FAKULTÄT FÜR ELEKTROTECHNIK UND INFORMATIONSTECHNIK

INSTITUT FÜR AUTOMATISIERUNGSTECHNIK

# Time as non-functional requirement in distributed control systems


Thomas Hadlich[1], Stephan Höme[1], Christian Diedrich[1], Karin Eckert[2], Timo Frank[3], Alexander Fay[2], Birgit Vogel-Heuser[3]

[1] Otto von Guericke University Magdeburg, Magdeburg
Institute for Automation
Thomas.Hadlich@ovgu.de, Stephan.Hoeme@ovgu.de
Christian.Diedrich@ovgu.de

[2] Helmut Schmidt University, Hamburg
Department of Mechanical Engineering
Institute of Automation Technology
karin.eckert@hsu-hh.de
alexander.fay@hsu-hh.de

[3] Technische Universität München,
Institute of Automation and Information Systems (AIS)
frank@ais.mw.tum.de
vogel-heuser@ais.mw.tum.de


## Technical Report


**IFAT-LIA 1/2012**

## 06. September 2012



Universität Magdeburg
Fakultät für Elektrotechnik und Informationstechnik
Institut für Automatisierungstechnik
Postfach 4120, D-39016 Magdeburg
Germany

## Abstract

*When designing a distributed control system consisting of heterogeneous components whose properties are usually not well known, it is difficult or even impossible to use an analytical approach to verify the fulfillment of all requirements. In this contribution, it is shown (at the example of different implementation scenarios for a typical control function) that detailed knowledge about the communication system is required to reliably assess and compare different designs of distributed systems with regard to a real-time characteristic. However, automation engineers typically can neither be expected to have such detailed knowledge about the internal details of computing and communication, nor do they have the time to study different design alternatives in detail. This motivates further work of the authors towards better engineering support for the design of distributed automation systems, by providing helpful advice regarding design decisions*

## 1. Introduction

Modern trends in manufacturing are defined by mass customization, small lot sizes (down to lot size one), which require often changes in the physical layout of the plant, including enlargement and technical updates, high variability of product types, and a changing product portfolio during the life cycle of the plant [3]. These trends imply more complex products which change faster and which need to be introduced faster, and also a volatile production output and reduced investments in production systems [5].

Different designs for manufacturing systems have been proposed to fulfill these requirements, including Holonic Manufacturing Systems (HMS) [3], [2], [5], Bionic Manufacturing Systems (BMS) [2], Reconfigurable Manufacturing Systems (RMS) [2], and Flexible Manufacturing Systems (FMS) [2], [6]. Mobile software agents [3], [4] constitute a more recent proposal. Although these system designs differ significantly in their approach to achieve the required flexibility in manufacturing, their common denominators are the spatial distribution of control intelligence onto separate controllers or so-called “intelligent devices” (henceforth called “nodes” in this manuscript), their (partial) autonomy regarding control decisions, and their cooperation, which requires means to communicate with each other. This communication is implemented by means of industrial network infrastructure and industrial communication protocols. In a spatially distributed control system, it is necessary to communicate the data flow as well as the control flow between different parts of the control application. That is why the requirements of data flow and control flow define the requirements to the communication system such as jitter or end-to-end-delay time between sensory input and actuator output.

Usually, research projects regarding FMS, RMS, HMS or multi-agent systems define certain properties of the nodes employed, and use specially designed-for-purpose hardware nodes and operating systems. Furthermore, in these research projects those who design the system structure are assumed to have detailed knowledge regarding the properties and the functionality of the nodes. In contrast, the research work presented in this manuscript targets at the engineering of distributed industrial control systems on the basis of existing components and devices, which are already widely employed in industrial applications today, and aims at the support of engineers who have only partial or superficial knowledge about the technical details of the nodes and the communication technology employed (this holds for many practitioners). This includes the aspects of distribution and communication.

