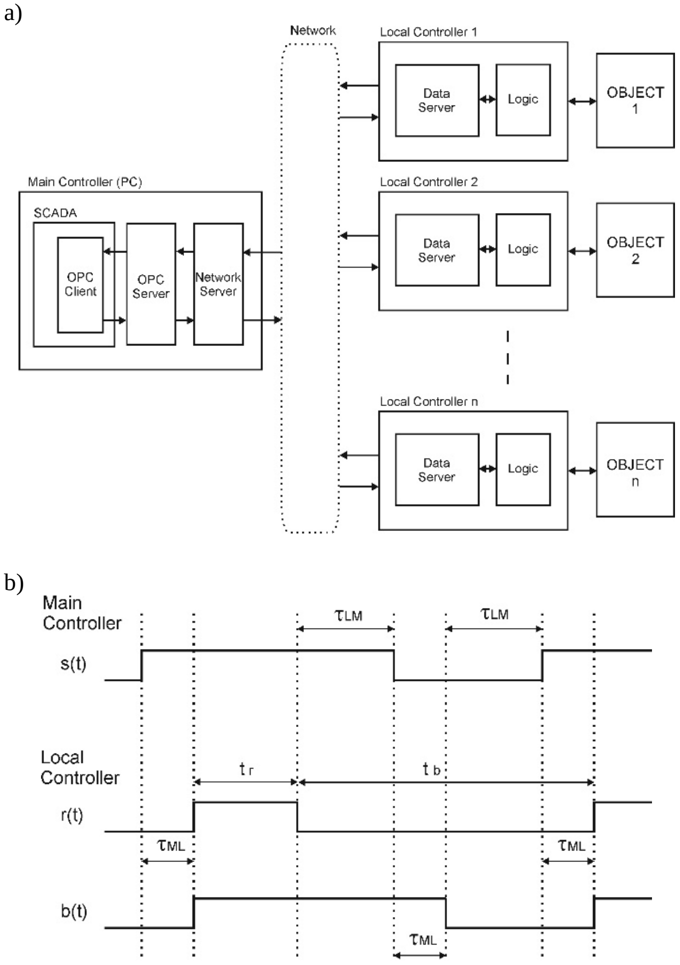


Fig. 3. The structure of a modelled NCS (a) and a time diagram of the events synchronisation method used (b)

A cyclical execution of this task calls for the application of a proper synchronisation method that would ensure the transmission of information about task initiation and completion between the node and the main controller. The author employs the algorithm (Fig. 3b), in which the main controller initiates the execution of the signal setting task *s(t)*, while the local controller confirms the initiation of the signal setting task *r(t)* and the additional signal *b(t)*. Once the task is completed, the signal *r(t)* is reset. This state, together with the state *b(t)* causes resetting *s(t)*. In this algorithm, the main controller performs the following programme:

```
function main_controller( )
if (s==0 and r==0 and b==0) {s=1};
if (s==1 and r==0 and b==1) {s=0};
```

While the function of the local controller is as follows:

```
function local_controller (t)
if (s==1 and r==0 and b==0) {r=1};
wait(t); r=0;
if (s==0) {b=0};
```

Compared to the easiest possible event handling method consisting in the monitoring of the status of the signal *r(t)*, the algorithm presented above is insensitive to delays on the main controller – local controller path, and does not need the identification of signal slopes, only their levels. The use the levels instead of rising or falling edges of a signal for detecting time delays increases the resistance to interference. The *main_controller* programme can be performed with the introduction of timeouts compensating stochastic delays introduced by the remaining elements of the system, which is particularly important in the case of repeated commands to perform the tasks.

The model was developed using the LabView package, and 20 nodes representing local controllers were used. Each node is built as a memory-resident OPC Server with shared variables. The software was implemented on a PC with an i5 2.53 GHz processor, and a 64-byte MS Windows Professional operating system.

