

Multilevel Control under Communication Constraints

Wojciech Grega, *Member, IEEE*

Abstract—The performance of distributed control systems, apart from the sampling period, depends on many parameters, such as the control loop execution time, jitter and communication parameters of data transmission channels. Limited throughput of transmission channels, combined with non-optimized hardware and software components introduce non-determinism in the real-time control system. Additionally, some control loops can be handled not only by local, device – level controllers, but also by the supervisory controllers in a multilevel, vertical control hierarchy. In this paper we propose a design method for a networked, multilevel control system, which can be used to select proper values of the design parameters. In the paper some control techniques improving the temporal robustness of networked control systems are analysed.

I. INTRODUCTION

Closed loop control over industrial communication networks has gained increasing attention in recent years due to the progress in communication technologies. Evolution moves to Industrial Ethernet networks, replacing very quickly the proprietary networks [1], [2].

The advantages of data transmission channels integration into control system are obvious, such as reducing wiring costs and increasing flexibility. Thanks to these important benefits, typical applications of these systems range over various fields, such as automotive, mobile robotics, advanced aircraft, and so on. However, introduction of communication networks in the control loops makes the analysis and synthesis of distributed control systems more complex. Digital control systems are based on periodic operations and real-time assumptions. The introduction of data transmission networks into the feedback loop in many cases violates conventional control theories assumptions such as non-delayed or evenly spaced sampling sensing and actuation.

The reason is that a computer network is characterized by its maximal throughput. This parameter limits the amount of data that can be sent within a time unit. Network-induced delays may vary depending on the network load and medium access protocol. Generally, networked control often introduces some additional dynamics and temporal non-determinism. For distributed control systems variable queuing delays, transmission delays and the loss of data leads to the deterioration of the quality of control [3], [4].

Manuscript received February 12, 2010.

Author is with Department of Automatics, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków (wgr@ia.agh.edu.pl). This work was supported by the European Regional Development Fund Grant no. UDA-POIG.01.03.01-12-171/08/00.

Control theory specialists and control engineers do not care very much about real-time, distributed implementations of control algorithms. In many cases they do not understand control-timing constraints. The typical proposed solutions are: “buy a faster computer” or “install more efficient data transmission network”. Control theory does not advise on how to design controllers to take limitations of the communication network into account. In some cases ones try to separate real-time aspects and dynamics of the control system. Designed controllers guarantee all task deadlines under the worst case of the controller load.

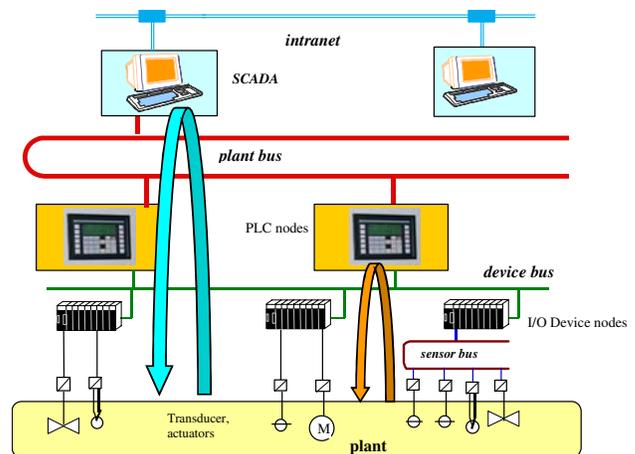


Fig. 1. Multilevel structure of the industrial control system.

Most of the industrial control systems adopts multilevel, vertical control hierarchy [6], [7]. Logically, the system is structured into three levels (Fig. 1), which are: the direct (device) control level, supervisory level and management level.

