FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 9, N° 1, 2010, pp. 19 - 39

NETWORKED CONTROL SYSTEMS

UDC 681.325 681.5.01

Žarko Čučej, Dušan Gleich

University of Maribor, Slovenia
E-Mail: zarko.cucej@uni-mb.si, duasn.gleich@uni-mb.si

Abstract. *Over recent years, the use of a data network in a control-loop has attracted increasing attention due to its cost effective and flexible applications. One of the major challenges from the control point of view is the network-induced delay effect in the control-loop and, from the communication point of view, the development of dedicated networks that minimize delays and guarantee reliability. This paper contains a brief survey of recent developments regarding network protocols for automatization and networked control methodologies.*

Key words: *control, communications, systems*

1. INTRODUCTION

Modern control theory is largely based on the abstraction that information (signals) are transmitted along perfect communication channels and that computation is either instantaneous (continuous time) or periodic (discrete time). This abstraction has served this field well for 60 years and has led to many success stories within a wide-variety of applications.

Future applications of control will be much more information-rich than those of the past and will involve networked communications, distributive computing, and higher levels of logic and decision-making for a recent analysis of future directions within this area. New theory, algorithms, and demonstrations must be developed in which the basic input/output signals are data packets that may arrive at variable times, not necessarily in order, and sometimes not at all. Networks between sensors, actuation, and computation must be taken into account, and algorithms must address the trade-off between accuracy and computation time. Progress will require significantly more interaction between information theory, computer science, control and artifice intelligence than ever before [2]–[8].

Received November 27, 2010

A. Communication aspect

The remote-control, which is the basis for today's factory automatization, depends heavily on those communication systems used in connection with all automatization elements, as they are controllers, actuators and sensors within one working system. At the very beginning a variety of specialized, so called industrial data communications systems were developed for these purposes. These developments have been justified because [24]:

1. In the control and automatization, the only acceptable are deterministic communication systems, which guarantee, that a message will be delivered, and for which the upper bound of delivery time can be computed.
2. The latency times of these communication systems had to be low.
3. The industrial communication systems had to be low-cost.

Most of these networks are typically reliable and robust for real-time control purposes.

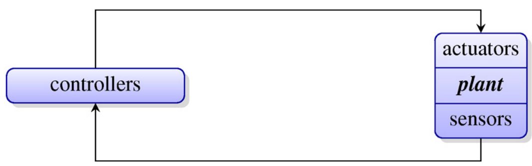
Meanwhile, those technologies concerning general computer networks, especially Ethernet, also progressed very rapidly. With their decreasing prices, increasing speed, widespread usage, numerous software and applications, and well-established infrastructures, these networks have become major competitors of the industrial networks regarding control applications. The development of Networked Control Systems (NCS) has gone off in two directions [24]:

1. improving the features of general communication systems so that they are able to work in real-time, be reliable, secure, etc, and
2. the development of new control paradigms and methodology, which are robust regarding any variation in communication parameters such as delays, jitter or even temporary loss of information.

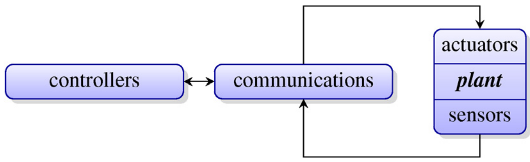
The latter efforts have lead to control applications that can use the Internet in order to perform remote control over much greater distances than in the past, without investing in the whole infrastructure. Although industrial networks have been enhanced for Internet connectivity, the cheaper price and widespread usage of general networks are still attractive for use in control applications.

B. Control aspect

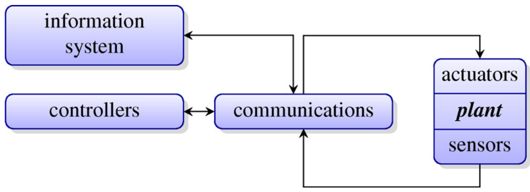
Over recent decades, control systems have evolved from "local`` control systems, where connections between systems' elements in system analysis and synthesis are considered as "ideal``, e.g. with unlimited precision and without any delay (Fig. 1a) rather than digital control with constant delays, which can also incorporate delays introduced using digital communication systems (Fig. 1b) regarding information reaching control systems, which enables the designing of autonomously controlled systems (Fig. 1c).



(a) classical view of control theory



(b) modern view of control theory



(c) information systems as a significant part of modern control

Fig. 1. Evolution of control systems

From the control aspect, regardless of the type of network used, the overall NCS performance is always affected by network delays. Although they may insignificantly affect an open-loop control system such as on-off relay systems in industrial plants, the open-loop control configuration, however may be inadequate for time-sensitive or safety-critical high performance control applications that require feedback data sent across the network in order to correct the output error.

2. NCS CONFIGURATION AND NETWORKS' DELAYS

There are two general NCS configurations listed as follows [9]:

- Direct structure, which is composed of a controller and a remote system containing a physical plant, sensors and actuators directly linked by a data network (Fig. 2).

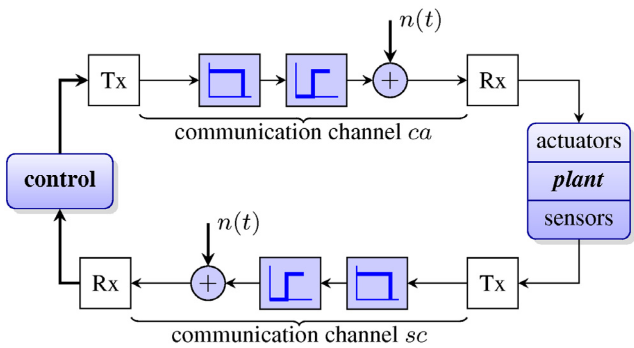


Fig. 2. Direct NCS configuration and network delays for NCS formulations

In a practical implementation of a direct control, multiple controllers can be implemented in a single hardware unit to manage multiple NCS loops in the direct structure.

- Hierarchical structures, which are generic composed from remote system consisting of physical plant sensors and actuators controlled by local control and main control of a remote closed loop system (Fig. 3).

