Concept of Distributed Interpolation for Skill-Based Manufacturing with Real-Time Communication

Caren Dripke[1], Ben Schneider[2], Mihai Dragan[1], Alois Zoitl[2], Alexander Verl[1]

[1] Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW)
Seidenstr. 36, Stuttgart, 70174, Germany, Tel. +49-711-685 84500
Caren.Dripke@isw.uni-stuttgart.de
[2] fortiss GmbH, An-Institut Technische Universität München
Guerickestr. 25, München, 80805, Germany, Tel. +49 89 3603522 0
schneider@fortiss.org

Abstract. Distributed manufacturing unit control can be implemented by equipping all manufacturing components with individual controls offering standardized skills. The control of multi-component groups often requires real-time communication and a communication architecture that is adapted to the distributed control concept. We present applicable communication concepts for distributed interpolation, where the distributed interpolation use case in particular demonstrates challenges caused by the synchronous execution of skills.

Keywords: distributed interpolation, deterministic real-time communication, time-Sensitive Networking, IEC 61499

1 Introduction

Fully networked and in real-time communicating automation systems, consisting of intelligent automation components - this vision is becoming more and more realistic in the age of digitization and industry 4.0. Controls, actuators and sensors can now be integrated into components because of the latest progress in electronics miniaturization. Embedded software makes of such components intelligent, communicative modules. These offer their functionality in the form of manufacturer-independent standardized automation functions called skills. Embedded software enables such components to take responsibility of their internal processes. Each component encapsulates the inner workings - all sensors and actors are controlled by integrated control - and offers abstract skills via a communication network. This signifies the integration of physical automation devices into the software control system [13] and is a step further towards the scientific goal of designing and building intelligent machines [18]. This further eliminates the need for classical wiring to a control cabinet and adjustment of signals/currents to specific protocols in order to interact with the mechanical component. The result is a component-based automation with a component-oriented engineering.

One research question is how component skills and their modularization can be (synchronously) executed in real time. A work package of the DEVEKOS research project aims to offer a communication solution for the envisioned component-based architecture. Goals are the continuous network-wide call for skills and the simple integration of third-party or previously unknown components with a focus on fault management, authentication, data confidentiality, and security.

DEVEKOS is funded by the German BMWi (Federal Ministry for Economic Affairs and Energy) with the goal to establish a consistent engineering as well as a workflow for safe, distributed and communicating multi-component systems in manufacturing units in the Industry 4.0 / Internet of Things (IoT) context [2].

The project focuses on the manufacturing unit and its components and does not specify solutions for distributed, decentralized manufacturing control or project planning. Integration into overlying levels of manufacturing execution systems (MES) and enterprise resource planning (ERP) systems are considered by using industry standards for information exchange as Automation-ML and OPC UA. DEVEKOS is looking to create concepts for engineering, safety, real-time communication and motion functionality that are easily applicable in industry.

The identified challenges of a communication concept have been presented in [6]. It is necessary to define a communication architecture that allows for real-time communication. Definition of the exchanged messages should aim for a compact representation of the information, including time stamping. Other aspects discussed were possible subnetwork structures as well as a handling strategy for possibly interrupted or corrupted communication. In this paper we introduce distributed interpolation as a use case to verify a concept that allows for real-time communication between individual components in the component-based architecture. We concentrate on real-time and deterministic communication as well as on the definition of compact messages. The definition of subnetwork structures is not in the scope of this paper.

2 State of the Art

We propose a concept of individually controlled components that interact with each other in order to fulfill an externally defined goal. The definition of Holonic Manufacturing Control (HMC) [13] can be applied to the skill architecture. HMC defines single components as interacting holons as well as groups that are made up of multiple components. This modularization has to be reflected on the software and communication architecture level. HMC allows to (re)combine, extend, amend or replace components or modules freely, which also implies publicly available interfaces that rely on industry standards [9]. Each holon offers skills to its environment, the offered skills depend on the level of the holon. A skill could be *OpenGripper()* for a holon describing a low-level gripper or *ManufactureWorkpiece()* for the uppermost holon level describing the manufacturing unit itself [10]. The high-level skill is itself a composed skill, meaning it is constructed by a sequence of lower-level skills. This orchestration of these lower-level skills and

especially their synchronized execution is the main focus of this paper. This concept advances the manufacturing unit towards a distributed, skill-oriented architecture, where submodules can be exchanged arbitrarily without loss of the general function of the manufacturing unit or the need for complex changeover processes. Components from various manufacturers are expected to work together in groups i.e. to connect and communicate with each other. The need to adhere to different protocols or setups depending on the manufacturer is replaced by universal skills. Up until now, an HMC implemented as a multi agent system has only been used in production process planning [3], where communication cycle times are significantly longer. When even the lowest-level components like axes, grippers or sensors are integrated into an HMC, real-time communication means cycle times below 10 milliseconds [7] for their assigned skills.
Distributed control concepts have been introduced for programmable logic controllers (PLCs) within the IEC 61499 [5]. 4diac [8] is an IEC 61499 compliant development environment for modeling distributed industrial automation systems. It simplifies the engineering workflow and supports different communication patterns and protocols including real-time and deterministic communication by a recently developed Time-Sensitive Networking (TSN) [19] communication layer described in Section 3.

