

Architectural approach to cope with network-induced problems in network control systems design

Ivan Popović*, Aleksandar Rakić**

In the field of process control engineering, network-based systems enable extensive, flexible and scalable applications in industrial automation and control. However, network-induced problems are influencing the stability and performance and they are introducing constraints in the system design and operation. While most of the existing design methodologies are searching for the specific solution within the domain of the control theory, we propose the comprehensive architectural approach that addresses wide range of the network-related issues and copes with them in the effective way. Presented solution combines several architectural styles encapsulating the actuating, sensing and control functionality into the unified service-oriented components, while the data transport is supported through event-triggered distributed middleware components. Given architectural approach decouples the design of process control functionality from the properties of the control network infrastructure. The effectiveness of the proposed solution is verified through the analysis of the system operation in the given case-study.

Key words: networked control systems, communication constraints, service-oriented architecture, predictive control, time-awareness

1 Introduction

The recent technology advances in the computer networking and communications enable the integration of various end-devices, objects, applications and services in complex distributed systems. The common approach for the integration of distributed system components assumes the loosely coupled interconnection, based on the appropriate level of abstraction at the different layers of the system architecture [1, 2]. Networked control systems (NCS) represent a class of distributed systems, where the process control loops are closed through the communication network [3]. The concept of the NCS has become more interesting in the last decade for its potential in different inherently distributed applications like terrestrial exploration, power systems, factory automation, remote diagnostic, experimental facilities, automotive industry, transportation systems and many others.

Visible trend in the implementation of the NCS is the utilization of the standard networking solutions, resulting in the reduced implementation costs and simplified system integration. Although the utilization of wired communication is common practice in the NCS design, wireless control system become more interesting since the installation flexibility reduces installation costs, at the same time sustaining reliability, security and scalability [4]. To the other end, packet switched data communication in both wired and wireless networks suffer from non-deterministic medium access time, congestion under the high network load and signal interference, so delays of values higher than sampling period, packet losses and packet disorder are to be expected in system operation,

affecting the control-loop performance and system stability [5, 6].

Recent survey papers [6–8] identify several network-related communication properties influencing the NCS operation and surveys different methodologies of the NCS design for handling such constraints. The classification of constraint properties regarding their varying, unbounded, non-deterministic or stochastic nature was linked with the characteristics of networking infrastructure utilized in different NCS applications. Two strategies are identified as frameworks addressing the networking issues in NCS design: robustness and adaptation framework [6]. In the typical robustness framework approaches, NCS is handled as a traditional input-delay system, where robustness of the control solution is established either by some optimization design approach [9–11] or the fuzzy-model-based approach [12, 13]. Common property of the robustness framework methodologies is that they do not use the actual time-delay information. The approach presented in [14] utilizes appropriate discretization of NCS in order to obtain stabilizing feedback controller via LMI and addresses the typical constraints, such as time-varying sampling intervals, time-varying delays, and consecutive packet losses by assuming that they are bounded. The design methodology presented in [15] is based on the trade-off between transmission intervals, delays and performance. Beside the previously given typical constraints, the paper additionally addresses the issues related to the competition of multiple nodes accessing networks. On the other hand, model-based NCSs architecture, proposed in [16], considers the typical constraints and clock asynchronization constraints.

University of Belgrade – School of Electrical Engineering, Bulevar kralja Aleksandra 73, Belgrade 11020, Serbia, * Department of Electronics, popovici@etf.bg.ac.rs, ** Signals and Systems Department, Distributed Measurement and Control Laboratory, rakić@etf.rs

In adaptation framework two mainstream approaches were identified as the predictive control approach and the stochastic switched system modeling & control approach. The predictive approach usually presumes the appropriate buffering and packet-selection logic for delay-adaptive control application, while packet losses are handled by using intrinsic predictive property of the control, ie control is calculated from the predictions found in the previously received packet [17, 18]. In the latter approach of stochastic switched systems, feedback and forward channel delays are appropriately modelled and control solution is given in the form of a delay-variable controller gain [19, 20].

The time property is found to be a primary application context [21] resulting in the time-aware model of the information processing. The time-awareness is related to the operation concept where the outcome of the operation is dependent on the time-related properties embedded in the operation request, but also on the time-instance when the operation is executing [22].

Adoption of service-oriented style for the design of software components as network-accessible services that implement distributed NCS functionalities enables modularity, reusability, and rapid development [23, 24]. The NCS architecture in [25] utilizes multi-agent middleware [26] and service-oriented approach to provide flexible and scalable framework for the consistent distribution of the control system functionalities to the dedicated time-aware components and identifies their basic properties.

The aim of this paper is to address network-related issues regarding closed-loop control in the domain of the system architecture, rather than in the domain of control theory that can be found in the common approaches. Since all NCS system components share the common temporal property at the functional level, we adopted the framework [25] for consistent stateless implementation of separate NCS functionalities compliant to time-aware operation concept.