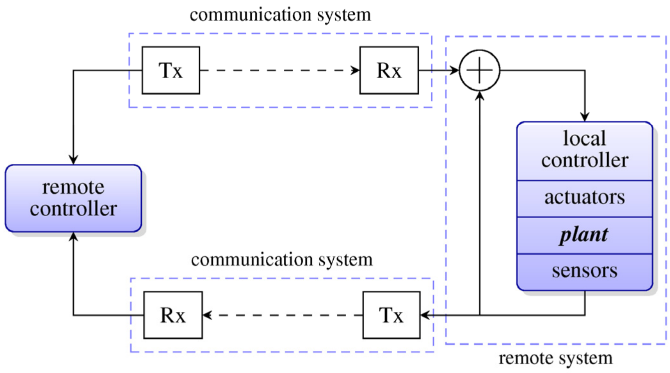


Fig. 3. Hierarchical NCS configuration

In the hierarchically structured NCS, the main controller periodically or randomly, activated by a certain event within the remote plant, computes and sends the reference data via a network to the remote system. The remote system then processes the reference signal in order to perform local closed-loop control and returns the sensor measurement to the main controller. The networked control-loop usually has a longer sampling period than the local control-loop since the remote-controller is supposed to satisfy the reference signal before processing the newly-arrived reference signal.

The use of either the direct structure or the hierarchical structure is based on the application requirements, and the designer's preferences. For example, typical applications of hierarchical control can be found in factory automatization (Fig. 4). Here sensors and actuators are linked to a local controller by specialized hard real-time networks which are, depending on safety demands, event or time driven. The local controllers are linked to supervised systems or central control, usually of some high-speed network such as HSE, EtherCat, Industrial Ethernet, TTE etc, which mainly serve for coordinating controllers usually employed in a certain production cell (Fig. 4).

Semi autonomous systems can be considered as special case of hierarchical control. Their local structures are similar to those depicted in Fig. \ref{fig:1-1c}. The information systems and internally hierarchically-structured control algorithms are components of artificial intelligence which, with the help of plethora additional sensors for sensing the plants behaviour as well as its surroundings, manages and controls such systems. Actually, these local systems take over main controls tasks and leave the remote side mostly supervision functions. Consequently, here communication between local and remote controls is not that critical anymore, since it serves only for exchanging information indirectly by being involved in performing the control algorithms.

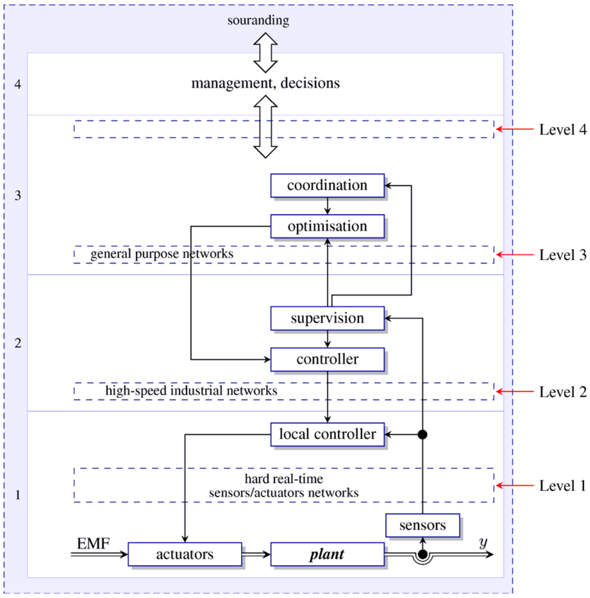


Fig. 4. Model of modern automatization. EMF: energy-mass flow in company production or services

A. Delays in the loop

Network delays in an NCS can be categorized in different ways [9]. For example, if they are constant or they vary stochastically. Regarding the direction of data transfers, they form three categories: sensor-to-controller delay τ_{sc}, controller-to-actuator delay τ_{sa}, and between them the time τ_{cc} needed for control algorithm computation (Fig. 5).

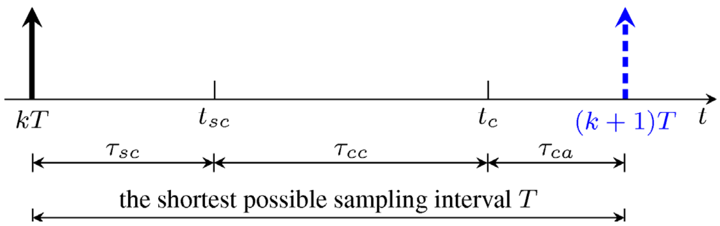


Fig. 5. Delays in NCS: the case of single sensor, actuator, and controller in the loop

During analysis and control synthesis, it is sufficient to know τ_{sc}, τ_{sa}, which can vary at around a certain average vale, and τ_{cc}, which is bounded and usually constant. However, in system and protocol designs, the subintervals of τ_{sc} and τ_{sa} are the pivotal parameters.

B. Delays characteristics

The delay characteristics of NCS basically depend on the type of network. Regarding network accessibility, they are divided into two basic classes [9]:

- Deterministic network. In local area network protocols such as IEEE 802.4 Token Bus, IEEE 802.5 Token ring, PROFIBUS, FIP etc has deterministic media access control, which enables cyclical service, i.e. control and sensory signals are transmitted over a cyclic order with deterministic behaviors. Thus, the delays are periodic and can simply be modeled as a periodic function such that $\tau sc\ (k) = \tau sc\ (k + 1)$ and $\tau ca(k) = \tau ca(k + 1)$
- Random access network. Random access local area networks such as CAN and "classical`` Ethernet involve more uncertain delays. The more significant parts of random network delays concern the waiting time delays due to queuing and frame collision on the networks. When an NCS operates across networks, several more factors can increase the randomness of network delays, such as queuing time delays at a switch or a router, and propagation time delays from different network paths. In addition, a cyclical service network connected to a random access network also results in random delays.

The deterministic network works perfectly in an ideal case; however, in practice it may experience small variations on periodic delays due to several reasons. For example, discrepancies in clock generators on both local and remote systems may result in variation of delays.

Random network delays can be modeled by using various formulations based on probability. These techniques range from simple approaches such as the Poisson process to more sophisticated approaches such as the Markov chain. These techniques have been introduced to NCS formulations from several studies, but may have to be modified or reformulated for specific network control methodologies.

C. Performance degradation due to delays in-the-loop

It is widely known that delays in a control-loop degrade controlled-systems performances. It also has an influence on the plant bandwidth. In general it is limited with the shortest possible sampling interval (Fig. 5).

3. Time Triggered Communications

From among those existing communication systems used for automatization and control, this section provides a brief overview of some currently-available fieldbus networks providing time-triggered features [29], [30].

Real-time communication systems have two major design paradigms:

- event-triggered systems, and
- time-triggered systems.

An event-triggered system follows the principle of reaction on demand. This approach, on the one hand, is well-suited for sporadic actions and data, low-power sleep modes, and best-effort soft real-time systems with high utilization of resources but, on the other hand, does not ideally cope with the demands for predictability, determinism, and guaranteed latencies –- requirements that must be met within a hard real-time system.

