IJEEC

International Journal of Electrical Engineering and Computing
Vol. 2, No. 2 (2018)

Case study

UDC 621.43:629.3]:005.72
DOI 10.7251/IJEEC1802091P

Networked and Cloud Control Systems - Modern Challenges in Control Engineering

Nataša Popović[1], Milica Naumović[2]

[1] Faculty of Electrical Engineering, East Sarajevo, Bosnia and Herzegovina
[2]Faculty of Electronic Engineering, Niš, Republic of Serbia

natasa.popovic@etf.ues.rs.ba, milica.naumovic@elfak.ni.ac.rs

Abstract— **The paper describes development of some control techniques used in the feedback control systems. Modern control trends, with attention to networked control systems (NCS), and basic assumptions for introducing a new control concept – control in the cloud are described. Networked control systems techniques initiated the development of concept of Internet of Things and are fundamental in cloud control systems design. Network induced delays that occur in NCSs, as a result of the presence of a communication network, and NCSs stability are considered, as well. Time delays and data dropouts influence on the NCS behavior and stability is analyzed, and some relations for maximum allowable delay bound estimation are provided. It is shown, on example of DC motor control, that networked system behaves in desired manner and remains stable if the delay is less then estimated maximum bound. When data dropouts occur, system delays become greater and lead to the system instability. Cloud control systems use some techniques of NCSs and cloud computing. These techniques are briefly presented and are necessary to complete control tasks in the cloud.**

Keywords- control systems; networked control systems; cloud control systems; delays; stability

I. INTRODUCTION

People have been trying to control and automate processes and different systems since ancient times [1]-[3]. Examples of this can be found in the ancient period and in the Middle Ages when such systems were based on mechanical control and automata. These systems were designed by observing the natural phenomena and processes on the basis of which scientists of that time made certain conclusions and used them in their inventions. Scientists were unable to mathematically describe such systems then, but despite that, systems functioned well and served the purpose for long periods of time. In the last hundred years, a mathematical apparatus, that enabled the most complex systems to be mathematically described and to define general guidelines for the analysis and synthesis of control systems, was developed. It is interesting that the classic control systems which have been used today are based on these from the ancient times. Development of electronics and communication technologies enabled existence of different types of control systems through last hundred years, such as analogue, digital, centralized, decentralized, networked, cloud control systems. Along with the development of control strategies, other technologies were developing as well, firstly communications and then computer science. This has led to the fact that modern control engineering has become a multidisciplinary field, which can be characterized as 3C - Computing, Communications, Control. In this paradigm all these three components are firmly coupled. Control engineering today cannot be imagined without computer and communication technologies, wired and wireless data networks.

From previously mentioned types of control systems, special attention has to be paid to networked control systems (NCS). The basic characteristic of the NCS is that the control loop is closed over the communication network and the operation of all its components (plant, actuators, sensors and controller) is coordinated via the network [4]-[11]. Communication network is, therefore, a key part of the NCS and a key element that makes such a system different from conventional control system. The industrial control systems were the first to use the communication network and were based on the use of control-oriented communication network technologies and protocols (CAN, DeviceNet, Profibus, Fieldbus). Nowadays, the Internet is increasingly used as a communication network in control systems, because of its rapid development and accessibility to everybody. In addition, the Internet allows the design and construction of large-scale control systems at a relatively low cost and provides them with the possibility of reconfiguration in a flexible way and easy maintenance. Due to all this, NCSs have found their place not only in industrial plants, but also in smart homes, traffic, teleoperations. Fundamental theoretical developments in the field of NCSs have enabled the development of today's ubiquitous concept of the Internet of Things (IoT) [9] and further of cloud control systems.

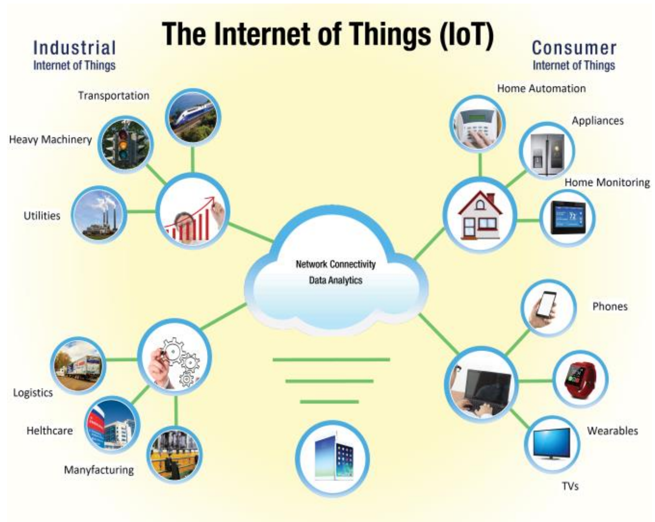


Figure 1. Internet of Things [12]

However, the benefits of using NCSs do not occur without shortcomings. The basis of each communication is the exchange of information. In conventional control systems, data exchange between system components is done virtually without the loss of information. However, the existence of a communication network in NCSs causes communication in which data losses are always present. The reason for this is the nature of the communication network itself, which in control systems causes the appearance of so-called communication constraints [9]. Communication constraints are mainly related to network induced delays (delays in transmitting sensor and control data), data consistency (some data packets may be missing during communication) and synchronization (different control components can work on different clocks). These limitations can significantly degrade the performance of the control system and make it unstable under certain conditions [13]-[16]. Maintaining the stability of a networked system is a basic task that needs to be fulfilled when NCS is designed. The analysis and criteria for the stability of NCS are mostly complex and developed primarily for scientific purposes. Therefore, they are largely inapplicable in practice. In recent years simpler ways of analyzing NCS stability have been proposed, trying to shorten the engineering time considerably. In this paper, a simple stability method is applied on networked DC motor to investigate the motor behavior in presence of network induced delays and data loss. The method is evaluated by simulation performed with TrueTime software, a Matlab extension for research of both network and control aspects of NCSs.