TSN is currently standardized by the Time-Sensitive Networking Task Group which is part of the IEEE 802.1 Working Group.

The advantages of TSN can be summarized as follows:

- **Real-time:** TSN guarantees real-time capabilities by bandwidth reservation and configured schedules
- **Determinism:** TSN guarantees the transmission of packets by bandwidth reservation
- **Interoperability:** standard Ethernet as key enabler for Industrial Internet of Things (IIoT), as the need for dedicated fieldbuses is decreasing
- **Convergent network:** real-time critical and uncritical traffic are transmitted on the same cable, which results in better bandwidth utilization and reduction of cabling effort

The TSN substandard 802.1Qbv defines the allocation of Virtual Local Area Network (VLAN) to tagged and encoded priority values [16]. These allow simultaneous support of scheduled traffic, credit-based shaper traffic and other bridged traffic over Local Area Networks (LANs), thus making TSN a key enabler for the advantages mentioned above.
Fieldbuses and industrial Ethernet solutions such as EtherCAT, PROFINET or Sercos III are evaluated for their compatibility with the project requirements. EtherCAT is a real-time-capable Ethernet-based fieldbus developed at Beckhoff Automation. The EtherCAT approach makes the connected devices only read data addressed to them, whilst each communication telegram continues through the device. Similarly, input data are inserted while the telegram passes through. The telegrams are only delayed by a few nanoseconds, making this solution one of the fastest available. The bus system offers access speeds similar to local I/Os. EtherCAT enables not only the position control loop, but also the velocity con-

trol loop or even the current control loop to be closed via the bus, resulting in cost-effective drive controllers [1]. PROFINET IO is based on Switched Ethernet full-duplex operation and a bandwidth of 100 Mbps. It therefore provides the possibility of real-time communication with reserved bandwidth in its IRT-mode (Isochronous Real Time). In this case, a "sync master" transmits a synchronization message to which all "sync slaves" synchronize themselves with a synchronization accuracy of less than one microsecond [17]. SERCOS is a digital motion control bus that interconnects drives, motion controls, I/O, sensors and actuators for numerically controlled machines and systems. Launched in the 1990s, it is utilized for automation applications with high requirements for dynamics and precision. SERCOS III is the open, IEC-compliant third-generation SERCOS interface enabling persistent, bi-directional motion control communications in real-time between all drives and the motion control [11].

The innovative communication approach of TSN is also considered here as a potential solution, as it could be suitable for a consistent and reliable communication in distributed, asynchronous, event-driven and service-oriented architectures. Future work within the communication work package of this project includes the evaluation of known solutions, followed by the development of a suitable communication solution and an experimental validation of this concept.

3 Applicable Communication Proposal

The afore mentioned challenges for distributed interpolation imply several communication requirements [6], mainly real-time and deterministic traffic on a convergent network as described in Section 1. There are plenty of already existing real-time-capable, proprietary fieldbuses on the market, as described in Section 2. These industrial protocols are not enabling an Industrial Internet of Things (IIoT). Reasons are the limited scaling of bandwidth to 100 Mbps, the limited interoperability due to special hardware and the lack of supporting mixed traffic on a single network cable. TSN meets all these requirements the other fieldbuses lack, as described in Section 2, and is therefore chosen as the communication protocol for this project.

The proposed communication pattern is 'publish/subscribe' (pub/sub). It is best suited for an asynchronous, event-driven and distributed architecture with real-time requirements. Additionally, it enables multicast communication. The 4diac framework already provides an UDP based communication layer for pub/sub data streams with TSN support, which was implemented in the BEinCPPS project. This communication layer adds virtual local area network (VLAN) and priority configuration data to IEC 61499 compliant publish Service Interface Function Blocks (SIFB). The ID data input parameter of publisher SIFBs is used to select a specific network stack and to configure the parameters.

The ID parameter of 4diac's TSN communication layer has the following format: `fbdk[].tsn[<ip>:<port>:<vlan_id>:<prio>]`. Whereas `fbdk[]` is responsible for the appropriate ASN.1 encoding defined in IEC 61499-1 Annex F [5], `tsn[..]` specifies that the TSN enabled communication layer of the 4diac

runtime environment is used.
There are different message types necessary for the current implementation of distributed interpolation. The proposed TSN approach maps the message types to priorities (0 highest, 7 lowest priority) and VLANs which are then treated differently according to the configuration of the TSN schedule. The different message types and priorities are mapped as follows:

– MSG 1: highest priority (0) - represents a skill request;
– MSG 2: priority (1) - for synchronization, negotiation and interpolation tasks;
– MSG 3: no priority - device internal data for motion control;
– Network control traffic: priority (2) - IEEE 1588 Precision Time Protocol (PTP) [15] for time synchronization;
– Best effort network traffic: lowest priority (7).