The independent module of the NCS's model is an element intended for the analyses of the recording of signal synchronisation for each event handled. The system does not enable direct measurement of τ_{LM} and τ_{ML} delays characteristic for an NCS with a hierarchical structure. These delays are available as time intervals observed from the client and local controller applications. The analysis of process performance through the client application allows the measurement of times t_{rs} and t_{rM}, where the first of them is a time between the signal of the task completion command and its confirmation *r(t)* as seen by the client application (Fig. 4). This time depends on the delay on the main-local controller

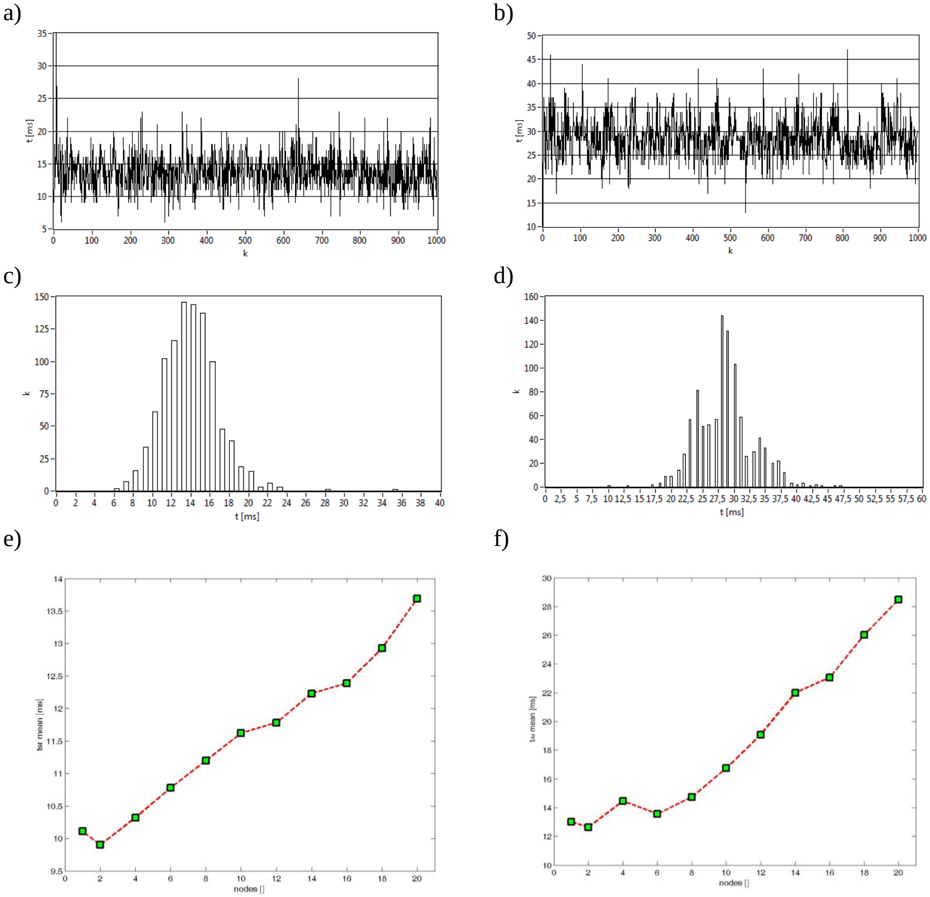


Fig. 4. The results of measurements of times t_{rs} (a,c) and t_{rM} (b,d) for $k = 1000$ task performance commands for each of the 20 network nodes ($t_r = 10$ ms) and graphs presenting mean values of times measured t_{rs} (e) and t_{rM} (f) for a different number of NCS nodes

path (τ_{ML}) and *vice versa* (τ_{LM}), as well as on the stochastic sampling time of the main controller $t_{\Delta M}$:

$$t_{rs} = \tau_{ML} + \tau_{LM} + t_{\Delta M} \tag{1}$$

On the other hand, time t_{rM} is a time of the event t_r as seen by the main controller, expressed in the following manner:

$$t_{rM} = t_r + (\tau_{LM1} - \tau_{LM2}) + t_{\Delta M} \tag{2}$$

where: t_r – the task set time,

τ_{LM1}, τ_{LM2} – value of a random variable τ_{LM} for a given task performed by the local controller,

$t_{\Delta M}$ – delay stemming from the sampling time of the main controller.

The measurements conducted using the system developed enable one to assess time t_b, which, for the algorithm presented in Fig. 3b, can be expressed with the following equation:

$$t_b = 2(\tau_{LM} + \tau_{ML}) \tag{3}$$

The task performance time t_r, on the other hand, does not depend on network delays and it is set by software.

The frequency of signal sampling in the client application is not constant, and it is determined by the operating system used (Windows) that supports its tasks with a different frequency. This prevents the exact measurement of times in the model proposed by the author, and concurrently perfectly represents the specificity of a nondeterministic Ethernet network.