The paper is organized as follows. In section two, NCSs principles are explained with a focus on network induced delays and a system stability method used to determine the maximum allowable delay bound in a continuous networked control system. Networked control system behavior in a presence of time delays and data loss is demonstrated on an example of a DC motor control. The section three deals with cloud control systems basic principles. At the end, some concluding remarks are given.

II. NETWORKED CONTROL SYSTEMS

One of the basic control principles, which has been present since ancient times, is feedback control in which the signals from the system output are fed back to the input to reduce error and improve stability. More complex systems (e.g. refineries, power plants, chemical plants) have many components (actuators and sensors) distributed in different locations. These systems work on the principle that the information provided by the sensors is lead to the central processing unit where, based on them, a control signal is generated and sent to the plant via an actuator. This way of centralized control has been used for decades and has given excellent results in terms of system speed and reliability, but it often requires expensive equipment and high maintenance costs.

In the 1980s, in order to avoid these problems, the idea of connecting actuators, sensors, controllers and plant over the network emerged. In other words, the control loop was proposed to be closed via a communication channel. These control systems were called networked control systems - NCSs. In this period, industrial networked systems that accepted new control strategy were incompatible since various manufacturers developed various communication protocols (Profibus, Fieldbus, DeviceNet) [5], what greatly limited their usage. In addition, equipment and software were expensive and unavailable to a large number of potential users. Ethernet technology, which was standardized in the 1980s, provided the opportunity to overcome these problems. In the 1990s, the World Wide Web technology appeared which enabled the use of HTTP protocols or the Internet in the communication of networked devices. As a standardized technology, Internet has also been accepted in control systems engineering, and has provided additional possibilities for remote control through the Internet browser. In the last ten years, the development of wireless networked control systems has intensively been done, which enabled even easier access and control of systems. In recent years, there has been tremendous development of new concepts of networked control, which is initiated by the rapid development of Internet of Things, Fig. 1. The objects and devices that make up the IoT are connected to large databases over the network (the Internet). Internet of Things involves the collection, storage and processing of a large number of data obtained from sensors that detect changes in the physical status of objects [17],[18]. Thus, control systems using the concept of the Internet of Things should operate with a large amount of data they receive from various devices (cameras, microphones, RFID readers, sensors, etc.). In addition, real-time control as a key feature of real-time control applications is provided by Internet of Things concept, unlike some conventional networked control techniques and topologies.

A. Network induced delays

The main phenomenon of NCSs that influences NCS behavior and stability is network induced delay. Due to the existence of communication network in control system, transmission of the control signals between the controller and the actuator as well as the measurement signals between the sensor and the controller is performed with a certain delay. Delays occur in all components of the control system as each of them needs some time to process data, execute a certain algorithm and do required computations. Therefore, it can be said that delays occur in the process of data exchange between sensors, actuators and controllers that is performed over the

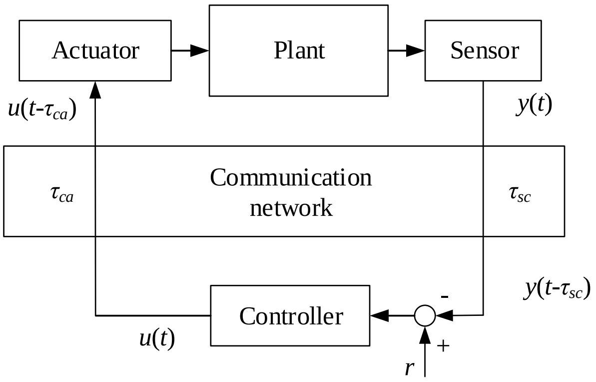


Figure 2. Networked control system and localization of delays

network. The structure of the NCS with appropriate network induced delays is presented in Fig. 2 [10],[11],[19],[20] . The analysis of NCS can be carried out in a continuous or discrete domain, although it is more advisable to perform it in a discrete domain, primarily because the digital data are transmitted over the communication network, and the components of modern control systems are mostly digital.

Characteristic delays in NCSs are shown in Fig 3. In general, there are two major network induced time delays in NCS [9]-[11],[13]-[18]: the time delay from sensor to controller (τ_{sc}) and the time delay from controller to actuator (τ_{ca}). According to Fig. 3, they are analytically expressed as

$$\begin{aligned}\tau_{sc} &= t_{cs} - t_{se} \\ \tau_{ca} &= t_{as} - t_{ce}\end{aligned} \tag{1}$$

where

t_{se} - time instant when sensor encapsulates output signals to a frame or data packet to be sent to controller,

t_{cs} - time instant when controller starts processing data received from sensor,

t_{ce} - time instant when controller encapsulates the control signal to a frame or data packet to be sent to actuator,

t_{as} - time instant when actuator starts processing control signal received from controller.

Communication or transmission delay is composed from τ_{sc} and τ_{ca} :

$$\tau_t = \tau_{sc} + \tau_{ca} \tag{2}$$

Delays τ_{sc} and τ_{ca} are different by their nature and time-varying as a result of the mechanisms used for data exchange. Fundamental delays that make up τ_{sc} and τ_{ca} are:

- queuing delay: the time for which the source (controller or actuator side system) has to wait for queuing and monitor network availability before starting to send frame or data packet,
- frame delay: the time it takes for the source to send the whole frame or data packet to the network,
- propagation delay: the time it takes for the frame or data packet to arrive from the source to the destination through a physical medium (communication network). This delay depends on the speed at which the signal propagates through the medium and the distance between the source and the destination.

Besides these three delays, additional delays can be induced by switches and routers that are used in the network. Network delays largely depend on other factors, such as the maximal network bandwidth specified by data exchange protocols, size of frame or data packets, traffic congestion, etc.

Delay τ_c represents the time it takes for the controller to process the data received from the sensor, generates the control signal and encapsulates it in the frame or data packet to be transmitted to the actuator via the network:

$$\tau_c = t_{ce} - t_{cs} \,. \tag{3}$$

Delay comprised from communication delay τ_t and controller delay τ_c represents control delay:

$$\tau_u = \tau_t + \tau_c \tag{4}$$

If it is known that the actuator and the sensor need a certain amount of time to process the data they received from controller and plant and to generate data to be transmitted, then the total delay can be defined as

$$\tau = \tau_t + \tau_c + \tau_a + \tau_s \tag{5}$$

where τ_a and τ_s are actuator and sensor delay, respectively. Delays in controller, actuator and sensor are usually very small in relation to the communication delay and they can be ignored. Therefore, it can be concluded that in NCS communication delay is dominant.