The control operation is given according to the predictive control approach as a modified form of generalized predictive control (GPC) [27] solution, while the closed-loop data-flow is supported by dedicated middleware components. The design of control algorithm is independent on the properties of communication network. The closed-loop performance is achieved under the condition that cumulative latencies and/or packet loss dropouts are bounded by the control horizon. The seamless utilization of various underlying wired and wireless communication technologies is enabled by the proposed communication model. The introduction of dedicated system-level components enables simplified migration from traditional tightly-coupled control system design to the loosely coupled, technology independent network-based design. The integration of the components at the system level is supported through the time-aware service execution concept and the unified time-stamped data model. Model of time-aware service execution is related to a pair of the time-aware read and write operations performed over the buffering data structure.

The rest of the paper is organized as follows. The overview and the effectiveness of the proposed solution is presented in Section 2, while the details regarding the operation and implementation of particular NCS functionalities are discussed in Section 3. Section 4 contains the case study of drying chamber temperature NCS operation, while concluding remarks and directions for the future work are given in the final Section 6.

2 The proposed NCS architecture

Architecture overview

Our architectural approach combines several architectural styles enabling flexible design that takes advantage of loose coupling between system components.

We adopted the service-oriented and component based architectural styles [23, 24, 28] as a uniform framework for the implementation of NCS functionalities. Sensing, actuating, control and communication functionalities are given in the form of independent system-level components manageable through the set of time-aware network accessible services [24, 29, 30]. The SOA architectural style defines an interaction model between service provider and service consumer, according to the service description available at the service registry. Adoption of SOA based architectural style enables decoupling of the process controller design from the properties of the control network. The service provider publishes service description and provides the implementation of actuating, sensing and control functionality in the form of passive system components. Provider services are bounded and invoked by service consumers that handle data transport between passive components. In this way, communication-related issues are dislocated from the scope of provider services and managed by unified middleware components named service agents (SAs). The coupling between provider and consumer services is provided through the endpoint buffering data structures and methods encapsulated in the form of unified buffering objects. These data structures, accessible through provider service interface, are containing the information according to the unified time-referenced data model. The model requires that each individual data is coupled with the time stamp information uniformly interpreted by all system level components and services. Together with inter-node time synchronization, data model enables time-aware execution of provider services.

Additionally, system level components as software objects have a defined communication interface containing methods and properties and conform to a recommended behavior common to all components within an architecture. Extending the same model of the plug & play functionality to system components, we introduce Electronic

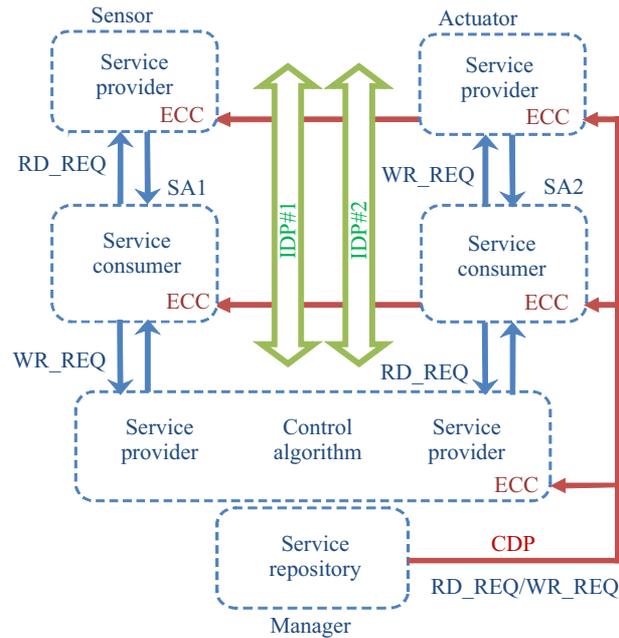


Fig. 1. Overview of NCS architecture

Component Configuration (ECC) as a component specific data structure that stores component-related information, like identification and configuration parameters. The basic elements of NCS architecture are presented in Fig. 1.

Data communication between system components is performed within IP network with the client/server end-system interaction model based on SOAP/XML messages [30]. Client and server reside at service consumer and provider side, respectively, and they exchange messages over asynchronous request-response messaging pattern. The both read and write service requests (RD_REQ and WR_REQ) at the server side are executing over the buffering structure containing the information according to the time-referenced data model. The payload of the response message corresponds to the time-context information embedded in request messages. The sensor, actuator and GPC processing are performed according to the time-aware processing concept, while the management of the time-context information is exclusively handled by SA operation. Further details regarding the operation of individual system-level components are given in the following sections.

From the deployment view, each physical network node contains a single component or a group of system-level components. The communication between components is performed over two different types of communication data paths: Information Data Path (IDP) and Configuration Data Path (CDP). IDP is defined as a data path for data exchange between two endpoint buffering structures accessible through the execution of provider services, while CDP is defined as a data path designated for the configuration of individual system component. The component configuration and service discovery processes are handled uniformly for all system components

through the Manager-node operation. IDP data communication between selected provider side buffering structures is performed through the configuration of ECC data structures of the registered service agent's. In this way, service agent is configured to manage automated data transport between distributed system-level components, regardless on its own and other components physical deployment. As the unified middleware components, SA isolates the idiosyncrasies of calling diverse services from a particular network node, whether they are local or remote, and it enables data exchange over the particular IDP.