Contributions discussing industrial communication systems often focus either on the quality and stability of the control loop and related non-functional requirements [8], [9], [7]. [8] discusses an evaluation of timing behavior on networked control systems with the focus on comparing different types of communication behavior (time- / event-based). The contributions [9] and [7] emphasize the importance of timing on the function of complex automation systems, but focuses on a formal model for communication networks. The work presented in this contribution targets automation engineering requirements, which must be satisfied in the process of the engineering process. For instance it should be possible to select appropriate components and communication protocols based on the integration of the communication model into models of distributed applications.

To show the necessity of a systematic method which supports the engineer during the design of distributed control systems, this paper discusses one non-functional requirement – timing – to show the tight relationship between design decisions regarding distribution alternatives, communication properties, and timing results. Section 3 describes therefore an example based on a thermo-hydraulic press. Two scenarios are derived from this example. In Section 4 measurements show the need for support in the engineering of distributed control systems. At the end of this document we provide conclusions.

Vogel-Heuser et al. [15] present a notation for modeling communication networks in automation engineering intuitively and independently from suppliers. This notation provides specification of automation architectures and imposition of real-time requirements the communication network has to meet. In this paper, the notation presented in [15] is used for the application example.

In order to fulfill the functional and non-functional requirements the spatially distributed control nodes need to communicate with each other. The expansion of the automation system leads to distributed control systems which execute automation tasks on different components and are connected together by a communication system. However the physical requirements, foremost the dynamics remain the same. Therefore the total reaction time from sensor to actuators has to be the main focus. This contribution therefore draws the focus from the definition and description of non-functional requirements (see 2) over the potential automation system architecture related to the communication system (see 2) and at least to timeliness of the end-to-end-transmission time influenced by the communication. It does not cover control loop design and steady state investigations.

## 2. Distributed control Architecture

When designing central or distributed systems it is a crucial success factor to consider non-functional requirements. Some specific non-functional requirements can be fulfilled more successfully by distributed systems than by central systems, for instance *reliability*. Besides reliability, there are other non-functional requirements that are harder to fulfill with distributed systems, e.g. time behavior. Time behavior depends on response time and processing time as well as on the transfer rate of the system of sensor - control - actuator. In distributed systems the response time increases due to the additional communication effort. In the design of distributed systems it is important that the real-time conditions must be strictly adhered.

In order to model the real-time conditions of distributed systems an integrated model of distributed application and communication has been presented in [12]. In the model the distributed control application is split into several parts (function blocks (FBs)) (similar to [1]), which are executed in application processes. The intention of the model is not to propose a specific implementation but to show the distributed nature of the application and the related communication system elements. Kernel of the model for distributed automation systems is the mapping of the application (and its parts) to a set of communicating application processes (see Figure 1).

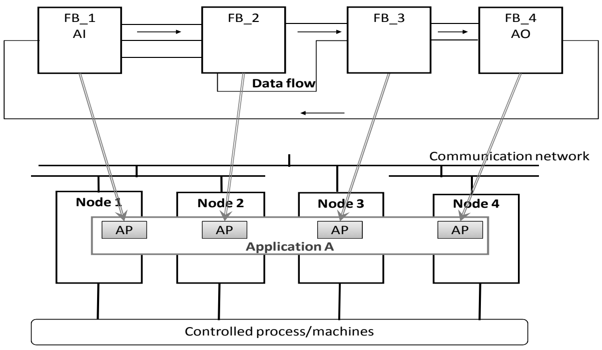



**Figure 1 – Mapping of the application parts to a set of application processes**

Each FB or each FB network segment is allocated to the functional part of one application-process(AP) – see Figure 2. The AP model is based on the definitions in [HDE11], which suggests to integrate both the IEC 61158 AP model with the function block model as described in the following.