If data dropouts, which can be modeled as a delay, are

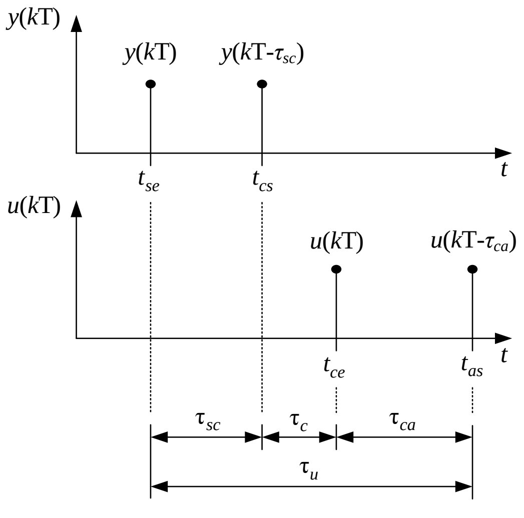


Figure 3. Characteristic delays in networked control systems

considered as well, then total delay in networked control system can be expressed as

$$\tau = \tau_t + \tau_c + \tau_a + \tau_s + dT \tag{6}$$

where d is the number of dropped data packets and T is the sampling period.

Depending on the type of communication network or media access control protocol that is used in the NCS, delays can be: time constant, time-varying, and random; bound and unbound [9],[11],[21]. The media access control protocol is designed for random access networks and scheduling networks.

Random access networks are based on CSMA (Carrier Sense Multiple Access) access control and are used in industrial control networks (DeviceNet, CAN) and Ethernet. Industrial control networks use CSMA/BA (Carrier Sense Multiple Access with Bitwise Arbitration) for collision detection while Ethernet uses CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Delays in random access networks are random and unbound, while in priority networks such as DeviceNet they are bound for high priority data packets and unbound for low priority data packets.

Scheduling networks are based on TP (Token Passing) and TDMA (Time Division Multiple Access) access control. TP is used in token bus (IEEE 802.4) and token ring (IEEE 802.5) topologies, while TDMA is used in FireWire. Delays in scheduling networks are bounded and can be considered constant since data transmission is achieved in predefined time intervals.

In delay analysis, the sampling period of the control system should also be considered [14]. The controller, sensor and actuator can have the same or different sampling periods. Namely, in networked feedback control systems, the values of the sampled signal must be transferred from the sensor to the plant between the two consecutive sampling instances, preserving the stability of the system. Therefore, the maximum allowable delay bound is defined as the maximum allowed time from the moment when the sensor reads the data from the output of the plant until the moment the actuator delivers the received data to the plant. A smaller sampling period involves better control quality, but in this case the network transmits a large number of data, what, on the other hand, results in longer delays. Additionally, a large number of data on the network can lead to traffic congestion and loss of a certain number of data packets, which violate the control quality and performance of the entire system.

B. *Stability of networked control systems*

So far, numerous methods have been developed to analyze the stability of NCS. They are mainly based on a complex mathematical apparatus. Lately, there are efforts to simplify them and thus make them easier to use in specific applications. In most cases these methods are based on the Lyapunov stability analysis and provide conditions for global asymptotic stability of the system. A simple stability analysis method will be presented below, which allows determination of the maximum allowable delay bound in a continuous NCS derived using finite difference approximation of the delay term and Lyapunov system stability theorem. Derived theorem and corollaries given bellow are from [13] where they are explained in more detail and appropriate proofs can be found there as well.

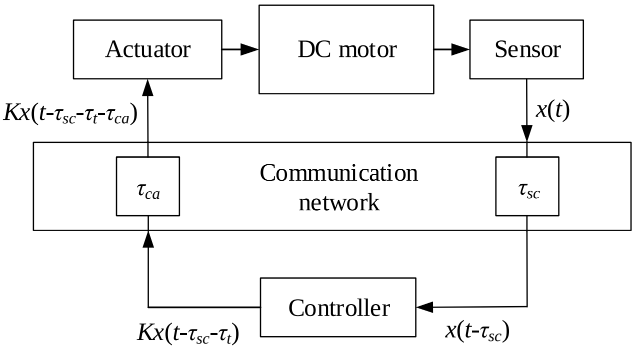


Figure 4. Networked control system and control law

Let us consider a continuous NCS from Fig. 4 with the state space representation

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)\\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)\end{aligned} \tag{7}$$

where $\mathbf{x}(t) \in R^n$, $\mathbf{u}(t) \in R^m$ and $\mathbf{y}(t) \in R^p$ are system state vector, system control input and system output, respectively, and **A**, **B**, **C** and **D** are constant matrices of appropriate size.

The state feedback controller is given with

$$\mathbf{u}(t) = \mathbf{K}\mathbf{x}(t-\tau) \tag{8}$$

where τ is defined in (4), and **K** is feedback control gains matrix.

The following is assumed:

- the sensor is time-driven (sensor samples the plant output periodically, in predefined time instants),
- the controller and the actuator are event-driven (they are activated when they receive messages over the network),
- data is transmitted as a single packet,
- old packets are rejected,
- all system states are measurable and ready to be sent,
- the time delay is small enough to be less than one unit of its measurement.

Theorem [13]: Suppose that assumptions above hold. For system (7) with the feedback control (8), the closed-loop system is globally asymptotically stable if $\lambda_i(\mathbf{\Psi}) \in C^-$, for i=1, 2, . . . , n and all the state variables' 2nd order reminders are small enough for the given value of τ , where $\mathbf{\Psi}$ is given by

$$\mathbf{\Psi} = \left[\left(\mathbf{I} + \tau \mathbf{B}\mathbf{K} \right)^{-1} \left(\mathbf{A} + \mathbf{B}\mathbf{K} \right) \right]. \tag{9}$$

Here, λ_i are eigenvalues of the $\mathbf{\Psi}$ and they must lie in the left-hand side of the complex plane. The 2nd order reminders are refered to Taylors series expansion of state variables $\mathbf{x}(t-\tau)$.