Basic task of the direct (device) control level is to maintain the process states at the prescribed set values. Device controller level provides interface to the hardware, either as separate modules or as microprocessors incorporated in the equipment to be controlled. Here, mainly PID digital control algorithms are implemented – in some cases more advanced control methods as multivariable control or adaptive functions. A number of embedded control nodes and Programmable Logical Controllers (PLC) are used as the front-ends to take the control tasks. High speed networks and fieldbuses are implemented at the direct control level to exchange in real time the information between front-ends and the device

controllers and, vertically, with the supervisory control level. This architecture has the advantage of locating the hard real-time activities as near as possible to the equipment.

The supervisory level comprises workstations and industrial PCs providing the high-level control support, database support, graphic man-machine interface, network management and general computing resources.

Classically, the supervisory level calculates set points for controllers according to the defined criterion. For this purpose more complex mathematical models of the process are employed at this level to find optimal steady-state, by solving optimisation and identification tasks [6]. Due to the rapid development of computer technology, there is growing scope of more advanced close-loop algorithms (predictive control, repetitive control) located at this level. However, increasing computational efficiency of PLCs at the device level supported by high performance networks transferring data and control signals vertically gives more flexibility to the designer. The control loops can be handled by local, device – level controllers, but also by the supervisory controllers (Fig.1). For example, predictive control algorithm can be handled by supervisory workstation as well as by a local PLC [8]. In some cases similar control algorithms must be located in both of the levels if redundancy of control system is required. It should be noted, that upper level loops usually offer shorter computational time due to the higher efficiency of the workstations.

Role of the communication networks at each level is to ensure data transmission and coordinating manipulation among spatially distributed control nodes. Evolution of industrial communication has moved to Industrial Ethernet networks [1]. Since Ethernet is a shared network, the packets containing the digitized measurements need to share the network bandwidth with external traffic; thus, the available channel capacity is limited and dynamically changing. Therefore, the control over the upper level loop usually offers longer data transfer time due to high vertical network traffic and longer delays.

Digital control theory normally assumes evenly spaced sampling intervals and constant control delay between sampling and actuation. However, this can seldom be practically achieved in a real resource-constrained system. Relations between control task timing and control systems properties have been described by several authors ([9], [10]). It is stressed, that care must be taken when real-time execution of control algorithms generates sampling – actuation jitters or other kinds of run time violation of the closed-loop timing assumptions.

It has been stated in a previous work [5], that integrated approaches combining two disciplines: real-time computation and digital control systems, results in better quality of control systems. This problem is continued in this paper for multilevel control architecture. We address the questions about selection of the application platforms and closed-loop execution times in such a way that

process dynamics and communication network properties are balanced.

The paper consists of six sections. In Section II we describe the model of distributed control system. In Section III we define design parameters and we demonstrate the construction of design chart for multilevel, networked control systems. In Section IV we discuss the problem how to improve the temporal robustness of distributed control systems and how to compensate unwanted dynamics introduced by data transmission channels. In this section we also demonstrate, how the design chart can be used for selection of design parameters. Section V presents experimental validation of this approach, and Section VI concludes.

II. CODESIGN OF DISTRIBUTED AND REAL-TIME CONTROL SYSTEMS

The basic block diagram of the distributed control system is shown in Fig.2. The process outputs are measured and control signals are applied through the distance I/O devices. The I/O devices are integrated with A/D and D/A converters.

The communication to and from the controller node is made over a network. From the point of view of digital control theory, it is natural to sample the process with an equal period T_0 and to keep the control delay as short as possible. This suggests that the sensors and actuators are time-triggered (sampling period T_0), while the controller is event-triggered, which means that is triggered by the arrival of the new data.

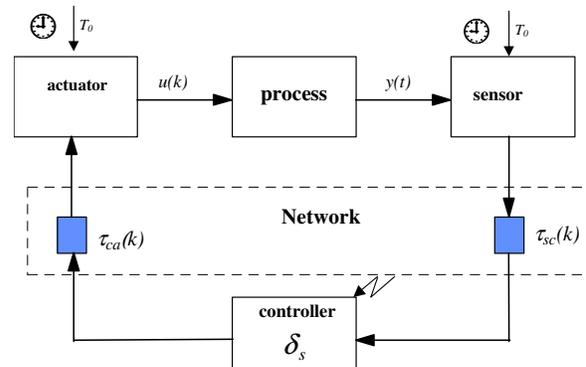


Fig.2. Basic model of distributed control system.