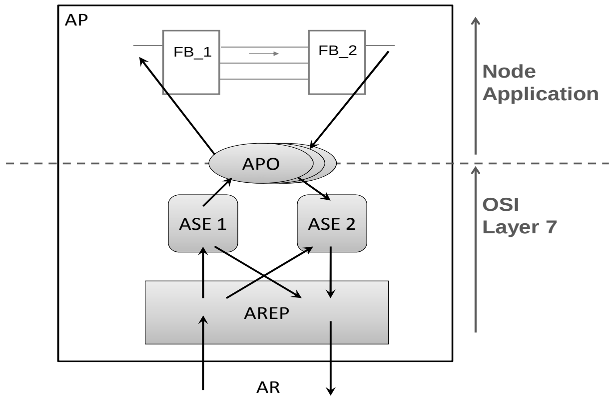



**Figure 2 – Model of an application process**

The FB input and output data are mapped to the application-process-objects (APOs) of the AP. The APO representing a real data object indicates mainly the data type, access rights (e.g. read/write or read only), communication related address and the implementation specific access to the real data.

The application-service-element-types (ASE types) which are interacting with the APOs determine the allowed services for accessing the APOs. For example: if FB input/output data can be read or written, then the IO-data-ASE or the Parameter-ASE has access to the APO. If one data object is read only and a write service tries to write data to it, then the APO denies the access.

The Parameter-ASE usually provides confirmed read services and write services based on a client server model. These services need a connection oriented AR and provide the according role at the application-relationship-end-points (AREPs). The ARs determine data transport properties. Properties of an AR are for example cyclic/acyclic transport, buffered/queued services, client/server or producer/consumer model, connectionless/connection-oriented services, transmission delay, etc. The ASE, APO and AR types together provide the communication roles which are offered by the used communication system. The features of these elements have strong influences to the fulfillment of the requirements.

In one AP there can be multiple ASE instances from one or more ASE types as well as multiple AR instances from different AR types. From the application point of view this means one AP can have communication relations to multiple other APs, which may use the same or different communication services. One device can offer multiple APs. A summary of the consequences of this model are that the data connections between FBs at application level becomes communication parameters which are conveyed over networks. The conveyance paths have extra properties which are determined by communication system specific roles. These extra properties are coming up in distributed applications. The communication systems details are defined in ASE, APO and AR type definitions.

## 3. Application Example for end-to-end-transmission time

### 3.1. Introduction of the application example

The application example is part of a continuous thermo-hydraulic press for wood-based products. It is composed of two co-rotating conveyers which can be controlled separately. Between upper and lower conveyer the raw mixture is compressed. The conveyers can be controlled in speed. To avoid material defects by shearing stress the upper and lower conveyer have to be synchronized regarding speed. To realize the synchronization the two local conveyer control loops are integrated in a “global” speed control loop. The different control loops are shown in Figure 3. The local control loop regulates the speed and the global control loop synchronizes the speed of two conveyor belts.

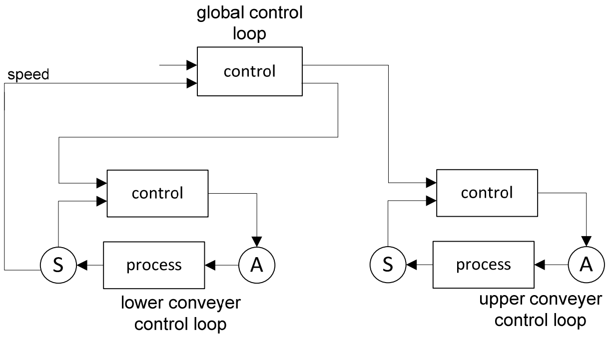



**Figure 3 – Local and global control loops ([14])**

The design goal is to implement a maximum speed deviation of 0.5% at a conveyer speed of 2000 mm/s, this requires a response time smaller than 6ms. In the following discussion we consider only input and output signals that are connected via industrial communication. The complete control application shown in Figure 3 may be allocated to one control node or to two control nodes. In the following some measurements are presented to show the importance of details in distribution and implementation to meet timing requirements. The scenarios are introduced in the following.

### 3.2. Introduction of the used schemes for the visualization of the example

In this example we are considering the network topology, the used communication services as well as the timing behavior of the involved system elements. Therefore the example is depicted with appropriate diagrams for each scenario.