Coping with network-induced problems

As identified in [6], networked system architecture gives rise to several issues influencing NCS design and operation. The effectiveness of the proposed solution in the context of these network related constraints is further analyzed.

Time delays, packet losses and disorder are the inherent properties of the switched-packet network as an underlying infrastructure of the NCS. The time-stamped data model along with the time-aware service-processing provides the framework for the effective decoupling of the control design and operation from handling such constraints. The system operation, under considered network effects of varying packet delays, packet losses and disorder, is verified and analyzed in the case study.

The effect of the time-varying sampling is usually put in the context of the varying transmission interval due to the scheduling of the packet transmissions in the NCS. Therefore, most of the research efforts, providing the robustness to this effect, are directed toward finding or optimizing the maximal allowable transfer interval. Since

Table 1. Comparison with the state-of-art approaches regarding network-induced issues

	Time delay	Packet loss & disorder	Time-varying sampling/transmission	Network access competition	Data quantization	Clock synchronization	Network security
[10, 11]	✓	✓			✓		
[15, 20]	✓	✓		✓			
[14]	✓	✓	✓				
[16]	✓	✓	✓			✓	
[17]		✓					
[18]	✓	✓					
Proposed	✓	✓	✓	✓		✓	✓

the transmitted data content in our approach is time-stamped, there is no uncertainty in data sampling due to the varying transmissions. On the other hand, the outcome of the time-aware service execution depends solely on the time-stamped data content and not on any information related to the data transport (packet arrival time). Therefore, the effects of time-varying sampling and transmission are equivalent to packet latency and they are handled in the same way at the system level.

The competition of multiple nodes accessing network is another typical network-induced phenomenon defined as a scheduling problem in the NCS literature. As each SA handles the data transport over the independent IDP, the scheduling of the network access within the single IDP is defined through the sequence of read and write service requests. Although the competition in network access between multiple IDPs exists, time-stamped data model along with the time-aware SA operation translate the problem to the domain of time varying transmissions, previously discussed. This issue can be alleviated by applying the scheduling policy to the configuration of time-related properties at the middleware layer, *ie* to the group of SAs.

In the context of network induced constraints, the data quantization is related to the limited network bandwidth. The solution is often found in the reduction of transmitted data size through the reduction of the word length. This leads to pronounced quantization problems of digital control systems. The proposed middleware solution realizes IDP transport of aggregated data with small transmission overhead, enabling the efficient utilization of the available bandwidth. Thus, our architectural solution does not elevate the data quantization problems due to network transport. The problems of data quantization, if any found, should be treated in the domain of the digital controller design. In the scenarios where network with high throughput is utilized the quantization problems are negligible.

The time asynchronization between the network nodes is known issue in the class of distributed systems that perform time-aware processing, like NCS. In our approach, the time synchronization of the system components is handled as a part of the Manager-node services. Although

the synchronization is performed periodically, the time asynchronization may occur inside the synchronization interval, leading to the effect of the model-process mismatch and the consequent problems of the closed-loop stability. In this case, the inter-node time synchronization can be established through the implementation of time-synchronization protocol, like [31], as a part of the standard IDP data transport request-response messaging pattern.

Though often omitted in control-related approaches to NCS design, network security and safety represent the important issue of the underlying networking. The problem is perceived to be even more significant in the future since the utilization of open standards and the solutions for data communication. Our solution encapsulates the system functionality in either network-accessible services or SAs as service consumers, so any security policy regarding the service usage can be implemented. Service access control is defined in the IDP settings at the Manager-node as a coupling between the provider and the consumer services. Other security objectives, like confidentiality, integrity and authentication, can be achieved by utilization of standard security communication protocols at different layers of the communication protocol stack.

The summary of the properties for the representative approaches to NCS design regarding the effectiveness in coping with the previously discussed network-induced constraints is given in Tab. 1. State-of-the-art approaches are chosen as the best ones in the sense of addressing most of the identified issues in their class of solutions. Beside the immediate benefits in the NCS design and implementation, proposed architectural approach directly addresses or provides support for handling wide-range of network-induced issues.

3 Operation of NCS components

As illustrated in Fig. 2, basic sensing, actuating and control algorithm functionalities are deployed to the separate network nodes, while the configuration and time synchronization of the system components are handled by the Manager-node.