In the previous paper [11] we claim that in the computer implementation of distributed control systems, real-time algorithms, data transmission models and digital control theory methods cannot be developed separately because an unexpected control system performance may occur. We had shown that three parameters need particular attention from the distributed control design perspective: sampling and actuation tasks period, controller task period and network parameters: latency and jitter. Due to the close relationships between the network and control parameters the selection of the best sampling period will be a compromise. In this section we will demonstrate the construction of a networked control design chart, which can be used to select proper design parameters.

A. Sampling and Actuation Tasks

We will assume, that the control algorithm design is based on correctly identified model of the process and the model of disturbances (referred to as “nominal models”). We assume that it is possible for the nominal models to estimate a maximal, admissible sampling period, which would guarantee acceptable control performance.

One accepted rule is [12] that the control task period should be a ($a > 1, a \in \mathbb{N}$) times smaller than the period of the cut-off frequency, approximated in some reasonable way for the nominal process model. This upper bound of T_0 is denoted as T_0^u (Fig.3).

For the design purpose we assume that performance of the closed-loop control system is a strictly monotonic function of T_0 : any sampling (actuation) period $T_0 < T_0^u$ improves the control performance. For $T_0 < T_0^l$ improvement is not observed. Finally, the sampling (actuation) task period can be estimated as $T_0 \in [T_0^l, T_0^u]$.

B. Controller Task Period

The applied control platforms (processor, peripherals hardware and operating systems) are characterized by a closed-loop execution time, estimated as $\delta_s \in [\delta_s^l, \delta_s^u]$, where δ_s^l - is the lower bound of the execution time for simple control algorithms, δ_s^u - is the execution time of complex control algorithms.

The control algorithm is classified as “simple“, if pseudocode of the controller task includes no more than 5-10 operations (loops are excluded). Examples of “simple“ algorithms are: incremental PID or state feedback controller. If the pseudocode of the controller includes more than 10 operations or loops are included then the algorithm is classified as “complex“.

C. Network Parameters

Presence of networks introduces communication delays and limits the amount of data that can be transferred between nodes. In some cases not all samples from sensor or to actuator (produced with period T_0) can be sent, because the network requires intervals longer than T_0 between the transfers of two consecutive packets. Therefore, constraints on the process data availability, introduced by the communication channel are defined.

The average communication delay between the sensor node and the controller node is denoted as τ^{sc} , τ^{ca} is average communication delay between the controller node and the actuator node, $\Delta(k)$ represents a total jitter in the feedback loop, k - is the number of the control step.

Actually, the communication delays and jitters can be added to the controller execution time creating an estimation of delays and uncertainty in the control loop. The total delay in the control loop is

$$\tau(k) = \tau_{sc} + \tau_{ca} + \delta_s + \Delta(k).$$

It will also be assumed that the jitter is bounded by $0 \leq \Delta(k) \leq \Delta^u$.

D. Codesign

In the previous section we have introduced a number of parameters that need special attention from the perspective of real-time digital control: T_0 - sampling period defining the temporal granularity related to the process dynamics, δ_s - execution time describing the efficiency of the hardware and software application platform and $\tau_{sc}, \tau_{ca}, \Delta$ - communication delays and jitter. Now, we will demonstrate, how these parameters interacts one to another, how to select the application platforms and how to set closed-loop execution times in such a way, that process dynamics and communication network properties are balanced.