The network topology is shown in Figure 4, Figure 6 and Figure 11. The notation for these figures was introduced in [15]. The sensors and actuators are represented as I/O devices and nodes in a PLC. Since the devices are connected using Ethernet, the switches are represented by four single small rhombuses forming another large rhombus. The number of ports can be indicated in the lower rhombus. If a device hosts an Ethernet switch, this is indicated by integrating the switch symbol into the device. Properties resulting from requirements are placed in the scheme as additional box to indicate its context in the system. In our example such a property is the end-to-end-transmission-time.

The logical view provides the model as defined in [12]. Such views are shown in Figure 5 and Figure 7. One can follow the transmission paths from sensor to actuator, i.e. the end-to-end-path.

### 3.3. Scenario 1 - Decentralized control hardware structure

The Decentral Scenario is characterized by the fact that the control function, consisting of lower conveyer control, upper conveyer control and global control loops, is allocated to one node and that the input and output devices communicate with the control node via the communication system.

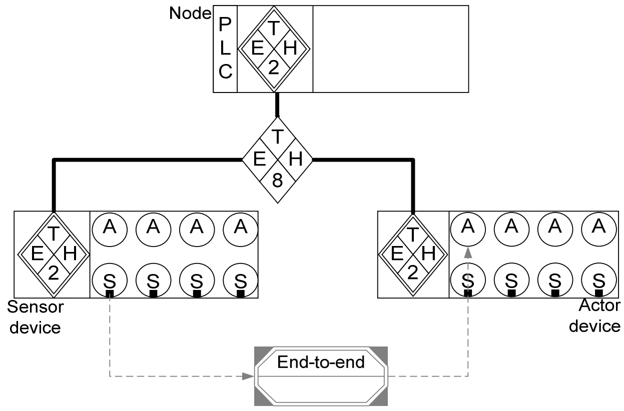



**Figure 4 – Decentral scenario: hardware structure**

In such a scenario the signal and control value update cycles of each individual device have to be considered. The input value is acquired from the sensor and transported cyclically (input cycle) as APO from the sensor to the control node.

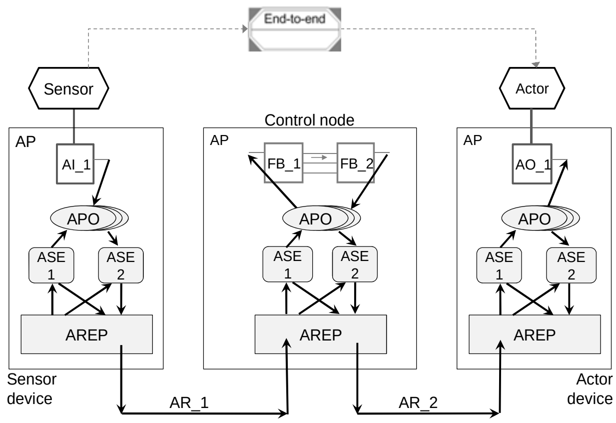



**Figure 5 – Integrated model of application and communication for Decentral scenario**

From the sensor the value is transported via the industrial communication system to the control node. This transport occurs cyclically or acyclically depending on the type of the application relationship (AR type) between control application and input device (AR_1 in Figure 5). The control application takes over the value. Then the control application processes the input values and calculates the output values. The output values then are transported via the communication system (AR_2 in Figure 5) to the output device. The value then is transferred over the APO to the actuator application (cyclically).

### 3.4. Scenario 2 - Distributed control hardware structure

If the control function is shared between two control nodes, for instance if lower conveyer control is executed on Node1 and upper conveyer control and global control loops are executed on Node2, then the input device communicates with one control node (Node 1) and the output device communicates with another control node (Node 2). Node 1 and Node 2 communicate with each other in order to implement the shared control function.

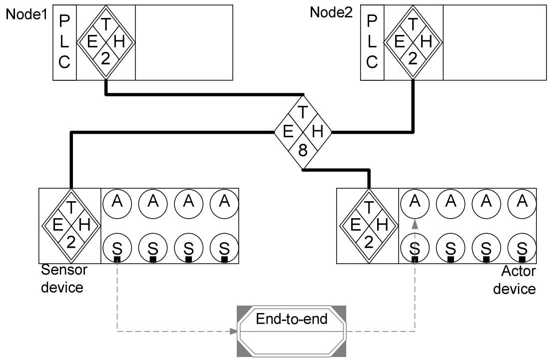



**Figure 6 – Distributed Scenario: hardware structure**

In such a scenario the same transport delays have to be considered like in the scenario with one control node. Additionally the transport between the control nodes (Node 1 and Node2) (AR_3) and the additional processing in Node 2 has to be considered.