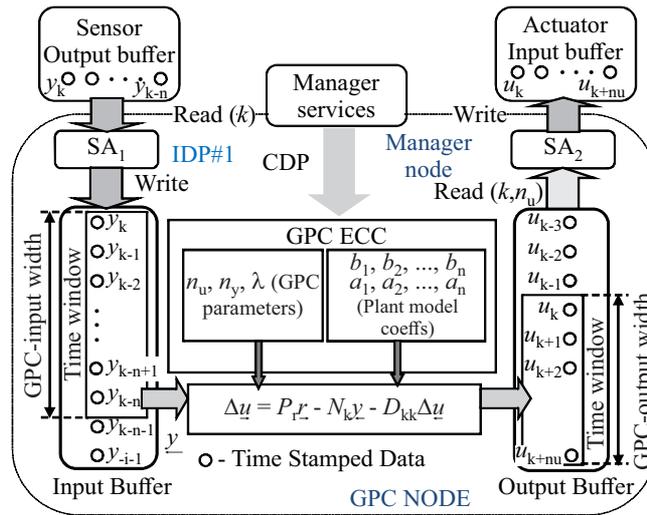


Fig. 2. The review of NCS operation

Actual inter-node data communication is performed over underlying Web Service technology [30] using SOAP/XML message format and standard HTTP protocol. Although data exchange can be implemented in a different application-specific way [24, 28], the SOAP/XML Web services is preferred implementation solution as it does not conflict with firewalls and HTTP proxies. The details regarding the operation, configuration and implementation of individual system-level components is presented in following sub-sections.

Service agent operation

The IDP communication is performed through a pair of service-read and service-write operations according to the predefined SA configuration. The individual read and write operation, from the SA’s point of view, is handled through the pair of the request and response messages. This way the IDP transport is a two-step process of successive data communications between the data producing and data consuming provider services through intermediate SA component.

The data transport over the IDP is handled through the selected transmission method supported by SA operation. The selection of SA transmission method and information regarding service provider endpoints are part of SA configuration. During regular operation, SA is controlled over the triggering interface resulting in the either periodically or sporadically performed data communication over IDP. Periodical execution of service agent is utilized with soft timer or hardware timer functionality and it is the appropriate model of the operation for the data communications with sensor devices, acquisition systems and other services and applications that are periodically updating their data structures. Sporadic execution of the service agent operation corresponds to the event-based notification triggering, realized through *eg* publish-subscribe messaging pattern. Regardless on the selected

transmission method, SA provides adequate time-context information as a part of read and write request messages. Basically, the time-context information defines the time window of requested or provided data used in the execution of read/write service requests, and the time instance when the IDP transport is triggered.

During regular operation, SA₁ from Fig. 2, performs IDP#1 data transport on the feedback channel, from sensor’s to GPC’s input buffering structure, ie it accepts read response packets of time-stamped sensor data and forwards those data over GPC write service interface call. SA₁ provides adequate time-context information in order to provide the continuous sequence of time-series data in the GPC input buffering structure. The SA₂ is configured for IDP#2 data transport on the forward channel, from the GPC’s output buffer, containing calculated control increments, to actuator side. The SA₂ updates the actuator side buffering structure with the updated information read from GPC output. The dataset size corresponds to the control horizon nu, selected to be big enough to compensate the cumulative latencies introduced along feedback and forward channels IDP#1 and IDP#2. The service agent configuration and IDP management are explained under the Data-path configuration and management sub-section.

Operation of transducers

The basic set of sensor-side operations is related to signal conditioning, data acquisition and time-stamping of acquired data samples. Although this basic set of functionalities is utilizing the time-referenced data, the execution of these operations is unrelated to the time-instance they are performed. Therefore, this set of the operations is referenced as a set of time-unaware operations at the sensor side. Beside these functionalities the sensor performs the local read and remote write operation accessible through the service provider interface. The both read

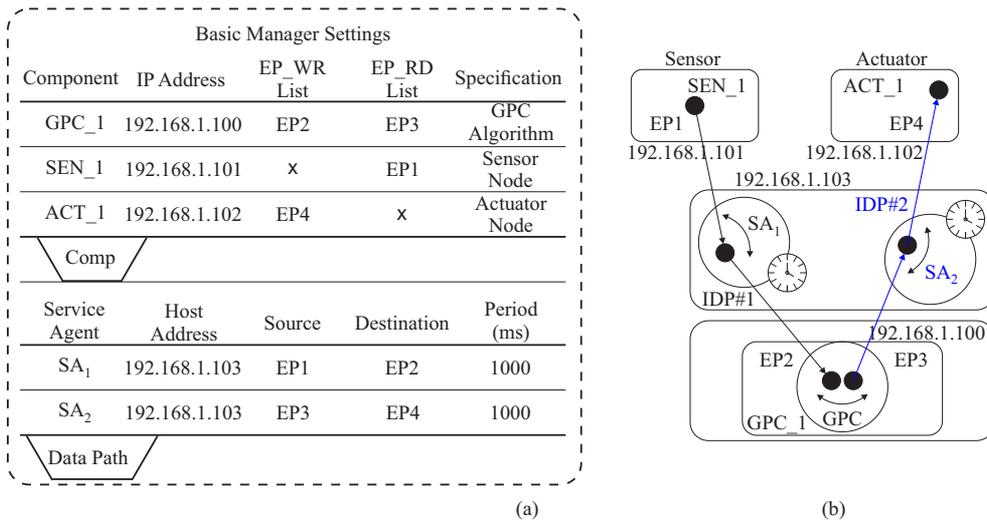


Fig. 3. Example of system configuration: (a) – component properties and SA settings, (b) – overview of IDP network configuration

and write operations are executed in time-aware fashion over the buffering data structure encapsulated in the unified buffering object. While the buffer write operation is related to the preserving of time-stamped data samples, the read operation is associated to the execution of the remote read service request initiated by corresponding SA. The read service response provides a demanded time window of buffered data. Time-window information is interpreted as a time span between the time instance when the read service request is generated (time instance k) and the time-stamp information of the last successfully transferred data.