Corollary 1: For the control system (7) with the control law (8), the closed-loop system is globally asymptotically stable if

$$\tau < \frac{1}{\left\|\mathbf{BK}\right\|}, \tag{10}$$

where $\left\|\cdot\right\|$ is the spectral matrix norm.

Corollary 2: The system (7) with the control law (8) is asymptotically stable if

$$\tau < \frac{1}{\left|\lambda_{\min}\left(\mathbf{BK}\right)\right|}, \tag{11}$$

where $\lambda_{\min}$ is minimum eigenvalue of $\mathbf{BK}$.

Corollary 3: For system (7) with the control law (8), the closed-loop system is globally asymptotically stable if

$$\tau < \frac{1}{\left|\mathbf{KB}\right|}, \tag{12}$$

where $\left|\cdot\right|$ is the absolute value.

In the frequency domain, a closed-loop system is stable if the roots of the characteristics equation

$$\det\left[s\mathbf{I} - \mathbf{A} - \mathbf{BK} + \tau s\mathbf{BK}\right] = 0 \tag{13}$$

lie in the left-hand side of the *s*-plane. The last term in (13) has the greatest impact on the performance and stability of the system since it can significantly move the poles of the closed-loop system to the right-hand side of the *s*-plane.

C. Controller design and delay and data loss influence on the NCS behaviour

Delay and data loss influence on the NCS behavior will be illustrated on the example of DC motor control. Block diagram of NCS is given in Fig. 4. The motor model is taken from [23]:

$$G(s) = \frac{K_m}{s\left(T_m s + 1\right)}, \tag{14}$$

where $K_m = 24.8$ and $T_m = 0.0379$ s. Its equivalent model in the state space is given by

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & -26.39 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 654.4 \end{bmatrix} \mathbf{u}(t) \\ \mathbf{y}(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(t) \end{aligned} \tag{15}$$

The controller is designed according to the relation (8) so that the step response of the continuous system is determined with a pair of dominant poles characterized by $\xi = 0.707$ and $\omega_n = 20$ rad/s. Thus, vector $\mathbf{K}$ is obtained in a form

$$\mathbf{K} = \begin{bmatrix} -0.6111 & -0.0029 \end{bmatrix}. \tag{16}$$

According to (12) maximum allowable delay bound for the motor is $\tau < 0.528$ s.

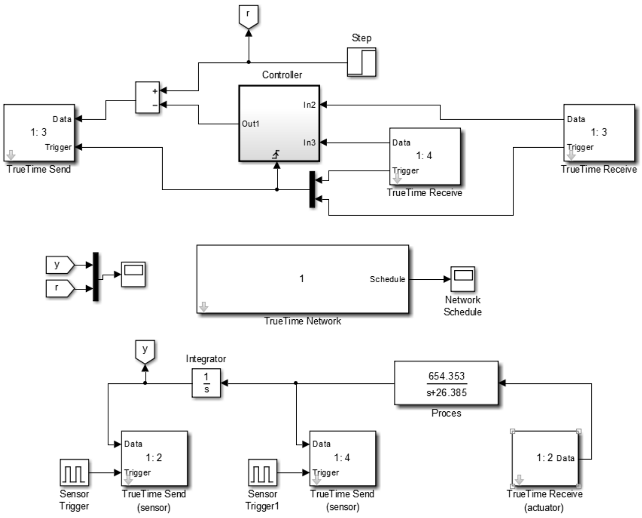
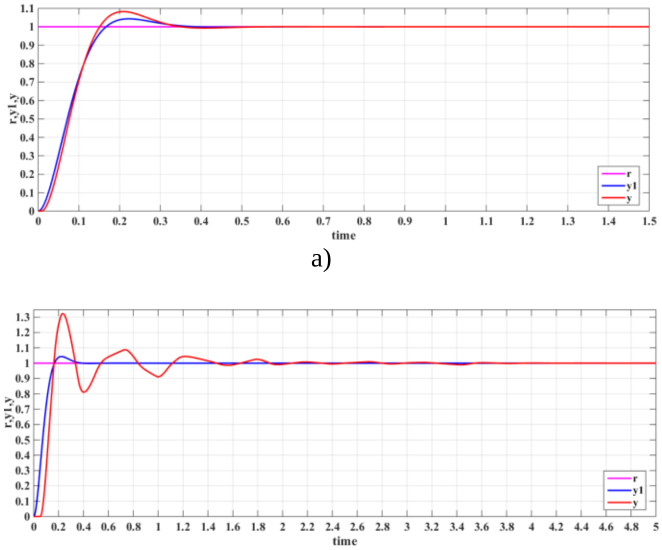


Figure 5. Simulation block diagram of DC motor control over communication network

D. Delay and data loss influence on the NCS behaviour

Simulation block diagram of NCS with DC motor is given in Fig. 5. TrueTime software [24], which is Matlab extension, was used to simulate control of a DC motor via a communication network. The simulation model in Fig. 5 is realized by a combination of Simulink blocks and TrueTime blocks. TrueTime blocks are TrueTime Send block and TrueTime Receive block, and they simulate sensors and actuators in the part of the system on the plant side and the input and output signals of the controller side, and the TrueTime Network that simulates the communication network. TrueTime Send and TrueTime Receive blocks are interfaces for the TrueTime Network block. TrueTime Network simulates media access protocols and data packets transmission in a local area network (LAN). It supports wired and wireless network types: CSMA/CD (Ethernet), CSMA/AMP (CAN), Round Robin (Token Bus), FDMA, TDMA, Switched Ethernet, FlexRay, PROFINET, NCM. Messages that come to this block are stored in appropriate input ports in the form of queues that function on the first-in-first-out principle. Then the messages are moved deeper into the block and after that they are sent to the corresponding output ports according to the network protocol used. When the message arrives at the output port, it causes an interrupt in the block that should receive the message. TrueTime Network block simulates packet loss and scheduling tasks in the network. For the purpose of the simulation the sensors are time-driven (the signals are sampled at predefined moments), and the controller and actuator are event-driven (depending on the messages generated at the network output).