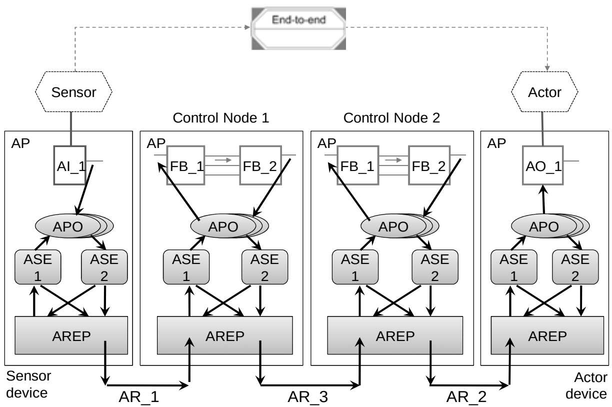



**Figure 7 – Integrated model of application and communication for Distributed Scenario**

The signals must pass the communication stacks of two control nodes. This certainly introduces additional delay. On the other hand, the processing effort in Node1 may be reduced, because some functions are moved to Node2 (see Figure 7). In the following section we research how the properties of the application relationships (AR_1, AR_2 and AR_3) may influence this delay.

## 4. Measurements of end-to-end-transmission-time

In order to underline the influence of AR properties the end-to-end-transmission-time was measured for following arrangements:

- Decentral Scenario (1 input, 1 control node, 1 output) → see Measurement 1 (see 4.1).
- Distributed Scenario (1 input, 2 control nodes, 1 output), with
  - Measurement 2(see 4.2): AR between the control nodes is based on a proprietary PLC-to-PLC protocol (connectionless, buffered, acyclic used by Parameter-ASE, Client/Server)
  - Measurement 3 (see 4.3): AR between the control nodes is based on same protocol like the communication with the IOs, (connection-oriented, buffered, cyclic used by I/O data ASE, Client/Server)
  - Measurement 4 (see 4.4): AR between all elements is based on same protocol but uses isochronous mechanism for communication between control nodes and the IOs, (connection-oriented, buffered used by cyclic, synchronous, I/O data ASE, Client/Server)

In all measurements an analogue input and an analogue output device was used. The analogue input was triggered with a step signal and the time was measured until the same signal was send out at the output device. The 'processing' in the control node was reduced to assigning the input value to the output value.

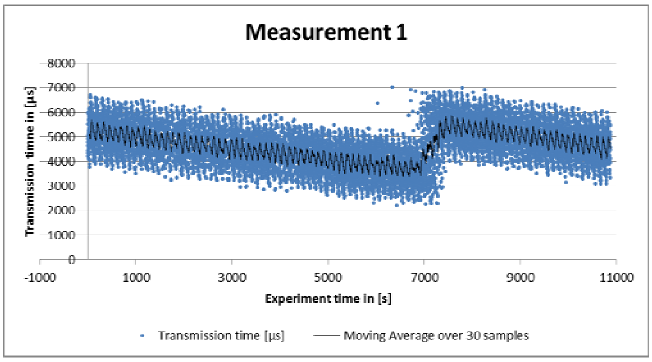
The estimated precision of the time measurements is 1 µs.

### 4.1. Measurement 1

Measurement 1 serves as a reference allowing to compare the results for a single controller with the results measured for a distributed control system (measurements 2 to 4).

Communication for AR_1 and AR_2 is based on IO application relationship (8.3.10.2.2 in [13]), this AR is using a Buffer-Buffer-Unconfirmed-Unidirectional (BBUU) communication relationship, which is characterized by properties like: connection-oriented, cyclic, producer/consumer.