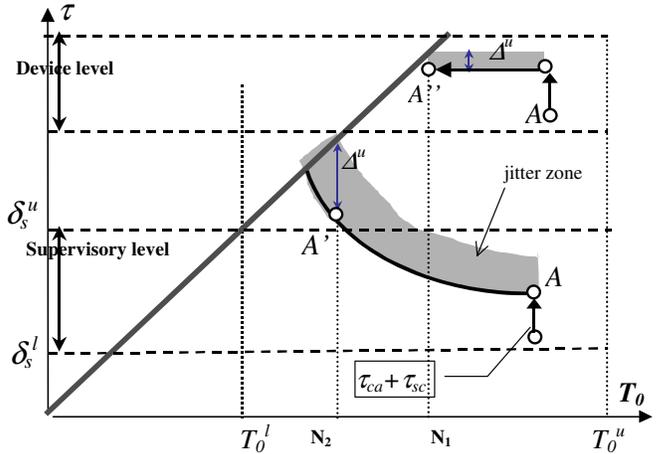


Fig.3. Distributed control system design chart: comparing location of the operating points of system at *Device level* and at the *Supervisory level*.

The operating point of the distributed control system should be located in the area between T_0^l and T_0^u in Fig.3, both for device level and for supervisory level applications. The operating must lie below the line separating “time critical“ solution, which simply means that control loop execution time must be less than sampling period. Points A, A' in Fig. 3 also represent a situation where the design is robust against possible variations (jitter) of the task execution and data transfer times (shaded area in Fig.3).

Let us assume, that Ethernet network is implemented at supervisory level. Computational delay of the controller δ_s is fixed, but for Ethernet network the transmission time delay increases linearly with increasing load - in same case exponentially, when the load on the network exceeds 35 - 40% [13].

It means, that a faster sampling rate for guaranteeing better control performance will saturate the network traffic load, and eventually increase the data transmission time. For the example given in Fig. 3, the best operating point at supervisory level is A' and is constrained by the process

data availability introduced by transmission time delays of the communication channel.

The constraint of this kind is not active on the device level, if communication can be supported by high-speed real-time – network, e.g. ProfiNet, Class 2 [14]. However, at this level another constraint becomes active and critical.

Control loop execution time at the device level can not be longer than the sampling period (A'' in Fig.3), including the jitter $\Delta(k)$. The reason is that cycle of the control loop do not accept intervals between transfers of the two consecutive packets shorter, than N_1 . The time diagram for this situation is given in Fig. 4. For the model from Fig.4 we must assume that

$$\tau_{sc} + \delta_s + \tau_{ca} + \Delta(k) \leq T_0 = N_1 \cdot T_s$$

It means, that the operating point (A'') must be located below the line separating “time-critical” zone, including the jitter zone (Fig.3).

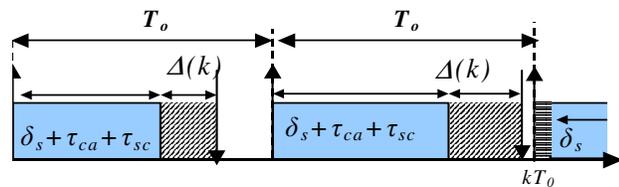


Fig.4. Timing model that can be used for a regularly sampled process.

III. ROBUST DISTRIBUTED CONTROL

For satisfying both control and communication performance a design methodology can be proposed, to apply the proper control signals and to save communication bandwidth.

We wish to design a control algorithm operating with sampling/actuation period as closed to T_0^l as possible. Let us assume again, that Ethernet network is implemented at the supervisory level and real-time network (e.g. ProfiNet) at the device level.

As shown in Fig.3 control system sharing a common data transmission bus at the supervisory level is not able to send all samples produced with period T_0^l from the controller to the actuator. The reason is that the network do not accept intervals between the transfers of two consecutive packets shorter than N_2

The idea now, is to increase network utilization by modification of the transmission pattern – for example by samples grouping. Let us suppose that one sample from sensor is two-byte long and that the remaining data in single a datagram occupy 48 bytes (including the network overhead). If we send four samples in four separate packets then $4 \cdot (2+48)=200$ bytes are used. With sample grouping algorithm we utilize only $48+4 \cdot 2=65$ bytes – over three times less. In this case, sampling and actuation “observed” by the actuator or controller can be lower than N_2 (Fig.3).