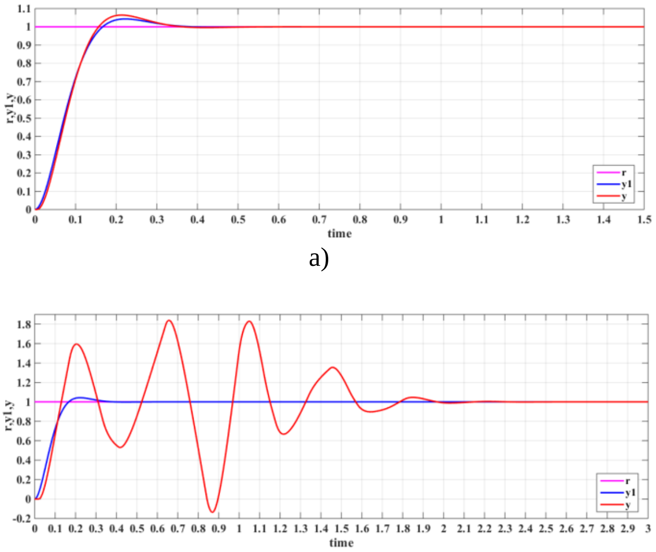
The behavior of the networked DC motor in a presence of network delay has been observed regarding the step response of the continuous non-networked DC motor in two cases. In the first case data dropouts have not been taken into consideration, while in the second case they have. In both cases 1Gb/s Ethernet and 80kb/s CAN (*Control Area Network*) network protocols were used. The chosen data rates are standardized for selected network type.



a)

b)

Figure 6. Step response of CAN system with no data dropouts (a) and with data dropouts (b)



a)

b)

Figure 7. Step response of Ethernet system with no data dropouts (a) and with data dropouts (b)

Step responses of the networked and non-networked DC motor are depicted in Fig. 6 and Fig. 7 with y and y1, respectively. These pictures demonstrate that there is a delay in the response of a NCS regarding the non-networked one.

Fig. 6a) and Fig. 7a) show responses in absence of data loss. Systems with no data loss give approximately the same response with an overshoot from 6.5% to 8.25%, a rise time of 0.1s and a settling time of about 0.4s. Ethernet system has better response and smaller delay than CAN system since it supports higher data rates. With the data rate increase, more data can be transmitted for the same period of time, so the system reaches the reference value faster, it has a smaller overshoot, smaller rise time and smaller network delay. The network delay present in the CAN system is 0.008s and in the Ethernet system it is 0.005s. According to the network delay value calculated with (12), the networked DC motor is globally asymptotically stable both, with CAN and Ethernet.

Fig. 6b) and Fig. 7b) show responses with data loss. Data loss probability in simulation is set to 0.8. It can be seen that both systems have inappropriate responses and they behave in undesired manner. Data loss induced additional delays which are now 0.054s for CAN system and 0.022s for Ethernet system. Longer delays lead to system instability and greatly impact the system performances. Smaller delay in Ethernet system with data dropouts gave worse response compared to CAN system which obtained grater delay. Ethernet is based on nondeterministic protocol primarily designed for applications where real-time features are not so important and supports higher data rates. Defined frame structure, collision detection mechanism, higher data rates and more data transmitting through the network cause communication in which the errors (dropouts) are very likely to happen. When, for example, sensor data dropouts occur, controller is not able to generate control signal and sends it to actuator on time. Retransmission of these lost data must be performed in predefined time interval, otherwise data are lost again. Protocol defines that retransmission continues until data goes through the network. If time for receiving retransmitted sensor data is too long, (e.g. longer than sensor sampling period) old sensor data are lost forever, and new sensor data will be transmitted in appropriate time instant. Since controller was not able to perform any control task, response is not as it is expected to be. On the other side, CAN is based on deterministic protocol designed especially for real-time applications and supports smaller data rates. CAN protocol tolerates some delays and has mechanisms to obtain desired system performances regardless delays and data dropouts. Data transmission procedure in CAN is based on message prioritizing, what enables proper allocation of network resources even in situations when errors happen. That is why, although data loss probability is high, CAN system produces better response than Ethernet system. Smaller values of data loss probability in simulation give better response for Ethernet system compared to CAN.

Besides its simplicity and easy application, the stability testing method used in the paper has some disadvantages. The main one is that, in the process of the mathematical formulation of the method, a certain number of approximations have been made and some higher order members have been neglected. These details can be found in [13]. In addition, by making the assumptions mentioned in II B., some aspects of data transfer that in real-time networks inevitably occur (e.g. packet segmentation, retransmission) are neglected. Initially, the method has been developed for communication networks with bounded delays. Although the delays are in principle unbound in CAN and Ethernet, the method can be applied to these network types for the following reasons. Namely, since CAN is based on deterministic protocol, it can be considered that the delays present there are constant and bounded (the control task must be performed at a predefined time interval). Although Ethernet is based on nondeterministic protocol, delays present there can be considered bound under certain conditions. Data rate is another important factor for considering the delay and stability of the system. Protocols using Ethernet and CAN have standard data rates that need to provide real-time control and minimum delays, i.e. to bound delays so that the control task is performed at a predefined time interval. Lower data rates in CAN mean larger delays, while higher data rates in Ethernet mean smaller delays. However, the value of maximum allowable delay bound obtained for DC motor control by the presented method is found to be much greater than one obtained by simulation. Namely, simulation results show that DC motor, without data dropouts, becomes unstable when $\tau = 0.08$ s. Hence, the method should be improved further and include all aspects of control over network as possible.

III. CLOUD CONTROL SYSTEMS

The term Cloud Control Systems is used to describe the new control engineering concept based on cloud computing technology and advanced networked control systems techniques [25].