Similarly, at the actuator side the fixed dataset size (with n_u data instances) of time-stamped control increments is written at appropriate time-dependent position in buffering data structure. The operation is a result of remote write service request initiated through the operation of the corresponding SA. In the packet disorder scenario, received packets are discarded since the buffer write operation employs past packets rejection logic. Beside the time-aware processing related to the execution of remote write and local read operation over the buffering data structure, actuator performs a set of basic functionalities unaffected by the time instance of their execution. These operations are related to the calculation and application of control variable, where the calculation operation, in the case of the adopted GPC-based NCS, is the simple integration (accumulation) of control increments.

The controller operation

The control functionality is implemented in the form of GPC algorithm [27], as one of the available solutions for addressing network related problems in process control applications. After configuration, GPC’s ECC contains algorithm’s parameters: prediction horizon n_y , control horizon n_u and control increments’ weight λ , as well as

the coefficients b_j and a_i of the discrete process model

$$G(z^{-1}) = \frac{\sum_{j=1}^n b_j z^{-j}}{1 + \sum_{i=1}^n a_i z^{-i}}. \quad (1)$$

In order to compensate data path latencies, GPC control law for the CARIMA process description [27] is adopted in the form

$$\Delta \underline{u}_k = P_r \underline{r}_k - N_k \underline{y}_k - D_{kk} \Delta \underline{u}_k \quad (2)$$

to produce vector $\Delta \underline{u}_k = [\Delta u_k + \Delta u_{k+1}, \dots, \Delta u_{k+n_u}]$ of n_u future control increment samples, where convenient “arrowed” convention [27] is used to describe vectors of buffered measurements $\underline{y}_k = [y_k, y_{k-1}, \dots, y_{k-n}]$ previously calculated and buffered control increments $\Delta \underline{u}_k = [\Delta u_{k-1}, \Delta u_{k-2}, \dots, \Delta u_{k-n}]$ and future references $\underline{r}_k = [r_{k+1}, r_{k+2}, \dots, r_{k+n_y}]$. Matrices P_r , N_k and D_{kk} are calculated prior to first GPC execution from algorithm’s parameters and plant model and they are kept persistent for the calculation of control increment vector in all future GPC calls.

The execution of write service operation over the GPC input buffering structure triggers the execution of GPC control algorithm, which takes preconfigured time-window \underline{y}_k of sensor data, calculates and writes control increment’s vector $\Delta \underline{u}_k$ to its output buffer. Although, the input and output buffer related operations are executed according to the time-aware concept, the execution of control algorithm represent processing of the sequence of discrete-time data.

Data-path configuration and management

Data path configuration and management is performed by manager operation. Initially system-level components publish the details regarding configuration details and services through the service discovery procedure.

During the discovery procedure the information regarding the provider service endpoints and the service description are gathered in the service registry.

Before defining IDP connections through the configuration of available SAs, the operator accesses Basic Manager Settings with sensor, actuator, and algorithm-related properties. Table in Fig. 3(a) shows properties of the registered system components, including their logical IP address, service endpoints information and basic description. The endpoints information identifies the location and binding for read and write provider service endpoints.

Additionally, the Basic Manager Settings provide the interface for the specification of IDPs through the SA configuration. As shown in Fig. 3(a) IDP is specified by defining the connected source (EP_RD) and destination (EP_WR) provider service endpoints, as well as the timing parameter. The SA₁ and SA₂ settings from the Fig. 3(a) correspond to the network configuration of IDPs presented in Fig. 3(b). After the IDPs are specified, the particular component-side SA configurations are updated over the CDP.

4 Case study: drying chamber temperature control

Since the data transport functionality in the proposed NCS architecture is exclusively handled by the SA components, the network induced problems affect only their operation. The particular SA actions regarding either missing or delayed service responses are defined by the already presented SA operational concept. However, the closed-loop operation of the NCS, under the induced communication constraints, is further analyzed in the presented case study of drying chamber temperature control. The NCS design utilizes the proposed architectural framework for the implementation of the sensing, actuating, control and communication components of the control system. Laboratory setup includes drying chamber process, controlled through heater voltage, sensing element for the measurement of the chamber temperature, as primary regulated variable, and Ethernet network accessible sensor, actuator and GPC nodes.

Preliminary process identification yielded continuous first-order model with dead-time

$$G(s) = \frac{K}{T_p s + 1} e^{-\tau s}, \quad K \in [0.05, 0.4], \quad T_p \approx 4 \text{ s}, \quad \tau \approx 1 \text{ s},$$

where dynamics (time-constant T_p and dead-time τ) is rather independent on operating conditions, while the dominant process nonlinearity is exhibited through the gain K change with the operating temperature. The gain takes high values at the mid-range temperatures (around 45 °C), but it drops to low values when the temperature goes to both limits of its range: at ambient temperatures (around 25 °C) for control dead-zone, and at high temperature values (up to 60 °C) for control saturation. Also, repercussions of the processes of heating and cooling on

the gain K are not equivalent, since temperature characteristics of the gain K exhibits hysteretic properties.