All the samples from sensor are transferred through network, however they are grouped together into a M -element packages ($M > 1$) before they enter the network.

This solution is illustrated in Figure 5.

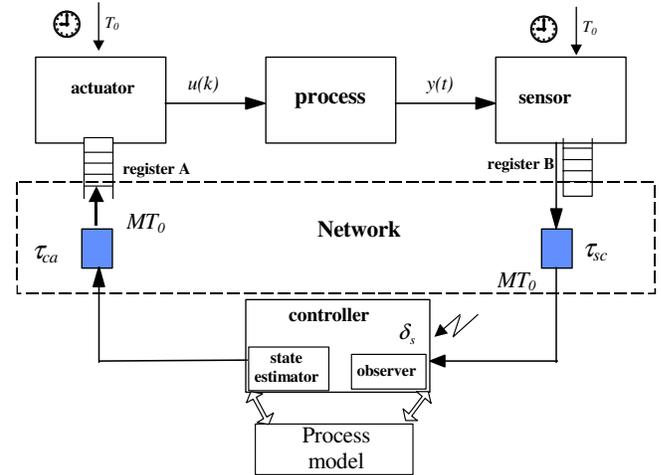


Fig. 5. Grouping of samples: control loop configuration

A new package is sent, after gathering up M elements. The collecting task is done by shift register B. Having received such a package controller calculates new package of M consecutive control values and sends it to register A, which passes them on one by one in the proper order and with the correct timing to actuator. In this case effective sampling periods of particular nodes are diverse. The sensor and actuator are triggered with frequency M times greater than controller. On the other hand, the controller obtains complete (although not punctual) information from the sensor and the actuator receives different (in general) control values for each sampling period T_0 .

Sample grouping effects can be compensated by an approximate model of the process (“observer” in Fig.5) at the controller side, for some range of the sampling period and modeling errors [15]. Extension of the controller by observer (input from sensors) and by control signal estimator (output to actuators) - as proposed in paper [15] - will increase execution time of the algorithm (it will be a “complex” algorithm, according to the previous classification). But this is acceptable at the supervisory level – workstations or industrial PCs’ are typically more efficient than PLCs’ operating at device level. Modified design chart is presented in Fig.6. The sampling period can be set as N_2 .

At device level the desired operating point can not be reached due to the limited control loop execution time (the line separating “time critical” solution in Fig.3). In this case, the actuator is not only dependent on the network; it is also heavily dependent on the controller to compute a new control update at required frequency.

The idea is to reduce temporal dependency of the individual components of the model for Fig. 2 by introducing buffers at the actuator [11]. Buffering can be easily implemented using PLCs’ or embedded controller at the device level. In digital control this operation can be handled by use of a zero-order holds on the control signal.

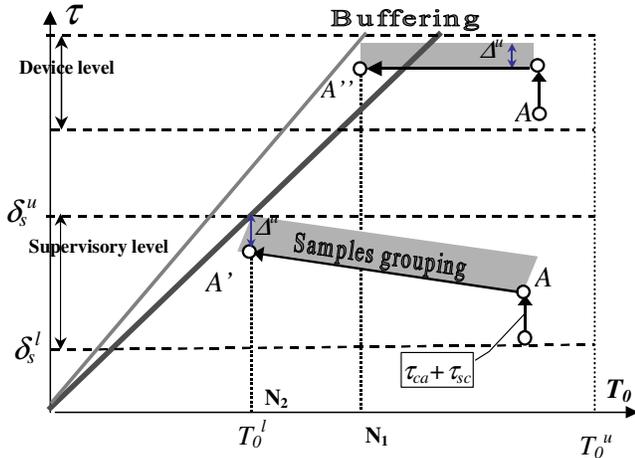


Fig.6. Robust control system design chart: comparing location of the operating point of system at *Device level* and at the *Supervisory level*.