Figure 8 shows the transmission time observed over a duration of approximately 3 hours. When observing the end-to-end-transmission-time, it is possible to observe 2 cycles in the moving average of the transmission time. The major cycle is about 8000 seconds long. During this cycle the medium transmission time is changing slowly from about 5,5 milliseconds to about 4 milliseconds, then the mean transmission time jumps back to around 5,5 milliseconds. The minor cycle is about 125 seconds long, where the medium transmission time is changing within a range of 700 microseconds.




End-to-end-transmission-time:
Mean value: 4625µs; Minimal value: 2210µs;
Maximal value: 7022µs; Standard deviation: 842µs

**Figure 8 –Decentral scenario: end-to-end-transmission-time**

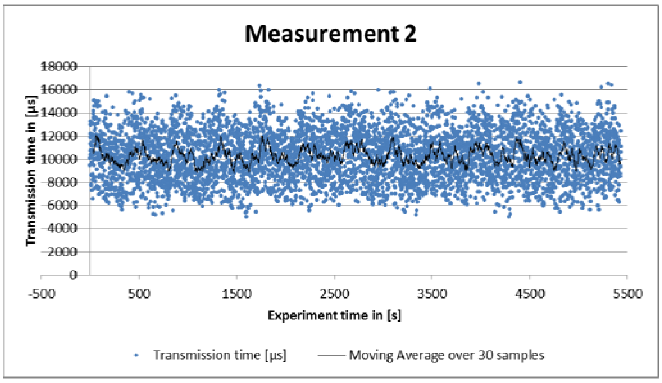
These 2 cycles are caused by slips in the different clock cycles of the system. The major cycle most likely is caused by the slip of the cycle of the output device in regard to the cycle of the input device. The cycles of the devices are out of sync in the beginning, because of the slip the cycles reach a good sync at about 7000 seconds and then loose the sync again. The minor cycle in the mean transmission time most likely is caused by slips in other cycles (e.g. the communication cycle).

All other measurements are based on the distributed scenario (two control nodes).

### 4.2. Measurement 2

In measurement 2 the same ASEs are used for communication of the IO data (AR_1 and AR_2), but the communication between the control nodes (AR_3) is based on a proprietary Ethernet-based protocol (Controller-Controller AR). This protocol is transported on the same Ethernet as the IO communication, which may lead to additional delays in the proprietary protocol, because it has a lower priority as the IO communication and less bandwidth is allocated for this protocol.

Figure 9 shows the transmission time observed over a duration of approximately 1,5 hours. In this measurement no cycles are recognizable. The variation in the transmission time is caused mainly by the proprietary non-deterministic Ethernet-protocol used in the communication between the control nodes.

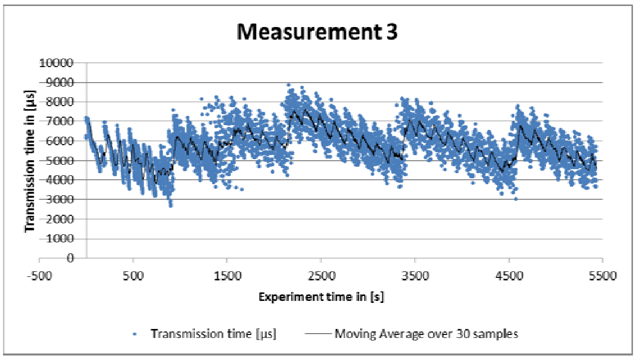



End-to-end-transmission-time:
Mean value: 10259µs; Minimal value: 4962µs;
Maximal value: 16632µs; Standard deviation: 2180µs

**Figure 9 – Distributed Scenario: Proprietary PLC-to-PLC protocol**

### 4.3. Measurement 3

In this measurement the communication between the control nodes (AR_3) is based on the same IO AR like the communication to the IO devices (AR_1 and AR_2). One of the controllers acts as producer and the other as consumer of the transmitted value.