In order to obtain process model, acceptable for the wide-range temperature control, sampling period and parameters of the model (3) are adopted as follows

$$T_s = 1 \text{ s}, \quad K = 0.2, \quad T_p = 4 \text{ s}, \quad \tau = 1 \text{ s}$$

giving zero-order hold discrete model for the prediction in the form

$$G(z^{-1}) = \frac{b_2 z^{-2}}{1 + a_1 z^{-1}}, \quad a_1 = -e^{-T_s/T_p} = -0.7788, \\ b_2 = K(1 - e^{-T_s/T_p}) = 0.0442.$$

Parameters for the GPC control algorithm are chosen as $n_u = 10$, $n_y = 10$, $\lambda = 0.3$, where adopted control horizon n_u enables reliable system operation with network latencies up to 10 sample periods, while prediction horizon n_y is chosen to cover the process open-loop rise-time. Weighting λ is tuned to limit the amplitudes of control increments up to the unity value. The operation of both service agent SA₁ and SA₂, located at GPC node, is configured with same timer event periods set to the value $T_s = 1 \text{ s}$, same as the adopted sampling period.

The operation of the NCS is verified in network configurations including induced nondeterministic long-delay packet latencies and packet loss scenarios. In the packet loss scenarios, the service agent SA₂ is forced to reject a predefined number of successive packets of control increments directed from GPC algorithm to the actuator. The scenario resembles the worst-case situation, where the control increments are not available at the actuator side while the reference change takes place. Figure 4 shows the obtained signal waveforms for process temperature, control increments and integrated control.

During the packet loss period, the control system works in the open-loop, where the actuator applies future control increments within the control horizon from the previously received dataset. This way the basic actuator functionality is unaffected in the absence of data packets. After the communication is retained, GPC algorithm regains the closed-loop control and effectively eliminates the further influence of the packet loss disturbance.

In the set of nondeterministic packet latency scenarios, a time-varying delay was injected by the SA₂ across IDP#2, resulting in the latency between moment when the packet is generated by the GPC algorithm and the time instant when the actuator, according to its local time, receives a packet. Figure 5 shows the obtained response signal waveforms for induced network latency up to 1, 2 and 6 sampling periods, as presented in Fig. 5(d).

The induced data-path packet latency forces actuator to apply the set of GPC-calculated control increments from the previously received packet resulting in the scenario similar to the packet loss. The influence of the packet latency on the obtained responses is observed only for successive packets with the latencies higher than

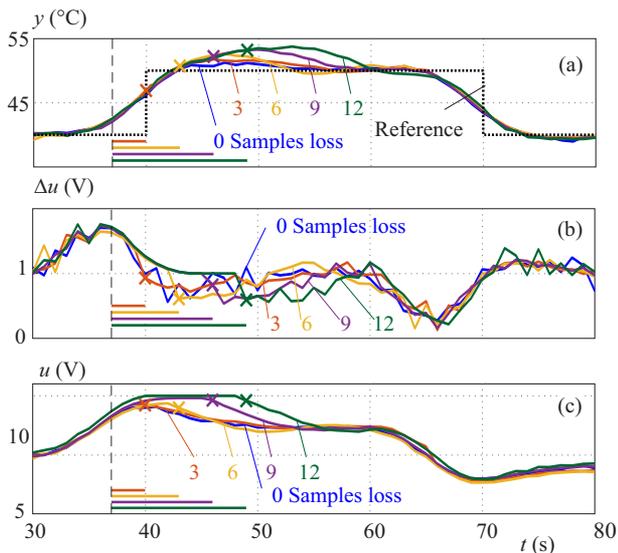


Fig. 4. Signal waveforms for the different packet loss scenarios: (a) – process temperature; (b) – actuator’s control signal increments; (c) – control signal applied by actuator

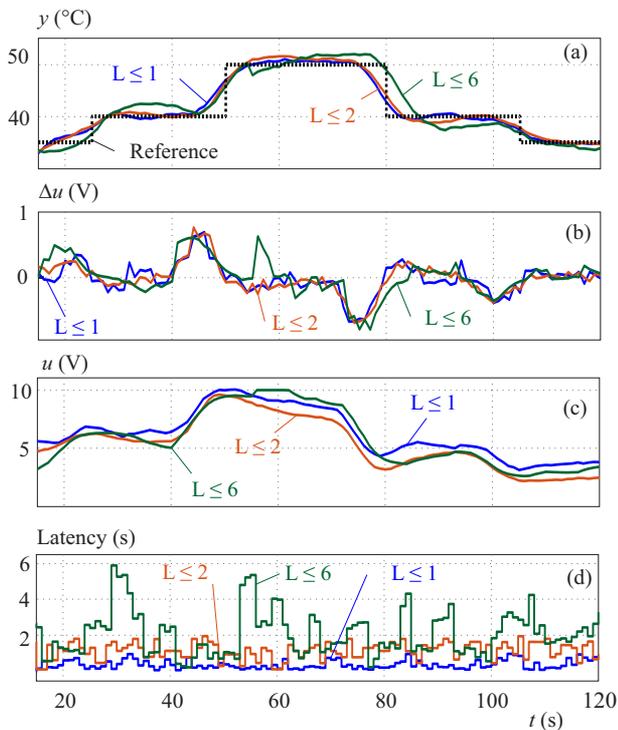


Fig. 5. The time domain responses for injected nondeterministic packet latency (number of samples L): (a) – process temperature, (b) – actuator’s control signal increments, (c) – control signal, (d) – packet latency

the several sampling periods. Nondeterministic latencies, resulting in packet disorder are handled by the logic, implemented on the actuator side, that rejects packet with the time-stamp older than time-stamp of the previously received packets.