The data package is delivered as soon as possible, but is hold in the buffer and is implemented to the process in the next sampling intervals. By this way synchronisation of the control loop is achieved. The synchronisation condition for p -step buffer is

$$\tau_{sc} + \delta_s + \tau_{sc} + \Delta(k) \leq pT_0.$$

Time diagram of buffering at device level, for $p=2$ is given in Fig. 7.

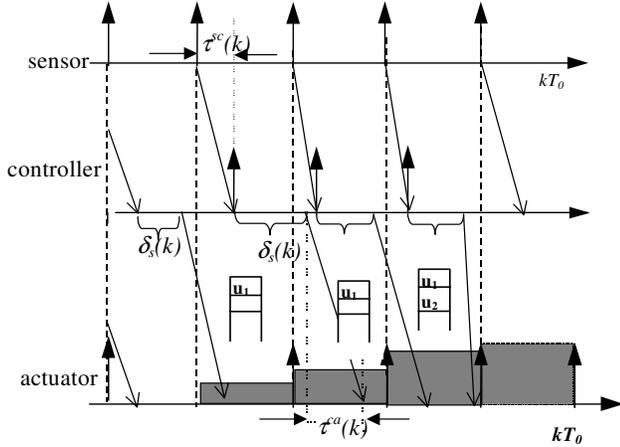


Fig.7. Time diagram of buffering at the *Device level* for $p=2$.

Referring to the design chart (Fig.6), after introducing the buffer the line separating time critical solution can be moved to the left. The sampling period can be set as N_1 .

IV. EXAMPLE: DISTRIBUTED CONTROL OF DC MOTOR

The process under consideration is a DC motor with a 2nd order transfer function

$$H(s) = \frac{K}{s^2 + a_1s + a}$$

with $K = 149,25$ rad/s.volt and two poles corresponding to time $\tau_1 = 502ms$ $\tau_2 = 1.4ms$, respectively. Using the criteria of at least 10 samples in the rise time results in a sampling period $T_0 = 30$ ms.

A discrete PI controller

$$C(z) = k_p \left(1 + k_i \frac{z}{z-1}\right)$$

has been implemented. The parameters k_p and k_i has been tuned to minimize the quadratic cost function (J) evaluated for assumed sampling period.

This control system was simulated for distributed architecture, as shown in Fig.1. The only difference was that only data from controller to the actuator were sent across the network. The connection between sensors and controller was modeled as direct digital link.

For simulation experiments the *True-Time* toolbox has been applied [16]. *TrueTime* is a *Matlab/Simulink*-based simulator for real-time co-simulation of controller task execution in real-time kernels, network transmissions and continuous process dynamics. Using *TrueTime* Network block the Ethernet data transmission link was simulated with the uniform distribution of delays and bandwidth share of 30%. Example results of the latency/jitter histogram are given in Fig. 8.

After number of simulation experiments it was found that improvement of the control quality is not observed for the reference model, if T_0 is less than 5 ms (T_0^l in Fig. 6). For this sampling period the following parameters of the controller were calculated: $k_p = 46 * 10^{-3}$, $k_i = 53 * 10^{-3}$.

For distributed control it was concluded that it is possible to stabilize the system for $T_0 \geq 10ms$, but the best control quality was achieved for $T_0=25-30ms$ ($N_1= 25ms$ in Fig. 6). The reason is, that shorter sampling rates guarantee better control performance, but increasing number of data produces network traffic and saturates the network (Fig. 10).

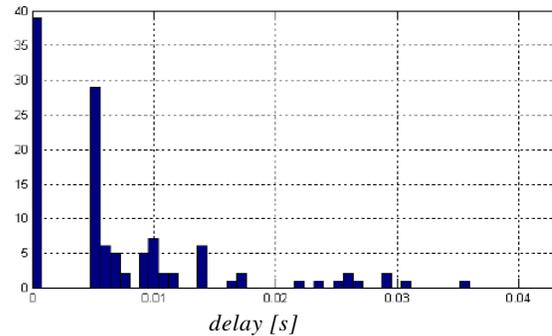


Fig. 8. Distributed control of DC motor: variable network-induced delays.