Simulations of the model and the analysis of the resulting times were carried out for different numbers of local controllers performing $k = 1000$ tasks. Some results of the measurements are shown in Figs. 4a-d. The results of simulation tests for different numbers of nodes are presented in Figs. 4e and f. With the increase in the number of nodes in the NCS network, one can notice a significant increase in the values of times t_{rs} and $t_{rM.}$ The remaining parameters of the statistical description of measurement results that were set (e.g., variance, standard deviation, skewness, or kurtosis) do not allow a clear distinction between a proper and disturbed operation of the system.

2. System with real controllers

Virtual OPC servers of a simulation model can be replaced with servers implemented in real controllers.

When a network controller type STP NIP 2212 with an output module STB DDO 3200K (Schneider Electric, the Advantys series) was introduced into the system as a twenty-first local controller, the author noticed that the times t_{rs} and t_{rM} were prolonged (15.28 ms and 38.76 ms, respectively), according to the general tendency depicted in Fig. 4e and f. Network controllers of this type do not allow the introduction of the *function local_controller (t)* procedure, because they are intended to work in a network almost entirely managed by the main controller.

The replacement of network nodes with PLCs enables the configuration of a network with a switched linear topology (Fig. 5). It includes three real nodes, and it can be extended with twenty more nodes. Each node consists of a local controller connected with a local PC, which programmes the controller and locally supports the control object, by means of an Ethernet switch.

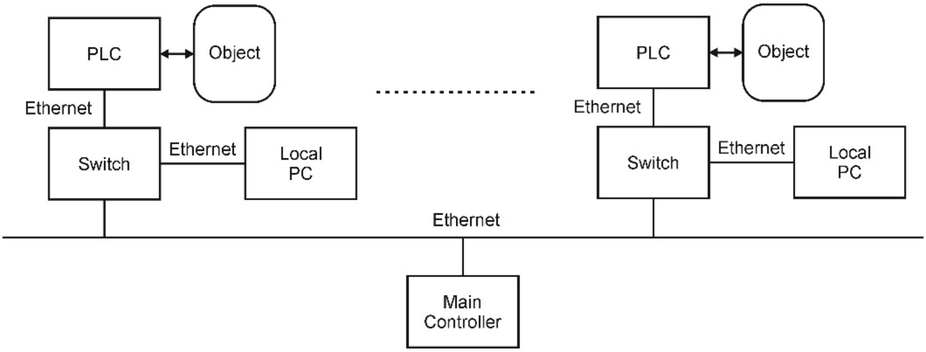
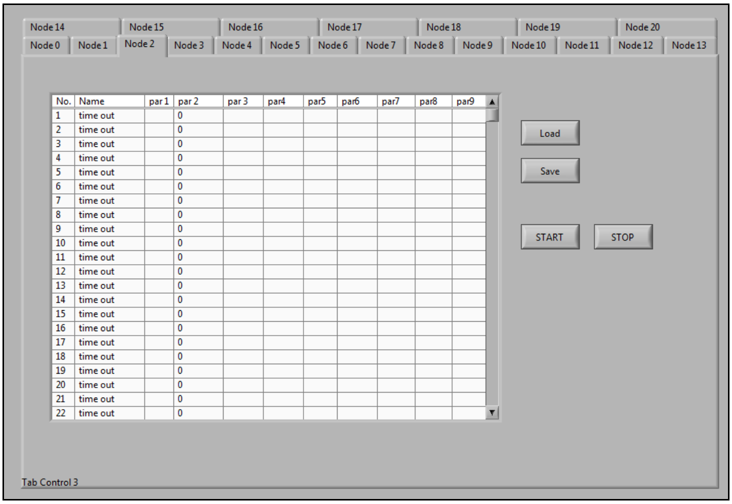


Fig. 5. The structure of the configured model NCS

The nodes are connected with the main PC (the main controller) through the same switch. As PLCs, the author used controllers with the Ethernet interface (750-881 by Wago, and M340 by Schneider Electric).

The application of the main computer uses the LabView package. The libraries of the application servers are organised as separate catalogues in which variables shared for each local node are specified. This makes it possible to switch on and switch off the individual nodes and to replace them with virtual controllers. The task of the main programme, like in the case of a simulation model, consists in the sequential generation of commands performed in the local nodes. The commands are forwarded in form of functions containing a set of ordered parameters introduced using the tables presented in Fig. 6.