Time-triggered systems support precise temporal specification of interfaces and the implementation of "temporal firewalls" to protect error propagation, via control signals. A basic concept in the time-triggered paradigm is the global time. For most real-time applications, it is sufficient to model time according to Newtonian physics without regarding relativistic effects [16]. Clock synchronization approaches for time-triggered systems typically implement concepts of fault tolerance and self-stabilization.

Global-time is used to define those instances when communications and computations of tasks take place within a time-triggered system. The message's length and the message sender are known a priori, according to a predefined message schedule. Computation is realized by the execution of Simple Tasks, that is tasks which cannot be blocked. For each task an a priori-known upper bound for their Worst Case Execution Time (WCET) is assumed [22].

Using a static scheduling algorithm, the tasks and messages are scheduled to form a collision-free communication pattern which guarantees that all tasks can be finished on time before their results are used. Such a pattern forms so-called rounds which are periodically repeated (Fig. 6)

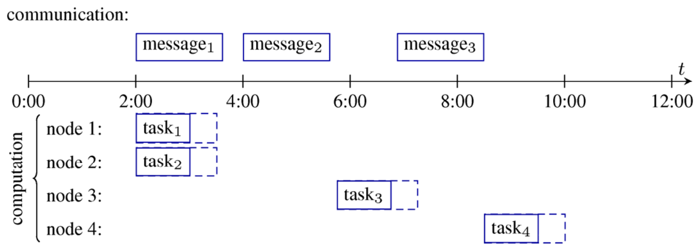


Fig. 6. Time-triggered scheme for communication and computation

For demonstrative reasons, the timeline is denoted to 12 hours, as on the face of an analog clock. The boxes above the timeline represent the planned messages on a shared communication medium. The boxes below the timeline correspond to the execution of local tasks. The upper bound of a task's execution time is denoted by the dotted lines.

A. Advantages and disadvantages of the Time-Triggered Approach

The time-triggered approach has been proven to show the following advantages:

- Low jitter during message transmission and task execution provides special advantages for distributed control loops.
- Its predictable communication scheme simplifies diagnosis of timing failures.
- Its periodically transmitted messages enable short and bounded error detection latency for timing and omission errors.
- Its principle of resource adequacy guarantees a nominative message throughout independent of the network load. Problems such as increasing delays at message floods or thrashing [11] are avoided by its design.
- The time-triggered paradigm avoids bus conflicts using a TDMA scheme, making explicit bus-arbitration obsolete.
- By using a sparse-time base, replica determinism between time-triggered components can be achieved without the need for complex agreement protocols.

A time-triggered system used within an event-triggered system provides several benefits, especially when related to safe critical applications, but there are applications for which an event-triggered approach is better-suited than a strict time-triggered scheme:

- When the major system requirement is low energy consumption over time, as is the case for wireless sensor networks where the event-triggered systems are better situated. However, for a low duty cycle the time-triggered approach can enhance the system's properties and make the most of its lifetime.
- When the average response time of the system is concerned. In this aspect, event-triggered systems may outperform time-triggered systems.
- Time-triggered systems have to plan for an upper bound for the execution time of each task, in contrast to an event-triggered approach, which can work with weaker assumptions such as a global time budget for a set of tasks.
- When it is difficult to fit messages with differing periods into a static schedule. The length of the static schedule is defined by the least common multiple of the message periods, which can lead to a very extensive static schedule causing memory problems within embedded systems.
- In wireless scenarios where a considerable rate of link failures cannot be handled by the standard time-triggered approach.
- When the required precision of the global time, e.g., in state-of-the-art chip design with clock frequencies of several GHZ, where it is impossible.

Most of the problems listed above arise in those applications where non-real-time or soft real-time requirements are prevalent over dependability issues such as reliability and safety.

4. Time-Triggered Fieldbuses: brief overview

A. Time-Triggered Protocol

Time-Triggered Protocol (TTP), Time-Triggered Protocol for SAE class A applications (TTP/A) and Time- Triggered Ethernet (TTE) are protocols belonging to a family of protocols for the time-triggered architecture (TTA). TTP for SAE class C applications (TTP/C) which focus on the interconnection of components in order to form a highly-dependable real-time system that is sufficient for critical applications such as X-by-wire in the automotive and avionics domains. TTP implements a replicated bus system and a guardian that prevents babbling idiot failures [26].

This protocol does not require a central node as time master (as has TTCAN) or bus manager. Instead, the nodes interact at startup to agree on a common synchronized time base that is used to define instants of action and communication within the system. It is used for distributed message scheduling, as well as for local process scheduling. Furthermore, the time base is available to the application as globally synchronized time.

TTP assumes to have a priori defined action and communication patterns. In order to support as well event-triggered legacy systems such as CAN applications, the time-triggered layer of TTP has been enhanced to emulate event messages. The protocol is also fault-tolerant to a single arbitrary faulty node during the start-up phase, as well as during the synchronous operation.

B. TTP/A

TTP/A is a time-triggered master-slave fieldbus system [17]. The master establishes a global time and announces the beginning of a communication round by issuing a "fireworks" message. After the fireworks message, what follows is a common collision-free communication pattern consisting of n predetermined slots (Fig. 7). As in TTP, the schedule and global time is known to all nodes; however, TTP/A supports an on-line update function of the schedule during operation.

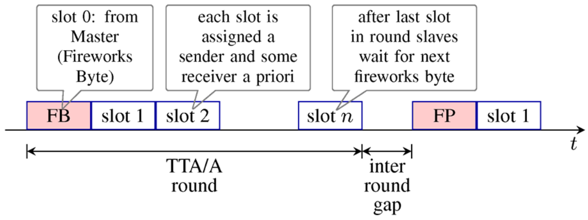


Fig.7. TTP/A communication round

TTP/A was designed to support an easy and economically-feasible integration of sensors and actuators into a real-time network. TTP/A can be implemented in software on low-cost microcontrollers[1] .The interface concept of TTP/A supports a modular design and an easy integration and management of transducers.

In contrast to TTP, TTP/A has no fault-tolerant capabilities able to handle arbitrary node faults, but is more flexible by providing a means of online configuration. The interface implemented by the TTP/A protocol has been standardized with the OMG Smart Transducer Interface [20]. At present, TTP/A supports transmission rates of up to 100 KBit/s.