Next, the samples grouping algorithm was applied for this model. The time diagram of the grouping control algorithm for the assumed package length $M=4$ is shown in Fig. 9. In the equations shown on the in the flow diagram the matrixes Φ , Γ and C were replaced by $\hat{\Phi}$, $\hat{\Gamma}$ and \hat{C} . This reflects the fact that, in practice, the exact model of the plant is unknown and instead, uncertain estimates must be used. Such influence of this uncertainty has been also the subject of the analysis [17].

Example results of the square wave tracking experiments are summarized in Fig. 10. Without latency compensation stability of the control system was lost for $T_0 < 10ms$.

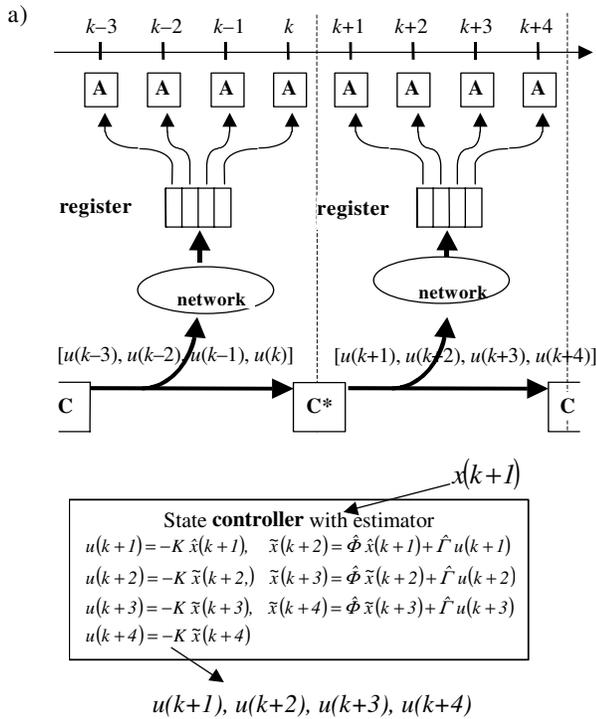


Fig. 9. Control algorithm for system with the grouping of samples (example for $M=4$).

After introducing samples grouping algorithm the control quality (J) was restored for $T_0=5ms$. Dotted line shows the system output for distributed control, continuous - for reference model. It may be concluded that the introduction of samples grouping improves network utilization: it was possible to remove the operating point to $T_0^l = 5ms$, while the control quality was not degraded by network traffic.

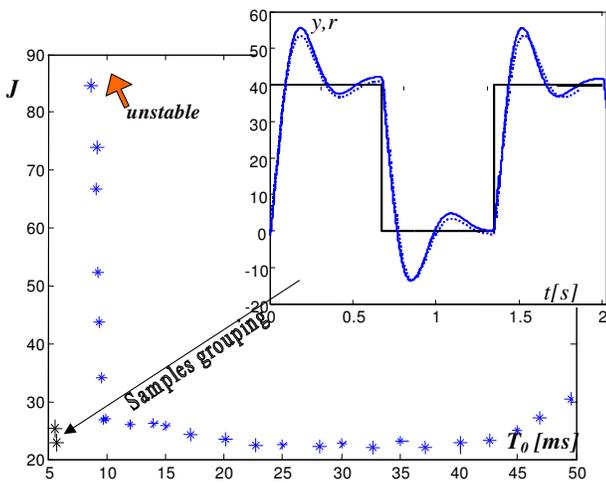


Fig. 10. Distributed control of the DC motor: effect of grouping of samples.