No.	Name	par 1	par 2	par 3	par4	par5	par6	par7	par8	par9
1	time out		0							
2	time out		0							
3	time out		0							
4	time out		0							
5	time out		0							
6	time out		0							
7	time out		0							
8	time out		0							
9	time out		0							
10	time out		0							
11	time out		0							
12	time out		0							
13	time out		0							
14	time out		0							
15	time out		0							
16	time out		0							
17	time out		0							
18	time out		0							
19	time out		0							
20	time out		0							
21	time out		0							
22	time out		0							

Fig. 6. Window with commands forwarded to local nodes. The executed command is the timeout with the shortest possible duration

An additional network load is the www server in the node with a 750 881 controller. This server is intended for the visualisation of the state of variables synchronising the work of the controller.

A separate module of the application of the main controller is a measurement loop with which times t_r and t_b can be measured.

The time performance of the system was determined by a 10-fold measurement conducted for the series of 1000 commands to measure a 1 ms time interval (Fig. 7).

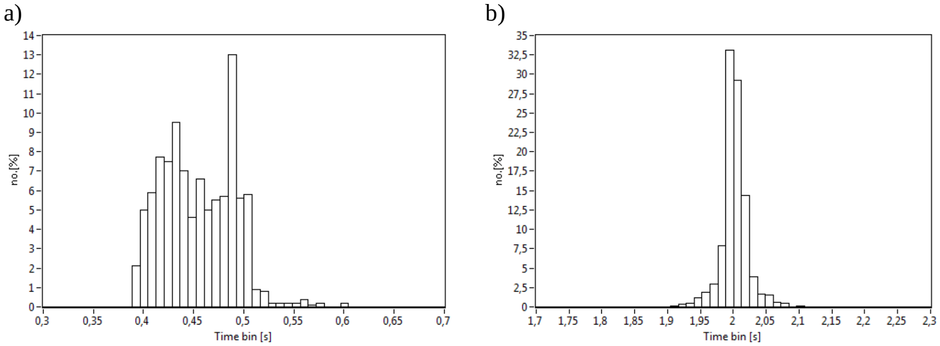


Fig. 7. Sample measurement results for times t_r and t_b for the NCS with one local node – a) 750 type controller, b) time t_r histogram, b) time t_b histogram

The obtained results (Table 1) show the similar time performance of the OPC servers of the local controller and indicate a relatively low elongation measured time at the concurrent increase in the network load.

Table 1. Juxtaposition of the obtained times of information exchange in a hierarchical NCS

Lp.	Configuration of NCS	Time t_r [s]			Time t_b [s]		
		Mean	Maximum	Dominant	Mean	Maximum	Dominant
1	1 node – 750	0.46	0.6	0.49	2	2.09	2
2	1 node – M340	0.39	0.5	0.39	2	2.11	2
3	2 nodes – M340	0.49	0.5	0.49	2.5	3.05	2.02
4	20 nodes, node 1- 750	0.49	0.6	0.39	2	2.16	2.02
5	20 nodes, node 2 – M340	0.39	0.525	0.39	2	2.16	1.99
6	20 nodes, node 3 – M340	0.32	0.36	0.39	2	2.17	2.02
7	20 nodes nodes 1, 2, 3 750 and 2x M340	0.52	0.6	0.49	2.5	3.15	2.02
8	20 nodes, node 1 – 750 with a www server	0.49	0.6	0.49	2	2.16	2.02

The apparent dependency of the obtained times on the structure of the main application, particularly on the solutions concerning the graphic presentation of the system operation, results in the need for tuning the system, both in terms of the parameters of the graphics used and the time constants occurring in the procedures to change the status of the signal of information exchange synchronisation with local nodes.

No significant delays were observed for a system with an increased number of nodes, and the time extensions do not have a negative impact when the system is used for the control over technological processes. Decisive importance should be attributed to the main controller. The sources of the delays introduced in this software are the following: the OPC server, which acts as a memory-resident programme, the concurrent software tasks, and other programmes supported by the operating system.

Summary

The results obtained are crucial for the design of NCSs in which tasks executed in the application of the main controller need to be performed within a defined time frame. The values of delays depend on the structure of a given application, the OPC servers, the operating system, and the efficiency of the PC. These factors have a dominant role, in comparison with the delays introduced by the network itself. Times measured in the application layer are in the order of individual seconds, while delays in the network with the same transmission protocol and the number of nodes are in the order of microseconds [9].

For a SCADA application, it is possible to perform tasks with loose time constraints, like batch control [12, 13], on-line analysis of process data, or optimisation tasks. The simulation model enables one to answer the question of how often the set values can be changed, and with what maximum frequency the signals can be read in the main computer.