C. Time-Triggered Ethernet

Today Ethernet is the prevailing communication technology in Local Area Networks (LAN). Its mass use, and consequently low-cost make it attractive for control applications too. Consequently, many attempts have been made to adapt Ethernet to specifics of control applications, i.e. to work in real-time.

The variety of solutions is the consequence of the standardization efforts of CENELEC and IEC to establish one common standard for Ethernet use in real-time, i.e. control application as fieldbuses was unsuccessful (they gave up in the late eighties of the last century) [27].

Real-Time Ethernet (RTE) uses various topologies. For basic topologies standard IEC 61158 [28] defines the so-called communication profiles (Table I).

Table I: Possible RTE Topologies

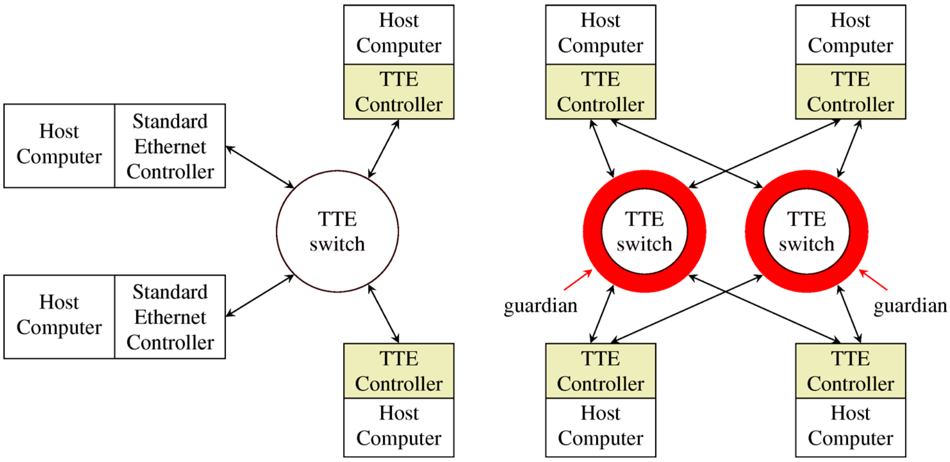
Basic network topology	Communication profile (CP)
Hierarchical star	CP m/1
Ring (loop)	CP m/2
Daisy-chain	CP m/3
Note: a real topology could be any combination of the three basic topologies.	

One RTE is Time-Triggered Ethernet (TTE), which incorporates solutions from TTP to Ethernet technology [15]. It establishes in Ethernet a global synchronized time that is

[1] An implementation of TTP/A for Atmel AVR is available under an Open Source License, see \url{http://www.vmars.tuwien.ac.at/ttpa/.

then used to execute a distributed time-triggered communication scheme. TTE also allows the use of standard Ethernet frames to support event-triggered data, whereas a dedicated TTE-Switch takes care that time-triggered frames are not delayed by other frames. The main difference between TTE and existing real-time Ethernet solutions is the interrupt mechanisms of the TT Ethernet Switch.

TT Ethernet is available for 100 Mb/s and 1 Gb/s, and is intended to support all types of applications, from simple data acquisition systems (Fig. 8a), to multimedia systems up to safety-critical real-time control systems (Fig. 8b).



(a) standard configuration (b) safety critical configuration

Fig. 8. Time-Triggered Ethernet

It also enables different combination of standard, time- triggered as well as safety critical communications (Fig. 9).

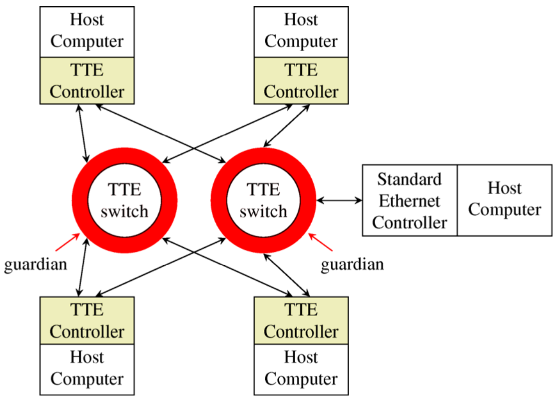


Fig. 9. Time-Triggered Ethernet - safety hybrid configuration

Basically it is designed for the communication profile CP m/1 (Table I) so it can exploit commercial infrastructures for the integration of real-time and non real-time traffic. It is compatible with Ethernet standard. The main difference between TTE and existing real-time Ethernet solutions is the interrupt mechanisms of the TT Ethernet Switch.

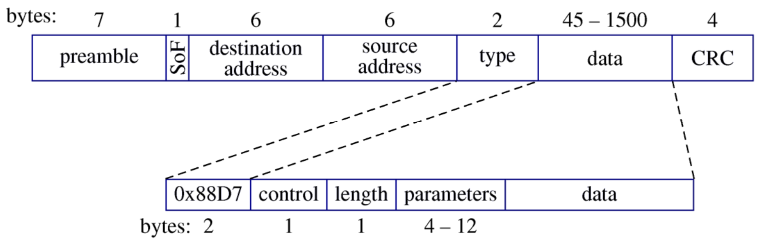


Fig. 10. Time-Triggered Ethernet frame format

For time-triggered application TTE use extended Ethernet frame (Fig. 10) registered at IEEE Registration Authority (http://standards.ieee.org/regauth/ethertype/). For support of the message type, identification and message instance it provides within the following message categories:

1) Event -Triggered (ET) messages
2) Free Form Time-Triggered (FFTT) messages
3) Unprotected Start-up messages
4) Unprotected Synchronization messages
5) Unprotected periodic TT messages
6) Unprotected sporadic TT messages

By using the TDMA scheme, TDMA rounds are divided into time slots for each message (Fig. 11). A message in the fixed TT part has a fixed length. They are sent through two redundant channels regardless of the amount of standard Ethernet traffic, and are protected by the bus guardian. Nodes that suffer from faults within the fault hypothesis cannot affect the transmission of protected TT traffic.

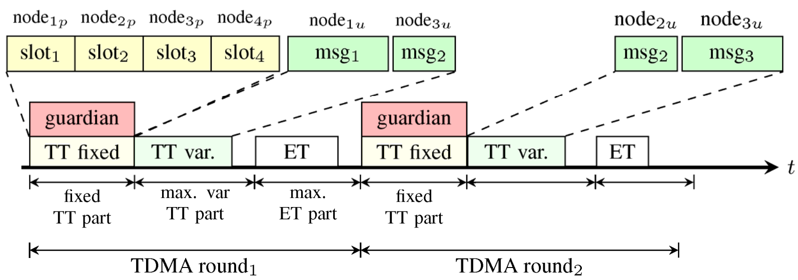


Fig. 11. Example of the TTE communication schedule

D. Flexray

Flexray [13] is a fieldbus system for automotive applications such as X-by-wire. Flexray was developed and is supported by a consortium of automotive manufacturers and suppliers including BMW, Daimler, Chrysler, Volkswagen, Bosch, General Motors, Freescale, and NXP Semiconductors.