MSG Type	Sender Addr.	Target	State Status	Max. Reach. Pos.	Current Pos.
MSG 1	4	max. 64	1	—	—
MSG 2	4	max. 64	1	1	4
MSG 3	4	max. 64	1	—	—

Table 1: Different message types and their tags including the size [Byte]

MSG 1 through 3 are directly connected to the distributed interpolation and will be presented in detail in the following chapter. The separation into three main message types is motivated by the fact that the skill request (MSG 1) should be a network-wide standard. The standardization process what a skill request message entails is not yet concluded. Changes are to be expected but will not affect the underlying messaging concepts. The separation between MSG 2 and MSG 3 is motivated by the distinction between internal messaging (on the individual axis control) and the negotiation between multiple axis controllers via the network. The preliminary message sizes are shown in Table 1. An address and a transmitted target (requested skill, possibly subdivided in tasks) as well as a communication status are always included. In case of a negotiation in the interpolation task, an axis will share its maximum reachable position as an offer, as well as its current position.

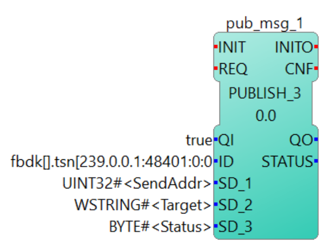

Fig. 1: TSN enabled publish SIFB in 4diac

The appropriate TSN enabled network SIFB can be derived when message type, priority mapping, VLAN mapping and size of the message are defined, see

Figure 1. The example ID parameter `fbdk[].tsn[239.0.0.1:48401:2:0]` of a publisher SIFB would map a message stream of the proposed interpolation MSG 1 to the UDP multicast group with IP address 239.0.0.1 on port 48401 to VLAN 2 with priority 0. The data types are chosen according to the different tags of the message (e.g., sender address, target, ...) and their size. The implementation of the TSN layer and the analysis of message types and their mapping to TSN priorities and VLANs will be the base for evaluating the communication concept.

4 Architecture for Distributed Interpolation

Distributed skill-based control brings communication challenges that are not covered by current centralized control communication architectures. Functionalities like synchronization, online adaptations to the motion target or status notifications are currently requests from the manufacturing network. These functionalities will be encapsulated in skills in the component based architecture. The first advances towards distributed control as IEC 16499 [5] are mostly used to implement sequential event-chains of events. Non-sequential skill request however, as e.g. the movement of multiple axes interpolating a specified path, are not specified. Here, the independently controlled individual movements (individual task decomposition of the skills has yet to be defined) have to be synchronized. If any discrepancy from the planned path is detected, all axes have to determine a compensation strategy (adapt motion target, abort movement, ...). Communication offers the possibility to solve such tasks in collaboration [18]. It is expected, that a communication architecture that meets the requirements of the distributed interpolation use case is transferable to any upper-level HMC / multi-component group solving non-sequential skill requests. The orchestration [12] of skill-execution or computation of transformations, kinematic relations or dynamic capabilities are not discussed here but will be subject of a future publication.

The distributed control architecture proposed requires all components to be ablt to interpolate motion have identical communication interfaces. The implementation is realized with three state machines (SM) which separate the modular responsibilities as follows (see also Figure 2):

- Interpolation Master (IPO-Master): This SM is available in all component controls, but for each multi-component group only a single interpolation master is chosen. In this context, we assume the choice of the IPO-Master as given (could e.g. be done by a bidding algorithm, by fastest response times,...). The IPO-Master receives the skill request in Message Format 1 (MSG 1). It passes all relevant information on to the Communication State Machines (CommSM) of all axes in the group via MSG 2. The coordination and execution of all motion, synchronization and negotiation until an agreement is responsibility of the IPO-Master.
- Communication State Machine: Checks incoming requests and negotiates with IPO-Master if necessary. Here, high loads of messages are possible and time-crucial. Passes on the agreed motion to MotionSM via MSG 3.