A. *Cloud computing*

Cloud computing has been extensively developed since 2006 and is a technology in which the resources, located on remote servers, are accessed over the network (usually the Internet) [26]. Practically, the cloud system represents a shared source of resources that includes software, databases, hardware, and many services (calculations, access to data, etc.), whereby end users do not need to know the exact physical location and configuration of the resource (service) provider. Depending on the users' needs, clouds can be private (accessible only to a specific user and usually located within one organization's network), public (available to all users and accessed via the Internet) and hybrid (a combination of private and public). Cloud computing services are generally categorized into three types:

- Software as a Service (SaaS) - the user uses applications that are available in the cloud infrastructure, e.g. Google Apps and Dropbox.
- Platform as a Service (PaaS) - the user develops, tests and distributes applications running in a cloud on platforms that provide operating systems, development environments and software packages, e.g. Google App Engine, Microsoft Azure,
- Infrastructure as a Service (IaaS) - the user is provided with the possibility of using computing infrastructure placed in a cloud comprised of server, network infrastructure, database, e.g. Amazon Web Services and Rackspace Cloud.

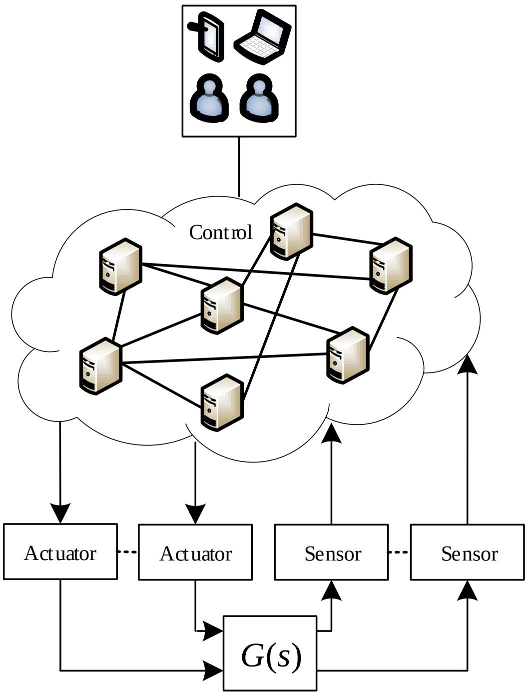


Figure 8. Control in the cloud

B. *Control in the cloud*

The *big data* term, which is present in modern computing, also exists in cloud control systems. Namely, the term refers to a large amount of data that is so extensive and complex that it is almost impossible to process it with up to now existing tools. In cloud control systems, these data are obtained from many sensors, cameras, RFID readers, and they also include various software records. The idea of the cloud control systems is to, after collecting data, send them to the cloud for processing, and then to send the control signals generated after processing to the actuator and further to the control object, Fig. 8.

Generally, the cloud control system can be described in discrete domain by the equations [25]

$$\begin{aligned} x(k+1) &= f\left(x(k), u(k), w(k)\right) \\ y(k) &= g\left(x(k), u(k), v(k)\right) \end{aligned} \tag{1}$$

where $x(k)$ is system state, $u(k)$ is system input, $y(k)$ is system output, $w(k)$ and $v(k)$ are disturbance and measurement noise, respectively. Functions $f\left(x(k),u(k),w(k)\right)$ and $g\left(x(k),u(k),v(k)\right)$ can be linear or nonlinear.

In fact, cloud control systems have their equivalent in networked control systems. The network through which the signals (obtained from the sensor and sent to the controller, and control signals generated in the controller and sent to the actuators) are transmitted corresponds to the cloud. The difference is that the cloud can be used to generate control signals and not only to provide network and computing infrastructure. This further means that some of the principles of NCSs can be used when designing a cloud control system, which is particularly useful in system stability testing and maintaining. In the cloud control system, the plant and controllers are observed as network nodes that are able to perform certain tasks and actions. The basic assumptions for these nodes are [25]:

- all nodes communicate with each other on the broadcast principle at the link level of the ISO-OSI reference model,
- all nodes in the cloud are intelligent enough to carry out a control task, they can all make the computations equally well, and the capacities of the nodes for computing are not predetermined,

- data transmission over the network is not ideal, time delays and data packet dropouts may occur,
- each node generates statistics of time delays and packet dropouts when communicating with other nodes,
- the plant (along with the sensors) is located in the node G (Fig. 8). A cloud task is initiated by a controller located in the node K, which also monitors the execution of the control task.

Control task can be implemented in two phases. The first phase is initial and the system acts as a classic NCS. The controller K receives the output signals from the plant and generates a set of variables according to the control algorithm (for example, a predictive algorithm that has been shown to be efficient in systems with delays and data dropouts [7],[25]). Network delays are compensated by a compensator which is an integral part of the plant node. After the first phase, the system switches to the cloud control phase, Fig. 9. The node K begins to broadcast the request over a predetermined frequency. All other nodes, belonging to the domain in which the node K is controller node (K1, ..., K6), receive this request. The request should contain the following information: the IP address of the plant node, the control algorithm and its corresponding parameters, the mathematical model of the plant, the complexity of the computations that the node should perform, etc. It should be emphasized that although K is a controller node, it does not need to perform any computations or to execute a control algorithm if there is another node in a domain with better performances to do the same task. When such nodes are identified (for example, K2, K4 and K5), they send acknowledgement to the node K that includes network delay information and statistics on the number of discarded packets when each identified node communicates with node G, and the available capabilities of the node to perform the necessary computations.

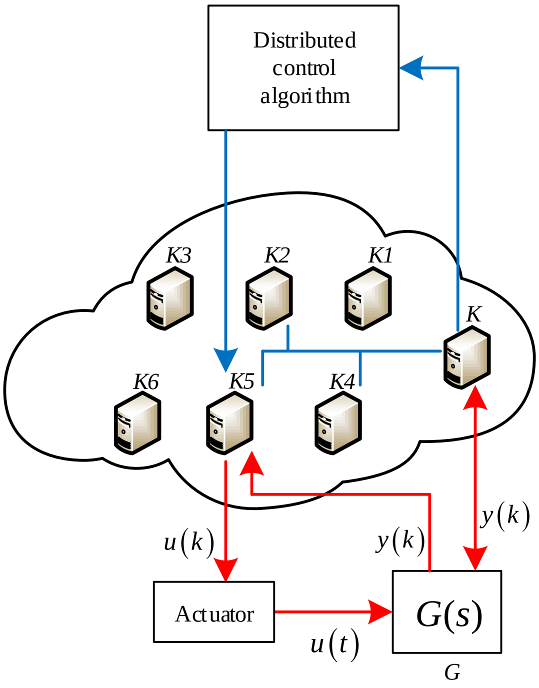


Figure 9. Block diagram of cloud control system

Based on the statistics received in acknowledgements, the node K evaluates the superiority of each node from the domain. Node K forms a list of potential nodes, including itself, and determines which nodes can be assigned a control task. Nodes whose superiority is small are ignored. If no node in a domain has superiority greater than the superiority of node K, the node K takes over the control task. By choosing superior nodes, controllers that will perform a cloud task are being selected.