The FlexRay protocol is a hybrid protocol consisting of a time-triggered part, where messages are scheduled according to an a priori defined TDMA schedule and a flexible part supporting sporadic traffic. The flexible part has media access based on the Byteflight protocol [21] that uses minis lotting in order to provide a collision-free communication that does not interfere with the time-triggered part.

E. TTCAN

Time-Triggered Controller Area Network (TTCAN) is a time-triggered protocol that builds on the event-triggered CAN protocol [14]. In its extension, TTCAN establishes a global synchronized time derived from periodically broadcasted synchronization frames by a time master node. This synchronized time can be used to program an event-trigger in the application code, thus enabling synchronized actions.

The TTCAN protocol is implemented in hardware using a dedicated TTCAN controller. TTCAN integrates time-triggered frames with standard event-triggered frames. The event-triggered part uses standard CAN arbitration to avoid collisions [25].

F. FoundationTM fieldbus

FOUNDATION^{TM} (FF^{TM}) is an open, fieldbus architecture for plant information integration. It contains a H1 bus running at 31:25 kbit/s, which is optimized for integration of field instrumentation and High Speed Ethernet (HSE) running at 100 Mbit/s intended for the integration of H1 and other control subsystems into a high performance control backbone. The intention of this is to reduce the number of different networks, gateways, and systems in the plant hierarchy whilst at the same time increasing information integration between automation systems, plant application packages, and Management Information Systems (MIS). A cost-effective system can be built on standard, high-volume, low-cost networking technology [23], [25].

FF^{TM} is an industrial fieldbus providing a hybrid approach for transmitting time-triggered and event-triggered data [23]. It is a functional superset of WorldFIP [19]. The concept is implemented by periodic and aperiodic processes. Periodic processes are time-triggered processes initiated at predetermined points in time. Aperiodic processes handle event-triggered traffic that is delivered as soon as possible but with considerable jitter in the message delivery time. Both message types are scheduled on a single bus with a MAC protocol based on centralized arbitration by a bus manager.

A periodic communication processes could be considered a time-triggered protocol, however it has the following properties that are untypical for time-triggered systems: (i) The scheduling table for the periodic data is not provided to the single nodes, (ii) the nodes are unaware of a global time which could be used in the application (iii) The scheduling decisions are not directly based on a global time, but are done by the bus manager (which bases its decision on timing).

G. Comparison of TT based communication systems

The main features of described fieldbuses are collected in Table II In this table, the level of automation (LoA) refers to the concept of field, cell, and management levels in industrial automation.

Table II: Feature comparison of time-triggered buses

Name	Bandwidth	CS	Dependability	TT concept	LoA
TTP/A	100 kb/s	M/S	low	fully	field
TTP/C	25 Mb/s	FTD	high	fully	cell
TTE	100 Mb/s, 1Gb/s	FTD	high	coexistent TT and ET traffic	field or cell
Flexray	10 Mb/s	FTD	high	coexistent TT and ET traffic	field
TTCAN	1 Mb/s	M/S	medium	coexistent TT and ET traffic	field or cell
FF^{TM} H1	31,25 kb/s	M/S	medium	application level	field
Legend:	Legend: CS: Clock Synchronization, LoA: Level of Automation, FTD: Fault-Tolerant Distributed, M/S: Master-Slave				

TTP/C is mainly considered feasible for the cell level when considering cost constraints. TTE might be a promising candidate for the cell level (due to its high speed and dependability) as well as for the field level (because of its expected low cost). A similar prospect exists for TTCAN. At the low-cost end, TTP/A, and FF^{TM} provide time-triggered solutions at the fieldbus level. However, among them only TTP/A can be considered a true time-triggered protocol that fully utilizes the concept of global time and thus supports all four types of applications. It is important to note that only FF^{TM} has considerable market share in the automation domain among these two.

5. Fault-tolerant Actuating

In the time-triggered fault tolerant protocols, for example TTP, TTE, Flexray, TTCAN, the fault hypothesis is mainly focused on network faults such as erroneous and faulty messages. From the applications point of view, also important are data acquisitions, i.e., measurement and actuation. Solutions for fault-tolerant and robust measurement can be found in [9], [11], [17], so there is only a brief overview of fault-tolerant actuation in the context of time-triggered systems.

Time-triggered systems are of special interest for fault-tolerant actuating, because in most cases:

- faulty action cannot be undone at a later instant,
- nonsynchronous execution of correct actions from independent actuators may cause unwanted behavior

For example, let's briefly examine a case of fault-tolerant actuation of three independent actuators, which show low-pass behavior due to physical issues, on a controlled object (Fig. 12). In order to ensure correct behavior even in the case of failure of one of the replicated actuators, it must be ensured that:

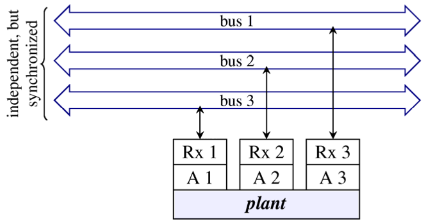


Fig. 12. Fault-tolerant actuating using triple modular redundancy

- control signals are synchronized with a precision better than the cut-off frequency of the low pass element,
- control decisions of the correctly operating components are replica-deterministic.

In order to achieve these requirements, a time-triggered architecture using a precisely synchronized fieldbus is adequate. Furthermore, a closely synchronized action setting is of advantage in order to minimize the mechanical stress created if actuators work against each other.

6. Networked Control Methodologies

Various control methodologies for an NCS have been formulated based on several types of network behaviors. The intention is to maintain the stability of the system in addition to controlling and maintaining the system performance as much as possible.

In the formulating of NCS control, methodologies usually make certain assumptions about network behavior. For example that data transmissions are error-free, all frames have the same constant length, network traffic cannot be overloaded, and that every dimension of the output measurement or the control-signal can be packed into one single frame or packet. Beside these, it is also usually considered that differences in sampling instants between controller and sensors, called time skew Δk and the computational delayτ_c are constant and are much smaller than the controller sampling period T.