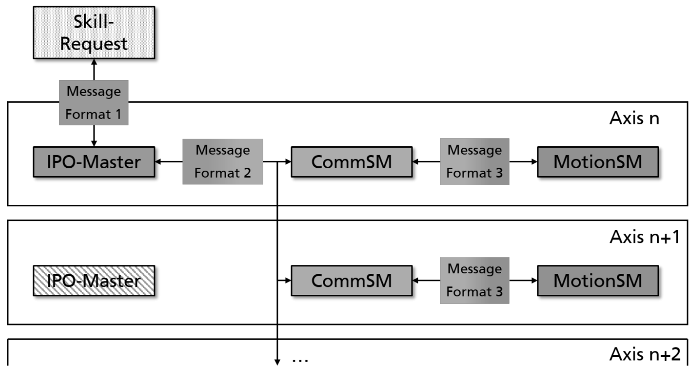

Fig. 2: Message-based communication between the state machines (SM): An IPO-Master is chosen, receives skill request (MSG 1). IPO Master communicates with the CommSMs through MSG 2. Communication between MotionSM and CommSM is internal over MSG 3.

– Motion State Machine (MotionSM): Executes motion, not accessible directly, but only communicates with the CommunicationSM via MSG 3.

An axis under a distributed interpolation HMC offers skills such as *GoToPosition()* and other motion profiles. The proposed set of skills are still under discussion, but are oriented to the PLCOpen [14] and CANOpen [4] standards. A combination of these motion skills then offers the possibility to combine them in a multi-component group to orchestrate interpolated motion.

5 Outlook and Conclusion

We presented the context of the DEVEKOS research project. The proposed communication of distributed interpolation can present an example for other non-sequential multi-component skill-requests. Challenges of the current communication concepts and possible technologies were also presented. The next step is the realization of the concept with a prototype using 4diac and TSN.

6 Acknowledgements

The DEVEKOS research and development project is funded by the Federal Ministry of Economic Affairs and Energy (BMWi). It is part of the technology program "PAICE Digitale Technologien für die Wirtschaft" and is guided be the DLR Project Management Agency ”Information Technologies/Electromobility“, Cologne. The authors are responsible for the contents of this publication.

The present work has been partially carried out in the framework of the BEinCPPS project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 680633.

References

1. Beckhoff: Beckhoff information system. https://infosys.beckhoff.com/index_en.htm (2017) accessed 27.10.2017.
2. Bundesministerium für Wirtschaft und Energie: DEVEKOS. http://www.digitale-technologien.de/DT/Redaktion/DE/Standardartikel/PAICEProjekte/paice-projekt_devekos.html (2017) accessed 06.04.2017.
3. Bussmann, S., Jennings, N.R., Wooldridge, M.J.: Multiagent systems for manufacturing control: a design methodology. Springer Science & Business Media (2004)
4. CAN in Automation (CiA) e. V.: Drives and motion control device profile; Part 2: Operation modes and application data. Draft Standard Proposal No. 402 (10 October 2016)
5. DIN EN 61499-1: Funktionsbausteine für industrielle Leitsysteme – Teil 1: Architektur (2013)
6. Dripke, C., Verl, A.: Challenges in distributed interpolation with multi-components systems in future-oriented manufacturing units. Proceedings of the 47th International Conference on Computers and Industrial Engineering 2017 (2017 - in press)
7. Dürkop, L.: Automatische Konfiguration von Echtzeit-Ethernet. Springer (2016)
8. Eclipse: 4diac Framework for Distributed Industrial Automation and Control. https://www.eclipse.org/4diac/index.php (2007) accessed 19.10.2017.
9. Frey, K.: Offene Steuerungssysteme-eine Zwischenbilanz. In: FTK'97. Springer (1997) 271–297
10. Helbig, T., Henning, S., Hoos, J.: Efficient engineering in special purpose machinery through automated control code synthesis based on a functional categorisation. http://link.springer.com/content/pdf/10.1007%2F978-3-662-48838-6_9.pdf (2016)
11. Hibbard, S.: Industrial ethernet: The key advantages of sercos iii. https://www.automation.com/pdf_articles/Rexroth_SERCOS_III__L-1.pdf (2010) accessed 27.10.2017.
12. Jammes, F., Smit, H., Lastra, J.L.M., Delamer, I.M., eds.: Orchestration of service-oriented manufacturing processes. Volume 1., IEEE (2005)
13. Leitão, P.: Agent-based distributed manufacturing control: A state-of-the-art survey. Engineering Applications of Artificial Intelligence **22**(7) (2009) 979–991
14. PLCopen: TC2 - Motion Control. http://www.plcopen.org/pages/tc2_motion_control/ (2011) accessed 30.10.2017.
15. Precise Networked Clock Synchronization Working Group: Enhancements for scheduled traffic (2008)
16. Precise Networked Clock Synchronization Working Group: Ieee standard for a precision clock synchronization protocol for networked measurement and control systems (2008)
17. Profinet: Profinet system description (2012)
18. Shen, W., Norrie, D.H., Barthès, J.P.: Multi-agent systems for concurrent intelligent design and manufacturing. CRC press (2003)
19. Time-Sensitive Networking Task Group: Ieee standard for local and metropolitan area networks – bridges and bridged networks – amendment 25: Enhancements for scheduled traffic (2015)