An important advantage of the method described is the modularity of the virtual local nodes. They imitate the behaviour of real nodes and can be used for the design of the hardware-in-the-loop structures [14]. This solution consists in the replacement, in the system simulation programme, of a given block with a real system (e.g., regulator) supporting the real control object. The module of the virtual local node can replace real hardware units, and it can be used at the time of the design, initiation, and diagnostics of complex machine and process control systems. The system can also be used in the diagnostics and maintenance of technological stands, e.g., by maintaining the functionality of the stand in the event of the failure of the existing local controller of the device.

Acknowledgment

Scientific work executed within the Strategic Programme “Innovative Systems of Technical Support for Sustainable Development of Economy” within Innovative Economy Operational Programme.

References

1. Nilsson J.: Real-time control systems with delays. Department of Automatic Control, Lund Institute of Technology, 1988.
2. Franzè G., Tedesco F., Famularo D.: Model predictive control for constrained networked systems subject to data losses. Automatica 54, 2015, pp. 272–278.
3. Tipsuwan Y., Chow M.-Y (2003): Control methodologies in networked control systems. Control Engineering Practice 11, 2003, pp. 1099–1111.
4. Li H., Kawashima K.: Experimental comparison of backdrivability for time-delayed telerobotics. Control EngineeringPractice 28, 2014, pp. 90–96.
5. Zhang J., Lina Y., Shi P.: Output tracking control of networked control systems via delay compensation controllers. Automatica 57, 2015, pp. 85–92.
6. Godena, G.:Proc-Graph: a procedure – oriented graphical notation for process-control software specification. Control Engineering Practice. 12, 2004, p. 99–111.
7. C. de Prada, Grossmann I., Sarabia D., Cristea S.: A strategy for predictive control of a mixed continuous batch process. Journal of Process Control 19, 2009, pp. 123–137.
8. Dong-Sung Kim, Dong-Hyuk Choi, Prasant Mohapatra: Real-time scheduling method for networked discrete control systems. Control Engineering Practice 17, 2009, p 564–570.
9. Robert J., Georges J-P.,Rondeau E., Divox T.: Minimum cycle time analysis of Ethernet-based real-time protocols. Int. J. Comput. Commun. Vol. 7 (2012), No. 4, pp. 743–757.
10. Sung M., KimI., KimT.: Toward a holistic delay analysis of EtherCAT synchronized control processes. Int. J. Comput. Commun., Vol. 8 (2013), No. 4, pp. 608–621.
11. Majcher A., Wiejak J., Przybylski J., Chudoba T., Wojnarowicz J.: A novel reactor for microwave hydrothermal scale-up nanopowder synthesis. International Journal of Chemical Reactor Engineering. Vol. 11, Issue 1, 2013, pp. 1–8.
12. Majcher A.: Automation of the process of microwave hydrothermal synthesis of nanopowders. Pomiary Automatyka Robotyka PAR, 2/2013, p. 208–212.
13. Majcher A.: Model of the event driver networked control system for the diagnostics use. Maintenance Problems 2 (2011), pp. 131–140.
14. Bacic M., Neild S., Gawthrop P.: Introduction to the special issue on hardware-in-the-loop simulation. Mechatronics, 19 (7) (2009), pp. 1041–1042.

Ocena opóźnień w warstwie aplikacji systemu sterowania sieciowego z obsługą zdarzeń

Słowa kluczowe

System sterowania sieciowego, sterowanie zdarzeniami, sterowanie wsadowe, protokół komunikacyjny.

Streszczenie

Problemy systemów sterownia sieciowego (NCS) związane są z inherentnymi dla nich opóźnieniami czasowymi. W systemach takich, szczególnie w sterowaniu procesami wsadowymi, wykonywane są także zadania o luźnych ograniczeniach czasowych, ale wymagające dokładnej znajomości granicznych wartości opóźnień. W artykule przedstawiono model symulacyjny sytemu NCS z obsługą zdarzeń wyzwalanych w sterowniku głównym i wykonywanych w sterownikach lokalnych. Przedstawiono wyniki badań wydajności modelu ocenianej w warstwie aplikacji oraz badań weryfikacyjnych z rzeczywistym sterownikami PLC i protokołem Modbus TCP. Przestawiona struktura systemu i metoda analizy jego wydajności może znaleźć zastosowanie w projektowaniu systemów sterowania procesami technologicznymi oraz projektowaniu symulatorów z wirtualnymi obiektami sterowania i rozwiązań typu hardware-in-the-loop.