The known methodologies can be grouped in the following classes [9]:

- Deterministic discrete-time model methodology
- Queuing methodology
- Optimal stochastic control methodology
- Perturbation methodology
- Sampling time scheduling methodology
- Robust control methodology
- Fuzzy logic modulation methodology
- Event-based methodology
- End-user control adaptation methodology
- Hierarchical methodology
- A controller architecture

Out of the aforementioned methods, only some are briefly summarised here.

A. Queuing methodology

These methodologies have been developed by utilizing some deterministic or probabilistic information from an NCS for control algorithm formulation. As a basis they use queuing mechanisms for reshaping random network delays on an NCS into deterministic delays, such that the NCS becomes time-invariant.

B. Deterministic queuing

This methodology uses an observer to estimate the plant states and a predictor to compute the predictive control based on past output measurements. The control and past output measurements are stored in FIFO (First-In-First-Out) registers defined as Q_A and Q_S (A as actuator, S as sensor); where their sizes are μ and θ respectively (Fig. 13).

- From the set of past measurements $z(k) = \{y(k - \Phi), y(k - \Phi + 1),\ldots\}$ where Φ is the number of packets in Q_S, the observer estimates the plant state $\hat{x}(k-\theta+1)$.
- From the estimated plant states the predictor calculates the future state $\hat{x}(k+\mu)$ for input to the controller.
- The controller computes from $\hat{x}(k+\mu)$ the predictive control $u(k + \mu)$ and sends it to be stored in Q_A

Since the performances of the observer and the predictor highly depend on the model's accuracy, the dynamic model of the plant has to be very precise.

C. Probabilistic queuing

Another queuing approach is probabilistic predictor-based delay compensation methodology, which utilizes probabilistic information along with the number of packets in a queue to improve state prediction (Fig. 14).

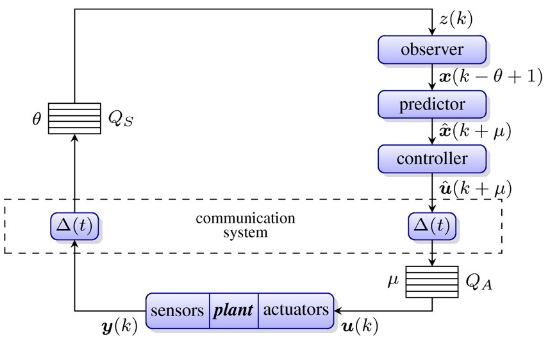


Fig. 14. Configuration of the deterministic predictor-based delay compensation methodology

Here the sensor sends output $y(k)$ when the network is available for a transmission. In the Q_A are stored $y(k)$, $y(k - 0.9),\ldots, y(k - \mu)$. If at sampling instant k the sensor cannot send $y(k)$, than values in Q_S are shifted so that in place for new $y(k)$ is set $\omega(k)$ with values of the last received y, i.e. the $\omega(k)$ can be identical to any value in $y(k)$, $y(k-0.9)$,…, $y(k-\mu)$. However, the possible choices of $\omega(k)$ require that delay index i has to be known. This condition requires that the value of i has to be attached to each packet of $y(k)$. The predictor then estimates the current state $\hat{x}(k)$ by the help of weighting matrices P_0 and P_1, which are computed from the probabilities of the occurrences of $y(k-i)$.

D. Sampling time scheduling

With this methodology it is possible to select a sampling period T_1 for an NCS such that network delays do not significantly affect the control system's performance, hence they remain stable. This methodology has been developed for networks where delays are known in advance. However, it can be adapted to random delay networks, too.

This methodology is based on the assumption that the dynamics of the most sensitive NCS, further denoted as NCS_1, is much slower than the network can provide. In such a case, a sampling interval T_1 of NCS_1 can be transferred for example r messages (Fig. 15), i.e. it can be computed by:

$$T_1 = r(\varphi_1 + L), 3 \le r \le M \tag{1}$$

where r is the number of data messages that can be served by the network during the worst-case network traffic and M number of NCS served by one network. In order to find the sampling times of other NCS's on the same network, their sampling intervals have to be scheduled from the worst-case delay bounds of the systems in an ascending order as $T_2,\ldots, T_M$. In a generic case, they are multiples of T_1:

$$T_i = k_i T_1, i = 2,3,\ldots.M \tag{2}$$

$$k_i = \Lambda\left(\frac{\varphi_i + (T_1 - L)}{2T_i}\right) \tag{3}$$

where T_i is the sampling time of NCS_i, and $a = \Lambda(b)$ indicates that $a = 2^{\nu_i}$; $\nu_i \in \{0,1,2,\ldots\}$; which is the "closest" to, but does not exceed b.

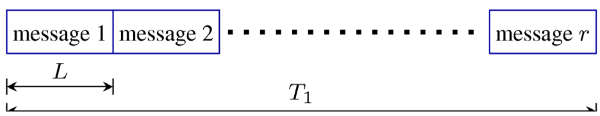


Fig. 15. Windows of data transmissions in the sampling period T_1 of the sampling time scheduling methodology

In a special case, when $M \le r$, the sampling intervals of NCS_2,….,NCS_M are determined by:

$$T_1 = \left(\frac{\varphi_i + (T_1 - L)}{2}\right), \quad i = 2,3,\ldots \tag{4}$$

This methodology can also give optimal network utilization. The condition for optimality is:

$$2\sum_{i=1}^{M}\frac{T_1}{T_i}=r \tag{5}$$

E. Robust control methodology

A major advantage of robust control methodology is that it does not require a priori information about the probability distributions of network delays. The network delays τ_{ca} and τ_{sc} are assumed to be bounded and able to be approximated by the fluid flow model as follows:

$$\begin{aligned}\tau_n &= \frac{1}{2}(\tau_{\max}+\tau_{\min})+\frac{1}{2}(\tau_{\max}+\tau_{\min})\delta, \quad 1\le\delta\le 1\\ &(1-\alpha)\tau_{\max}+\alpha\tau_{\max}\delta, \quad 1\le\delta\le\frac{1}{2}\end{aligned} \tag{6}$$

where τ_n can be τ_{sc} and τ_{ca} with upper bound $\tau_{\max}$ and lower bound $\tau_{\min}$, and α and δ are real numbers depends on an application. The first term in (6) represents a constant part of the whole delay, whereas the second term represents the uncertain part delay variation around the first term.