End-to-end-transmission-time:
Mean value: 5749µs; Minimal value: 2682µs;
Maximal value: 8866µs; Standard deviation: 1003µs

**Figure 10 – Distributed Scenario: IO protocol**

Figure 10 shows the transmission time observed over a duration of approximately 1,5 hours. It is possible to recognize multiple cycles in the transmission time. Compared to measurement 1 it is recognizable that not only two cycles have major influence on the transmission time, but multiple cycles are out of sync. Compared to measurement 2 the variation of the transmission time has been reduced. The absolute end-to-end-transmission-time has not yet reached the same quality as in using one control node.

### 4.4. Measurement 4

In this measurement the communication between the controllers is based on the same IO AR like the communication to the IO devices. One of the controllers acts as producer and the other as consumer of the transmitted value. The communication between controllers and the IOs is synchronized (isochronous ASEs).

Because of the synchronized communication it was not possible to use an unmanaged switch in the hardware setup. That is why the structure of the communication system had to be changed to a line structure (see Figure 11).

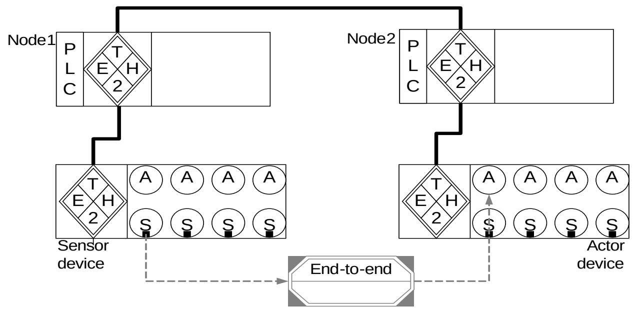
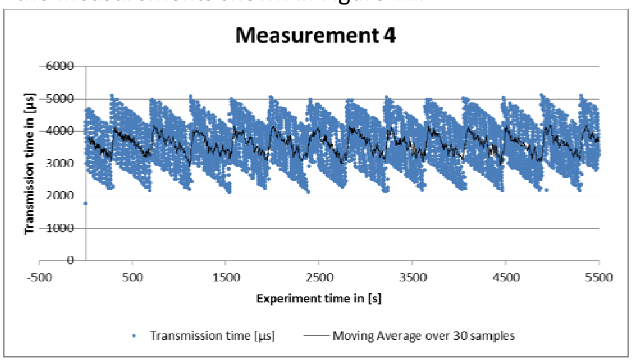



**Figure 11 – Distributed Scenario: Distributed control loop with line structure**

Measurement in this communication structure results in the measurements shown in Figure 12.




End-to-end-transmission-time:
Mean value: 3608µs; Minimal value: 1758µs;
Maximal value: 5124µs; Standard deviation: 651µs

**Figure 12 - Distributed Scenario: Isochronous protocol**

Figure 12 shows the transmission time observed over a duration of approximately 1,5 hours. The variation in the transmission time has been strongly reduced. Actually this measurement shows a better behavior than measurement 1. In measurement 1 the main source for variation was the slip in the IO cycle (2 ms), while in this measurement the control nodes are synchronized with the respective input or output.

### 4.5. Discussion of the measurement results

There are strong differences in the end-to-end-transmission time for different AR types. The properties of the AR types influence the result even if there is the same distributed structure.

The consequences for our example introduced in 3.1 are:

- Measurements 1-3 show that the respective network solutions do not meet the requirements. In all solution the jitter is too large. In measurement 2 and 3 the average transmission time is too large.
- In measurement 4 the average time for speed control cycle is approximately 3,6ms with a range from 1,8ms to 5,2ms. This meets the requirements because the average transmission time and the jitter is small enough.
- In measurements 1, 3 and 4 it is possible to observe frequency beats, which are the result of an interference of cycles with different frequencies. These frequency beats can be explained by the various cycles of software/hardware components in the communication chain, but are subject of ongoing research.

## 5. Conclusion

In this paper we show, that the selection of the used AR types and related communication services has significant influence on the quality of the automation solution. If the properties of the chosen AR and communication services are not considered, the quality of the automation solution degrades with distributed control. But choosing the correct AR and communication services allows implementation of distributed controls at a quality similar or even better than a un-optimized central control. On the other hand, it cannot be expected that a developer of control applications understands such details of communication services,

Further research of the authors will focus on developing suitable methods to support less experienced engineers in their choices to meet non-functional requirements when designing distributed control systems.