The analysis of the system operation in all considered scenarios confirms that the predictive nature of control algorithm, the time-aware concept of service execution and the support for automated data transport represent

the adequate framework for the implementation of NCS functionalities.

5 Conclusion

Systematic architectural approach for the design and implementation of NCS is utilized to overcome the constraints of NCS closed-loop operation related to network-induced communication imperfections. The proposed solution decouples the communication-related issues from the design of basic sensing, actuating and control functionality. The decoupling model adopts component-based and service-oriented architectural styles, providing the framework for the implementation of system level components in the form of network accessible services. The inter-component communication is performed through service provider-consumer interaction model, where the provider services are bounded and invoked by service consumers. The effectiveness of the NCS operation is successfully verified in experimental setup with the extensive packet loss and latency scenarios.

Although applied to NCS design, the presented architectural approach introduces the framework for seamless integration of different end-devices and algorithms in various networked applications. The hierarchical extension and its utilization in complex distributed systems are planned as a part of the future work.

Acknowledgements

This work was supported by the Serbian Ministry of Education, Science and Technological Development under contracts no. TR-32043 and TR-33024.

REFERENCES

- [1] P. Gaj, J. Jasperneite and M. Felser, “Computer Communication Within Industrial Distributed Environment – a Survey” *IEEE Transactions on Industrial Informatics* vol. 9, no. 1, Feb 2013, pp. 182–189.
- [2] S. Komorita, “Loosely Coupled Service Composition for Deployment of Next Generation Service Overlay Networks”, *IEEE Communications Magazine* vol. 50, no. 1, Jan 2012, pp. 62–72.
- [3] J. Qiu, H. Gao and M. Chow, “Networked Control and Industrial Applications [Special section introduction]” *IEEE Transactions on Industrial Electronics* vol. 63, no. 2, Feb 2016, pp. 1203–1206.
- [4] P. Park, S. C. Ergen, C. Fischione, C. Lu and K. H. Johansson, “Wireless Network Design for Control Systems: a Survey” *IEEE Communications Surveys & Tutorials* vol. 20, no. 2, 2018, pp. 978–1013.
- [5] S. Yang and Y. Cao, “Networked Control Systems and Wireless Sensor Networks: Theories and Applications”, *International Journal of Systems Science* vol. 39, no. 11, 2008, pp. 1041–1044.
- [6] L. Zhang, H. Gao and O. Kaynak, “Network-Induced Constraints Networked Control Systems: a Survey”, *IEEE Transactions on Industrial Informatics* vol. 9, no. 1, Feb 2013, pp. 403–416.
- [7] D. Zhang, P. Shi, Q. Wang and L. Yu, “Analysis and Synthesis of Networked Control Systems: a Survey of Recent Advances and Challenges”, *ISA Transactions* vol. 66, Jan 2017, pp. 376–392.