When controller nodes in the cloud are verified, the node K sends the control data that contain the mathematical model of the plant, the estimated values of the state variables of the plant and the input variables at the moment of sending the control data, as well as the controller parameters. At the same time, node K sends a copy of the list of selected controller nodes to the node G enabling node G to communicate directly with them. When G receives this list, it sends the values of the current sensor readings to the cloud controller nodes, as well as the values of the readings obtained earlier. At the moment when one of the controllers receives the control data from node K and readings from node G, it applies them to the control algorithm and sends the computation results to the plant node G via the actuator in the form of data packets.

To obtain the cloud control system to work properly, all active controller nodes are required to send feedback to the node K at predefined time instants. If node K does not receive the information from the particular node at expected time instant, it removes the node from the list and substitutes it with the first one from the list of inactive nodes. At the same time, node G receives information of this change. The node G can receive packets with computed values of control variables from more than one active controller. In this case, the compensator which is an integral part of the node G, selects those that have arrived the last and uses them as inputs of the plant.

In Figure 4, K2, K4, and K5 are displayed as controller nodes, where K5 is currently the active node.

In addition to this type of cloud control, a cooperative way of cloud control is proposed. The idea is that multiple active controller nodes perform the control task in the cloud at the same time. In other words, each of the active nodes is assigned a part of the control task that needs to be done according to its current capacity. At predetermined moments, all active nodes (e.g. K2, K4, and K5) send the calculated data to the node K, it combines them into a control signal and sends it to the plant node G. Here, node K also checks the availability of other potential nodes (obtained by evaluation of superiority) so they could adequately take over a part of the control task from another active controller node if its superiority goes down. Node K acts here as a controller node and a task management server at the same time.

As control in the cloud is a new concept, there are a small number of solutions practically realized in automation and robotics so far. The main attention is paid to the development of the theoretical fundaments that will be applied in the future, especially when it comes to industrial systems. Thus, for example, the RoboEarth [27] project offers an infrastructure with all the necessary elements that allow closing the control loop in the robotic systems via the cloud, while the RoboEarth

Cloud Engine allows performing the cloud computing. In the Arduino [28] project, an open source platform that can be used in robotic and other cloud control systems has been developed. The WOAS (Web-Oriented Automation System) project [29] has been launched to explore the possibility of using cloud technology in industrial automation. In October 2018, Google announced that they were developing an open-access Cloud Robotics platform to "combine the power of artificial intelligence, robotics, and the cloud" that will be available in 2019 [30]. More examples can be found in [31]-[34]. Although the idea of using cloud computing in the control systems is interesting and some solutions have already been offered, the problems related to security (authentication, privacy, data confidentiality, system attacks and intrusions), big data processing and transmission, transmission delays in communication with a cloud, delays due to data processing and computing in a cloud and other issues are still to be investigated and solved. All these problems affect the performance of cloud control systems and thus real-time control applications.

IV. CONCLUSIONS

Control systems have been studied for decades. In this process many control strategies were developed and used in different applications. All of them have special requirements to be satisfied and limitations to overcome. Networked control systems paradigm is important control strategy and is present not only in automation but in different segments of everyday life. In such control system where the feedback is closed over the communication network, network delays are inevitable and always present. They lead to degradation of the performance of the system and may disturb its stability. Before designing a control part of the system, it is necessary to estimate the maximum allowable delay bound that will enable desired behavior of the NCS and preserve the system stability. The method specified in this paper has been used for determination of such delay bound and the validity of this method is confirmed on the example of DC motor control over the network. The same example illustrated how network delays and data dropouts affect the performances of the control system. Despite the delay and other problems related to NCS, these systems found place in many applications and enabled development of modern control techniques, such as cloud control. Besides NCS principles, cloud control systems use cloud computing techniques to obtain control signals. The development of such systems is ongoing process with outcomes that are yet to be obtained.

This paper is one of the first attempts of the authors in the field of NCSs and modern control techniques. Current and future research is focused on NCSs features implementation into real IoT solutions and possibilities of their control using cloud control techniques.

REFERENCES

[1] C. C. Bissell, A History of Automatic Control, Springer Handbook of Automation, 2009, Available: http://www.springer.com/978-3-540-78830-0

[2] S. Bennett, “A Brief History of Automatic Control”, IEEE Control Systems, pp. 17-22, June 1996

[3] E. Hayden, M. Assante, T. Conway, “An Abbreviated History of Automation and Industrial Controls System and Cybersecurity”, SANS Institute, August 2014

[4] P. V. Zhivoglyadov, R. H. Middleton, “Networked Control Design for Linear Systems”, Automatica 39, pp. 743-750, Elsevier Science Ltd. 2003

[5] Y. Tipsuwan, M. Chow, “Control Methodologies in Networked Control Systems”, Control Engineering Practice 11 pp. 1099–1111, Elsevier Ltd. 2003

[6] J. Baillieul, P. J. Antsaklis, “Control and Communication Challenges in Networked Real-Time Systems”, Special Issue of the Proceedings of the IEEE, Vol. 95, No 1, January 2007

[7] Y. Xia, Y. Gao, L. Yan, M. Fu, “Recent Progres in Networked Control Systems–A Survey”, International Journal of Automation and Computing 12 (4), pp. 343-367, August 2015

[8] R. A. Gupta, M. Y. Chow, “Overview of Networked Control Systems”, Networked Control Systems theory and Applications, pp.1-23, 2008.