V. CONCLUSIONS

The introduction of networks, limited throughput of data transmission channels, combined with non-optimised hardware and software components introduce non-determinism in the real-time control system. For multilevel systems this problem becomes even more complex. Some

control loops can be handle by local, device – level controllers, but also by the supervisory controllers.

This paper poses a problem of selection of the application platforms, sampling periods and closed-loop execution times in such a way that process dynamics and communication network properties are balanced. The design parameters were defined and the design methodology for networked, multilevel control systems was demonstrated.

Two algorithms improving the temporal robustness of the distributed control system has been analyzed in this paper. A DC motor control example have shown that samples grouping algorithm improves network utilization and increases robustness of the control loop. The general conclusion is that in the computer implementation of distributed control systems, real-time aspects, data transmission models and digital control theory methods can not be developed separately because an unexpected control system performance degradation may occur.

VI. REFERENCES

- [1] M. Felsner, "Real-Time Ethernet – Industry Prospective", in *Proceedings of the IEEE*, vol. 93, pp. 1118- 1129, June 2005.
- [2] D. Jansen and H. Buttner, "Real-time Ethernet the EtherCAT Solution", *Computing & Control Engineering Journal*, vol. 15, pp. 16 – 21, Jan. 2006.
- [3] W. Zhang, M. Branicky and S. Philips, "Stability of Networked Control Systems", *IEEE Control System Magazine*, vol. 21, pp. 84-99, 2001.
- [4] L. Almeida (2008). *Networks for Embedded Control Systems, ARTIST2 China Summer School*, Shanghai. Available: <http://www.artist-embedded.org/docs/Events/08/China>.
- [5] W. Grega, "Design of Distributed Control Systems: from Theoretical to Applied Timing", in: *Recent advances in control and automation*, Warszawa, pp. 343–352, 2008.
- [6] W. Grega, W. Byrski and J. Duda (2009). *InStePro – Integrated Production Control*. Available: <http://www.InStePro.agh.edu.pl>.
- [7] P. Tatjewski, *Advanced Control of Industrial Processes*. Springer-Verlag, London, 2007.
- [8] S.J. Qin, and T.A. Badgwell, "A survey of industrial model predictive control technology", *Control Eng. Practice*, vol. 11, pp. 33-764, 2003.
- [9] K-E. Arzen, A. Cervin, J. Eker and L. Sha, "An Introduction to Control and Scheduling Co-design", in: *Proc. of 39th IEEE CDC*, Sydney, 2000.
- [10] F.-L. Lian, J.K. Yook, D.M. Tilbury, and J.R. Moyne, Network architecture and communication modules for guaranteeing acceptable control and communication performance for networked multi-agent systems, *IEEE Transactions on Industrial Informatics*, vol. 2, 12-24, Feb 2006.
- [11] W. Grega, "Codesign of distributed and real-time control systems", in: *14th International Conference on Methods and Models in Automation and Robotics*, Międzyzdroje, August 2009, pp. 85-88.
- [12] K.J. Aström and B. Wittenmark, *Computer Controlled Systems*. Prentice Hall, London, 1997.
- [13] R. Zurawski (ed.), *The Industrial Communication Systems Handbook*. CRC Press, 2005.
- [14] A. Amiguet, L. Etienne and A. Jaggi (2008). "Performance analysis of PROFINET networks". Available: http://lamspeople.epfl.ch/decotignie/Midterm08_profinet_slides.pdf
- [15] W. Grega and A. Tutaj, "Network traffic reduction by sample grouping for distributed control systems", in: *Proceedings of 3rd Int. Workshop on Networked Control Systems*, (Réseau de Formation des Chercheurs en Automatique et Productique Ed), pp. 1-8, Nancy 2007.
- [16] M. Ohlin, D. Henriksson and A. Cervin, *TrueTime 1.5—Reference Manual*. Lund University, 2007, Sweden.
- [17] L. A. Montestruque and P.J. Antsaklis, "On the model-based control of networked systems", *Automatica*, vol. 39, pp. 1837-1843, 2003.