- [8] X. Ge, F. Yang and Q. Han, "Distributed Networked Control Systems: a Brief Overview", *Information Sciences* vol. 380, no. 20, Feb 2017, pp. 117–131.
- [9] M. Cloosterman, N. van de Wouw, W. Heemels and H. Nijmeijer, "Stability of Networked Control Systems with Uncertain Time-Varying Delays", *IEEE Transactions on Automatic Control* vol. 54, no. 7, July 2009, pp. 1575–1580.
- [10] H. Gao and T. Chen, "Network-Based Output Tracking Control", *IEEE Transactions on Automatic Control* vol. 53, no. 3, 2008, pp. 655–667.
- [11] E. Tian, D. Yue and C. Peng, "Quantized Output Feedback Control for Networked Control Systems", *Information Sciences* vol. 178, no. 12, 2008, pp. 2734–2749.
- [12] Y. Wang, Y. Zheng, R. Yang, H. Fang and H. Ren, "Performance Analysis of Takagi-Sugeno Fuzzy Model for Run-to-run Control with Stochastic Metrology Delay", *IFAC Proceedings Volumes* vol. 44, no.1, 2011, pp. 489–494.
- [13] M. Blaho, J. Murgas, E. Viszus and P. Fodrek, "Reducing the Impact of Uncertainties Networked Control Systems Using Type-2 Fuzzy Logic", *Journal of Electrical Engineering* vol. 65, no. 6, 2014.
- [14] M. Cloosterman, L. Hetel, N. van de Wouw, W. Heemels, J. Daafouz and H. Nijmeijer, "Controller Synthesis for Networked Control Systems", *Automatica* vol. 46, no. 10, 2010, pp. 1584–1594.
- [15] W. Heemels, A. Teel, N. van de Wouw and D. Nešić, "Networked Control Systems with Communication Constraints: Tradeoffs between Transmission Intervals, Delays and Performance", *IEEE Transactions on Automatic Control* vol. 55, no. 8, 2010, pp. 1781–1796.
- [16] I. Polushin, P. Liu and C. Lung, "On the Model-Based Approach to Nonlinear Networked Control Systems", *Automatica* vol. 44, no. 9, 2008, pp. 2409–2414.
- [17] Q. Nguyen, V. Veselý, A. Kozáková and P. Pakshin, "Networked Robust Predictive Control Systems Design with Packet Loss", *Journal of Electrical Engineering* vol. 65, no. 1, 2014.
- [18] G. Ewald and M. Brdys, "Model Predictive Controller for Networked Control Systems", *IFAC Proceedings Volumes* vol. 43, no. 8, 2010, pp. 274–279.
- [19] K. Kobayashi and K. Hiraishi, "Design of Networked Control Systems using a Stochastic Switching Systems Approach", *IECON 2012 – 38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, QC, 2012, pp. 3170–3175.
- [20] M. Donkers, W. Heemels, D. Bernardini, A. Bemporad and V. Shneer, "Stability Analysis of Stochastic Networked Control Systems", *Automatica* vol. 48, no. 5, 2012, pp. 917–925.
- [21] E. Pascalau, G. J. Nalepa and K. Kluza, "Towards a Better Understanding of Context-Aware Applications", *2013 Federated Conference on Computer Science and Information Systems Krakow*, 2013, pp. 959–962.
- [22] M. Weiss, J. Eidson, C. Barry, D. Broman, L. Goldin, B. Iannucci, E. A. Lee and K. Stanton, "Time-Aware Applications, Computers, and Communication Systems (TAACCS)", *NIST Technical Note* 1867, 1.2.2015.
- [23] S. Karnouskos, A. W. Colombo, F. Jammes, J. Delsing and T. Bangemann, "Towards an architecture for Service-Oriented Process Monitoring and Control", *IECON 2010 – 36th Annual Conference on IEEE Industrial Electronics Society*, Glendale, AZ, 2010, pp. 1385–1391.
- [24] T. Cucinotta, A. Mancina, G. Anastasi, G. Lipari, L. Mangeruca, R. Checcoza and F. Rusina, "A Real-Time Service-Oriented Architecture for Industrial Automation", *IEEE Transactions on Industrial Informatics* vol. 5, no. 3, 2009, pp. 267–277.
- [25] A. Z. Rakic, N. S. Bezanic and I. T. Popovic, "Novel Architecture for Networked Control Systems", *2016 International Symposium on Industrial Electronics (INDEL)*, Banja Luka, 2016, pp. 1–6.
- [26] M. Razzaque, M. Milojevic-Jevric, A. Palade and S. Clarke, "Middleware for Internet of Things: a Survey", *IEEE Internet of Things Journal*, vol. 3, no. 1, Feb 2016, pp. 70–95.
- [27] J. A. Rossiter, "Model-Based Predictive Control: a Practical Approach", Boca Raton: CRC Press, 2003.
- [28] N. Kaur, R. Harrison and A. A. West, "A Service-Oriented Approach to Embedded Component-Based Manufacturing Automation", *2015 IEEE International Conference on Industrial Technology (ICIT)*, Seville, 2015, pp. 2964–2969.
- [29] N. Bezanic and I. Popovic, "Service-Oriented Implementation Model for Smart Transducers Network", *Computer Standards & Interfaces* vol. 38, 2015, pp. 78–83.
- [30] S. P. Lee, L. P. Chan and E. W. Lee, "Web Services Implementation Methodology for SOA Application", *2006 4th IEEE International Conference on Industrial Informatics*, Singapore, 2006, pp. 335–340.
- [31] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Std 1588-2008, 2008.

Received 26 June 2018

Ivan Popović received the Dipl Ing, Magister, and PhD degrees from the School of Electrical Engineering, University of Belgrade, Serbia, in 1996, 1999, and 2007, respectively. Since 1997, he has been affiliated with the University of Belgrade, School of Electrical Engineering, Serbia, where he is currently Assistant Professor, teaching courses in microprocessor system design and applied electronics. His research interests include real-time embedded system design, multiprocessor application, low power system design and analysis, multiprocessor architecture and smart transducer networks.

Aleksandar Rakić received the Dipl Ing, Magister, and PhD degrees, all in electrical engineering (process control) from the University of Belgrade, School of Electrical Engineering, Serbia, in 2000, 2004, and 2010, respectively. Since 2001 he has been affiliated with the University of Belgrade, School of Electrical Engineering, Serbia, where he is currently Associate Professor, teaching courses in conventional and advanced control. His research interests include control design in electrical drives, power electronics, unmanned aerial vehicles, distributed and networked systems, as well as advanced metering and customer-side systems within the smart grid.