In controller design in frequency space the delay τ_n can be approximated by the first-order Padé approximation as:

$$e^{-\tau_n s}=e^{-s(1-\alpha)\tau_{\max}}e^{-s\alpha\tau_{\max}\delta}\approx\frac{1-s\tau^n/2}{1+\tau^n/2}\approx\frac{1-s(1-\alpha)\tau_{\max}\delta/2}{1-s(1+\alpha)\tau_{\max}\delta/2}\cdot\frac{1-s\alpha\tau_{\max}\delta/2}{1+s\alpha\tau_{\max}\delta/2} \tag{7}$$

The uncertain delay part is then treated as the simultaneous multiplicative perturbation:

$$\frac{1-s\alpha\tau_{\max}\delta/2}{1+s\alpha\tau_{\max}\delta/2}=1+W_m(s)\Delta \tag{8}$$

where Δ is the perturbation function, and

$$W_m(t)=\frac{\alpha\tau_{\max}s}{1+\alpha\tau_{\max}s/3.465} \tag{9}$$

is a multiplicative uncertainty weight which covers the uncertain delay. The factor 3.465 is selected based on the designer's preference. In the next design step the above formulation is put in H_∞ framework, and μ-synthesis is used to design a continuous time controller $G_C(s)$ for a plant $G_P(s)$, where $R(s)$, $U(s)$, $Y(s)$, and $E(s) = R(s)-Y(s)$ are the reference, control, output, and error signals in the frequency domain respectively (Fig. 16). Ultimately the controller is discretized using bi-linear transformation.

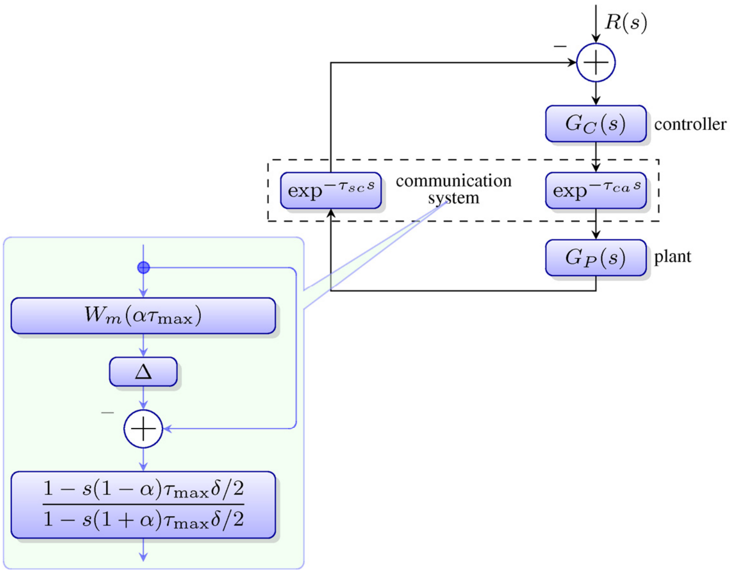


Fig. 16. Configuration of NCS in the robust control methodology

F. Event-based methodology

This methodology was originally developed for a hierarchical structure, but it can be applied for a direct structure as well. Its concept is quite different from all the previously mentioned methodologies. Instead of using time, this methodology uses system motion as the reference of the system. The motion reference, defined as s can be, for example, the path of a robotic manipulator. In order to guarantee the system's stability, s has to be a nondecreasing function of time.

The sensor output $y(t)$, sent across a network, is used as an input for a motion reference mapping (Fig. 17).

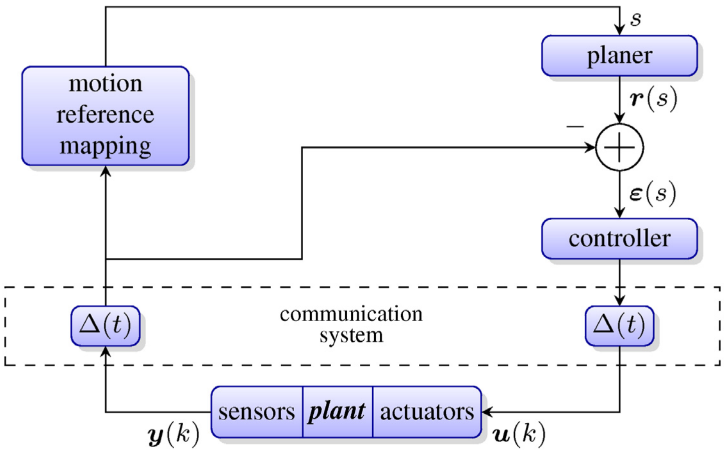


Fig. 17. Configuration of NCS in the event-based control methodology

The mapping converts $y(t)$ to the motion reference s, which is then used as the input for the planner to compute the reference $\boldsymbol{r}(t)$. Thus, $\boldsymbol{r}(t)$ becomes a function of $y(t)$, and it is updated in real-time to compensate all disturbances and unexpected events. Because the overall system is not based on time, network delays cannot destabilize the system.

G. End-user control adaptation methodology

The main concept of end-user control adaptation is to adapt controller parameters with respect to the current network traffic condition or to the current given network Quality-of-Service (QoS). For this adaptation the local as well as the remote system's communication part has to be able to measure network traffic conditions and negotiate for QoS in real-time. If the desired QoS cannot be granted, the controller will adapt the control parameters to aim for the best possible performance.

7. Conclusion

This paper gives a brief survey of the fundamentals and recent developments to control and automatization dedicated communication and research results and developments of control methodologies used in NCS. An NCS can be designed as direct structures, mostly used in special circumstances, or as hierarchical structures, commonly used in factory automatization. The selection depends on the application requirements and the designer's preferences.

It can be seen that a joint effort is needed for the further development of NCS. There are noticeable trends in the researching and developing autonomous systems using integration of control, information, and telecommunication theories.