[9] Y. N. Zhao, X. M. Sun, J. Zhang, P. Shi, “Networked Control Systems: The Communication Basics and Control Methodologies”, Mathematical Problems in Engineering, Hindawi Publishing Corporation, Vol. 2015, doi: http://dx.doi.org/10.1155/2015/639793

[10] S. Mittal, A. S. Siddiqui, “Networked Control Systems: Survey and Directions, Journal of Engineering Research and Studies”, Vol. 1, pp. 35-50. 2010

[11] M. Beindha, J. K. Mendiratta, “Networked Control Systems- A Survey”, I. J. Modern Education and Computer Science, 2013, 6, 42-48, doi: 10.5815/ijmecs.2013.06.06

[12] http://etsconnect.com/crash-course-things-inside-internet-things/the-internet-of-things-iot-pdf-8-638

[13] A. Khalil, J. Wang, “Stability and Time Delay Tolerance Analysis Approach for Networked Control Systems”, Mathematical Problems in Engineering, Hindawi Publishing Corporation, Vol. 2015, doi: http://dx.doi.org/10.1155/2015/812070

[14] W. Zhang, M. S. Branicky, S. M. Phillips, “Stability of Networked Control Systems”, IEEE Control Systems Magazine, 2001

[15] X. Zhang, X. Jiang, Q. L. Han, “An Improved Stability Criterion of Networked Control Systems”, American Control Conference, 2010, pp. 586-589

[16] Y. Shang, Y. Yuan, “Finite-Time Stabilization of Networked Control Systems with Packet Dropout”, TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol. 12, No. 9, 2014. Pp. 6750-6757, doi: 10.11591/telkomnika.v12i9.5061

[17] Tariq Samad, “Control Systems and the Internet of Things”, IEEE Control Systems Magazine, 2016, doi:10.1109/MCS.2015.2495022

[18] Internet of Things from Research and Inovation to Market Deployment, Edited by O.Vermesan and P. Friess, River Publishers, Aalborg, 2014,

Available: http://www.internet-of-things-research.eu

/pdf/IERC_Cluster_Book_2014_Ch.3_SRIA_WEB.pdf

[19] Y. Wang, L. He, M. Su, “The Research of Delay Characteristics in CAN Bus Networked Control System”, Journal of Computational Information systems 9: 18 (2013) 7517-7523

[20] N. P. Aung, Z. M. Naing, H. M. Tun, “Analysis of Time Delay Simulation in Networked Control System”, International Scholarly and Scientific Researc & Inovation 10(3) 2016

[21] A. S. A. Dahalan, A. R. Husain, M. B. N. Shah, M. I. Zakaria, “Time Delay Analysis in Networked Control System Based Controller Area Network”, proc. Of International Conference on Advances in Computer and Information Technology – ACIT 2012, doi: 10.3850/978-981-07-3161-8_ACIT-249

[22] Y. Shang, Y. Yuan, “Finite-Time Stabilization of Networked Control Systems with Packet Dropout”, TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol. 12, No. 9, 2014. Pp. 6750-6757, doi: 10.11591/telkomnika.v12i9.5061

[23] M. B. Naumović, “PI-like Observer Structures in Digitally Controlled DC Servo Drives: Theory and Experiment”, Electronics, Faculty of Electrical Engineering, University of Banja Luka, Vol. 15, No. 1, June 2011, pp. 30-39, ISSN 1450-5843,

http://electronics.etfbl.net/journal/Vol15No1/xPaper_05.pdf .

[24] A. Cervin, D. Henriksson, M. Ohlin, “TRUETIME 2.0 – Reference Manual”, Department of Automatic Control, Lund University, 2016

[25] Y. Xia, “Cloud Control Systems”, IEEE/CAA Journal of Automatica Sinica, 2(2), pp. 134-142, April 2015

[26] Cloud Computing, Available:

https://en.wikipedia.org/wiki/Cloud_computing

[27] What is RoboEarth, Available: http://www.roboearth.org

[28] Arduino, Available: http://www.arduino.cc

[29] Automation services from the cloud, Available: http://www.iebmedia.com/index.php?id=9254&parentid=74&themeid=255&hft=74&showdetail=true&bb=1

[30] http://goldberg.berkeley.edu/cloud-robotics/

[31] B. Kehoe, S. Patil, P. Abbeel, K. Goldberg, "A Survey of Research on Cloud Robotics and Automation", IEEE Transactions on Automation Science and Engineering, Vol. 12, No. 2, pp. 398-409, April 2015

[32] D. Hallmans, "Challenges and Opportunities When Introducing Cloud Computing into Embedded Systems", Paper prsented at INDIN 2015 IEEE, Cambridge, UK, Julu 2015

[33] P. Trivedi, K. Deshmukh, M. Shrivastava, "Cloud Computing for Intelligent Transportation System", International Journal of Soft Computing and Engineering, Vol. 2, Issue 3, pp. 568-572, July 2012

[34] M. Whaiduzzaman, M. Sookhak, A. Gani, R. Buyya, "A Survey on Vehicular Cloud Computing", Journal of Network and Computer Applications 40, pp. 325–344, 2014

Nataša Popović was born in Sarajevo, Bosnia and Herzegovina, in 1975. She received the B.Sc. degree in automation and electronics and M.Sc. in automation and informatics from the Faculty of Electrical Engineering, University of East Sarajevo. She is a Teaching Assistant with the Department of Automation and Control Systems with the same faculty. Her research interests are control systems, information and communication technologies, Internet of Things. She is currently preparing doctoral dissertation in the field of modern IC technologies implementation in control systems.

Milica Naumović was born in Niš, Serbia. She received the B.Sc., M.Sc. and Ph.D. degrees from the Faculty of Electronic Engineering, University of Niš, Serbia. She was professor with the same faculty and visiting professor with Faculty of Electrical Engineering, University of East Sarajevo. Her rich career has been focused on different problems in control systems and automation, theoretical and experimental investigations in system dynamics identification, digital and optimal control techniques. She published over one hundred scientific and research papers and a few monographies.