Such a method shall able to:

- capture functional and non-functional requirements for the distributed production system and for the control application and maintain these requirements during the design flow,
- support a stepwise design flow, with design stages from functional design to implementation (see [11]),
- identify the characteristics of the system under design at all stages of the design,
- use the characteristics of the system under design to support the engineers in their design decisions, and
- provide suggestions for solving design steps (see [10]).

**Acknowledgement**

The work presented in this contribution is funded by the German Research foundation (DFG) under project FAVA.

**References**

[1] Christensen, J. H.: “HOLONIC MANUFACTURING SYSTEMS: Initial Architecture and Standards Directions,” in First European Conference on Holonic Manufacturing Systems, Hannover, 1994.

[2] Leitão, P.: Self-Organization in Manufacturing Systems: Challenges and Opportunities. In (IEEE Hrsg.): Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, 2008; p. 174–179.

[3] Lüder, A. et al.: Distributed Automation: PABADIS versus HMS. In IEEE Transactions on Industrial Informatics, 2005, 1; p. 31–38.

[4] Lepuschitz, W. et al.: Toward Self-Reconfiguration of Manufacturing Systems Using Automation Agents. In IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2011, 41; p. 52–69.

[5] McFarlane, D. C.; Bussmann, S.: Developments in Holonic Production Planning and Control. In Int. Journal of Production Planning and Control, 2000, 11; p. 522–536.

[6] Herrero-Perez, D.; Martinez-Barbera, H.: Modeling Distributed Transportation Systems Composed of Flexible Automated Guided Vehicles in Flexible Manufacturing Systems. In IEEE Transactions on Industrial Informatics, 2010, 6; p. 166–180.

[7] Ghanaim, A.; Borges, G. A.; Frey, G.: Estimating Delays in Networked Control Systems Using Colored Petri Nets and Markov Chain Models. In Proceedings of the 14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2009), Palma de Mallorca, Spain, Sept. 2009.

[8] Greifeneder, J.; Frey, G.: Reactivity Analysis of different Networked Automation System Architectures. In Proceedings of the 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008), Hamburg, Germany, pp. 1031-1038, Sept. 2008

[9] Greifeneder, J.; Liu, L.; Frey, G.: Comparing Simulative and Formal Methods for the Analysis of Response Times in Networked Automation Systems, In Proceedings of the 17th IFAC World Congress, pp. 5113-5118, Seoul, Südkorea, July 2008

[10] Eckert, K.; Hadlich, T.; Frank, T. ; Fay, A. ; Diedrich, C.; Vogel-Heuser, B.: Design Patterns for Distributed Automation Systems with Consideration of Non-Functional Requirements, ETFA 2012, Krakow, 2012

[11] Frank, T.; Hadlich, T.; Eckert, K.; Diedrich, C.; Vogel-Heuser, B.; Fay, A.: Erweiterung des V-Modells® für den Entwurf von verteilten Automatisierungssystemen. In: Ulrich Jumar, Eckehard Schnieder und Christian Diedrich (Hg.): EKA 2012. Magdeburg, 2012

[12] Hadlich, T. et al.: Common communication model for distributed automation systems. In (IEEE Hrsg.): INDIN 2011. IEEE 9th International Conference on Industrial Informatics, Caparica, Lisbon, Portugal, 26-29 July 2011, 2011.

[13] IEC: IEC 61158-500: Industrial communication networks – Fieldbus specifications. IEC, Genf, 2009.

[14] Riefenstahl, U.: Elektrische Antriebssysteme. Grundlagen, Komponenten, Regelverfahren, Bewegungssteuerung ; mit 9 Tabellen und 70 Beispielen. Teubner, Wiesbaden, 2006. p. 407

[15] Vogel-Heuser, B. et al.: Modeling network architecture and time behavior of Distributed Control Systems in industrial plant automation: IECON 2011. 37th Annual Conference of the IEEE Industrial Electronics Society, 2011.