References

1. Žarko Čučej, Dušan Gleich: Networked Control Systems (invited paper). X Triennial International Conference on Systems, Automatic Control and Measurements - SAUM. Niš, Serbia, November 10. - 12. 11. 2010

A. Control issues

2. Refaat Y. Al Ashi and prof. Abdulla Ismail: Network Control Systems: Performance Analysis and Design. UAE University, Department of Electrical Engineering. http://www.uaesocietyofengineers.com/downloadarticle.asp.
3. Christopher Auburn, Dominique Sater, Joseph Yame: Fault Diagnosis of Networked Control Systems. Int. J. Appl. Math. Comput. Sci., Vol. 18, No. 4, 527–537, 2008. http://matwbn.icm.edu.pl/ksiazki/amc/amc18/amc1849.pdf
4. S. Hart, N. Vozdolsky, T. E. Djaferis: A Class of Networked Control Systems: Architecture, Design and Implementation. IEEE Conference on Decision and Control, 2002 http://www-robotics.cs.umass.edu/~shart/publications/cdc02.pdf
5. Feng-Li Lian, James R. Moyne,Dawn M. Tilbury: Performance Evaluation of Control Network for Manufacturing Systems. Proceedingd of the ASME, Dynamic Systems and Control Devision, SC.Vol.67, Nov. 14-19, 1999. Nashwille, Tennessee http://www-personal.umich.edu/~tilbury/papers/lmt99asme.pdf
6. Xiangheng Liu and Andrea Goldsmith: Fault detection of networked control systems with missing measurements. Stanford University, Dept. of Electrical Engineering, Stanford, CA 94305-9515, USA http://wsl.stanford.edu/Publications/Xiangheng/acc_final.pdf

7. Richard M. Murray (editor): An Introduction to Networked Control Systems. Draft v1.0, California Instiue of technology. March 2006. http://www.cds.caltech.edu/~murray/wiki/images/3/34/Ncsbook-introduction.pdf
8. Sekhar Chandra Takikonda: Control Under Communication Constraints. Ph.D. thesis. Massachusetts Institute Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA 94305-9515, USA http://dspace.mit.edu/bitstream/handle/1721.1/16755/48245028.pdf
9. Yodyium Tipsuwan, Mo-Yuen Chow: Control methodologies in networked control systems. Control Engineering Practice 11 (2003), 1099-–1111, http://www4.ncsu.edu/~chow/Publication_folder/Journal_paper_folder/2003_NBC_Tutorial_Yod.pdf

B. Communication issues

10. P. Chew and K. Marzullo. Masking failures of multidimensional sensors. In Proceedings of the 10th Symposium on Reliable Distributed Systems, pages 32–41, Pisa, Italy, Oct. 1991.
11. P. J. Denning. Thrashing: Its causes and prevention. In Proceedings AFIPS Fall Joint Computer Conference, volume 33, pages 915–922, 1968.
12. W. Elmenreich. Fusion of continuous-valued sensor measurements using confidence-weighted averaging. Journal of Vibration and Control, 13(9-10):1303–1312, 2007.
13. Flexray Consortium. FlexRay Communications System Protocol Specification Version 2.1, 2005. Available at http: www.flexray.com
14. T. Führer, B. Müller, W. Dieterle, F. Hartwich, R. Hugel, and M. Walther. Time triggered communication on CAN (Time Triggered CAN–TTCAN). In 7th international CAN Conference, 2000.
15. H. Kopetz, A. Ademaj, P. Grillinger, and K. Steinhammer. The Time-Triggered Ethernet (TTE) design. In Proceedings of the 8th International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC), pages 22–33, Seattle, WA, USA, May 2005.
16. H. Kopetz and N. Suri. Compositional design of RT systems: A conceptual basis for specification of linking interfaces. Research Report 37/2002, Technische Universität Wien, Institut für Technische Informatik, Vienna, Austria, 2002.
17. H. Kopetz et al. Specification of the TTP/A protocol. Research Report 61/2002, Technische Universität Wien, Institut für Technische Informatik, Vienna, Austria, Sept. 2002. Version 2.00.
18. K. Marzullo. Tolerating failures of continuous-valued sensors. ACM Transactions on Computer Systems, 8(4):284–304, Nov. 1990.
19. P. Noury. WorldFIP, IEC 61158 and the internet: A new look at fieldbuses, 1999. Available at http: www.worldfip.org/noury02.html
20. Object Management Group (OMG). Smart Transducers Interface V1.0, Jan. 2003. Specification available at http:==doc.omg.org/formal/2003-01-01 as document formal/2003-01-01.
21. M. Peller, J. Berwanger, and R. Giessbach. Byteflight specification draft version 0.5. Technical report, BMW AG Munich, 1999.
22. Puschner and A. Burns. A review of worst-case executiontime analysis. Journal of Real-Time Systems, 18(2/3):115–128, May 2000.
23. Z. Wang, Z. Yue, K. Chen, Y. Song, and Y. Sun. Realtime characteristic of ff-like centralized control fieldbus and its state-of-art. In Proceedings of the IEEE International Symposium On Industrial Electronics, pages 140–145, 2002.
24. Žarko Čučej: Komuniakcije v sistemih daljinskega vodenja. UM-FERI Maribor, 1998 (in Slovene)
25. Žarko Čučej and Karl Benkič. Fieldbuses in Manufacturing Automation. In 9th Triennial International SAUM Conference, 2007.
26. Wilfried Elmenreich: Time-Triggered Fieldbus Networks – State of the Art and Future Applications. Proceeding of IEEE Symphosium on Object Oriented Real-Time Distributed Computing (ISORC'08), Orlando, FL, USA, May 2008 http://osg.informatik.tu-chemnitz.de/lehre/old/ws0809/sem/online/ttp.pdf
27. Max Felser: Real-Time Ethernet – Industry Prospectives. PROCEEDINGS OF THE IEEE, VOL. 93, NO.6, JUNE 2005, http://www.felser.ch/download/FE-TR-0507.pdf
28. International Electrotechnical Commission, IEC 61158, Digital data¸communications for measurement and control - Fieldbus for use in industrial control systems, 2003.
29. Kevin Hipp: Seminar Paper: Introduction to TTP - Time-Triggered protocol. Chemnitz University of Technology, Februar 2009 http://osg.informatik.tu-chemnitz.de/lehre/old/ws0809/sem/online/ttp.pdf
30. Klaus Steinhammer, Petr Grillinger, Astrit Ademaj, Hermann Kopetz: A Time-Triggered Ethernet (TTE) Switch. Vienna University of Technology, 3-9810801-0-6 June 2006, http://www.date-conference.com/proceedings/PAPERS/2006/DATE06/PDFFILES/07E_1.PDF

UMREŽENI UPRAVLJAČKI SISTEMI

Žarko Čučej, Dušan Gleich

U proteklim godinama, upotreba mreže podataka u kontrolnoj petlji privlači sve veću pažnju zbog isplative i fleksibilne primene. Jedan od najvećih izazova iz ugla upravljanja je efekat kašnjenja izazvan umrežavanjem u kontrolnoj petlji kao i, iz ugla komunikacije, razvoj pouzdanih mreža koje smanjuju kašnjenje i garantuju pouzdanost. Ovaj rad sadrži kratak pregled skorašnjih napredaka u pogledu mrežnih protokola za automatizaciju i metodologija mrežnog upravljanja.

Ključne reči: *upravljanje, komunikacije, sistemi*