Sehr geehrter Herr Professor Jumar,

wir freuen uns über die positive Rückmeldung und bedanken uns recht herzlich bei Ihnen, den Editoren und Reviewern für die Kommentare, die uns dabei geholfen haben, das Paper zu verbessern.

Anbei finden Sie eine Tabelle mit der Zuordnung von Reviewer-Kommentaren zu der jeweiligen Revision, die wir im Paper vorgenommen haben. Des Weiteren ist eine Version des Papers beigefügt, in der alle Änderungen sichtbar in blauen Text hinterlegt sind.

Danke und mit freundlichen Grüßen

Dominik Hujo-Lauer

Wissenschaftlicher Mitarbeiter,

Lehrstuhl für Automatisierung und Informationssysteme

Technische Universität München

#	Reviewer 1 comments	Response and revision by the authors
#R1-1	Kapitel 1-3 sind eigentlich nur Stand der Technik, hier sollten die Autoren genau ihren Beitrag und den Mehrwert des Artikel herausarbeiten.	Vielen Dank für den Kommentar. Die Introduction wurde überarbeitet, um den Mehrwert des Artikels präziser zu verdeutlichen. Folgende Passagen wurden hinzugefügt: […] This potential remains underutilized in production environments due to technological fragmentation and lack of interoperability. Model-based design support can overcome these challenges arising from heterogeneous technologies used in industrial automation systems by handling several aspects addressing the integration of autonomous robots into challenging industrial environments. Besides others, one critical aspect is securing the system's real-time capability while using heterogeneous technologies in distributed network-controlled systems. Requirements arising from the technical process speed and the latency properties of system components must be matched to achieve this real-time capability. This system's latency occurs from the execution of software skills, communication between controllers, and latency of sensors, actuators, and network equipment [4]. This paper addresses the design support for integrating distributed heterogeneous control systems by examining the execution and time behavior of skills in production plants and robotic systems. Based on a domain-specific language from [4] and [5], the proposed concept enables the modeling of skill and communication latency from industrial automation and robotics in a unified way. The skill descriptions used are taken from the Capability, Skills, Service (CSS) approach [6], which defines capabilities, skills, and services in the context of industrial automation, and from [7], where robotic skills for manufacturing are defined. The main contribution of this paper is the adaption of a unified skill description to the modeling approach in [5], enabling a model-based problem description of distributed heterogeneous control systems, which allows for an analysis of the system during the runtime and even before setting up the system. The analysis is achieved through modeling and subsequent calculation of system latency based on established metrics, visualizing the currently active skills, and analyzing the communication between their outputs. The DSL aims to support application engineers, deploying identified control functions to different available controllers considering requirements from the technical process, achieving faster design phases and lower costs in control technology. A representative use case was selected to analyze the differences between the domains, which is further discussed in Section 2. . Weiterhin wurde der Hinweis eingefügt, dass es sich bei Kapitel 2 um Beschreibungen aus dem Stand der Technik handelt: The remainder of the paper is structured as follows. First, an overview of standard skill definitions in robotics and industrial automation from state-of-the-art literature is given, and a combined use case is presented.
#R1-2	Generell trifft das Paper keine genaue Aussage. Die Autoren müssen deutlich machen was genau der Mehrwert ist und dabei auf unnötige Details verzichten (Quellcode, Lange Liste ungenutzter Skills).	Die Herausstellung des Mehrwerts wurde geschärft durch Anpassungen in der Einleitung (Section 1, siehe Antwort auf #R1-1) und der Anpassung der Zusammenfassung und Ausblick (Section 7, siehe Antwort auf #R1-6). Außerdem wurde verdeutlicht, was der Mehrwert der Listen und Abbildung ist (siehe Antwort auf #R1-3)
#R1-3	Alle enthaltenen Figures müssen einen Mehrwert für das Paper bieten, referenziert und erklärt werden.	Vielen Dank für den Kommentar zur Verbesserung des Papers. Figures 5, 6 und 7 wurden angepasst, um den Mehrwert besser herauszustellen und den Bezug zwischen der Tabelle, der Skillmodellierung und dem Code besser zu verdeutlichen.

#R1-4	Die Zusammenführung von Skillmodelling und Latency kommt nicht klar raus wie das gemacht wird. Falls das Skill Modelling bereits Stand der Technik ist müssen hier unbedingt relevante Zitate eingeführt werden.	Die Zusammenführung zwischen Skillmodelling und Latency analysis wurden detaillierter erklärt, und die Quellen zur Stand der Technik deutlicher herausgestellt: […] The latency analysis should show if the requirements derived from the technical process and the latency properties of the system correlate and if the process is feasible to operate. On the one hand, it can be analyzed if a PLC task can handle the number of skills executed. This analysis focuses on a cyclic execution of control code. On the left side of Fig. 11, an example of this cyclic controller execution is given. The skills included within this controller are benchmarked, and the maximum execution time for one cycle is given. The connectors state if the skills on the controller can be executed in parallel (Parallel Skill Connector or Parallel Skill Illustrator) or in sequential (Sequential Skill Connector) order. This is important for calculating the overall code execution time within one cycle. If the skills are modeled in sequence, this states, that the skills cannot be executed within one cycle and, therefore, must not be added to the overall execution time. If a skill is stated with a parallel execution, the skill times are added to an overall execution time. Communication between skills is not explicitly measured. The communication between the two shown controllers can be measured with ca. 4200 nanoseconds, as stated in the octagonal annotation element on the sequential connector between the skills DirectionCotnrol1 and HomeGripper. Another analysis can be done for the sequential execution controller. In order to secure a high throughput of the machine, the overall interaction between the two systems, CX2040 and Fraca control, should not take longer than 20 seconds. When the process is even slower than 50 seconds, the queuing line on the conveyor of the MyJogurt plant also can overflow. That leads to two requirements, which fulfillment can be calculated based on the skill execution and the communication between the CX2040 and the Franca Control. Therefore, the sensor variable that triggers the process is identified (Cp. Fig. 11 Sensors (within a skill)), and the signal flow is followed to finish the process. All the times that occur in between are then added. Additionally, the hardware view of the DSL is used for the full description of latency in the system, which is not demonstrated in this paper. The calculation of the two analyses is given in further detail.
#R1-5	Die Modellierung der Skills sollten genauer ausgeführt und detailliert werden. Das Paper würde sehr davon profitieren die Latency-Analyse restlost zu streichen.	Die Modellierung der Skills wurde weiter ausgeführt innerhalb der Erklärung des Zusammenhanfs zwischen Skillmodelling und Latency Analysis. (siehe Antwort auf #R1-4). Die Bedeutung des Latency Analysis Teils wurde genauer erläutert im Paper um den Zusammenhang besser herauszustellen. (siehe Antwort auf #R1-4).
#R1-6	Der Outlook muss entsprechend überarbeitet werden, um den Mehrwert des Paper herauszustellen.	Vielen Dank für den Kommentar. Der Outlook wurde überarbeitet, um den Mehrwert des Artikels präziser zu verdeutlichen. Folgende Passagen wurden hinzugefügt: Designing distributed heterogeneous networked control systems is a challenging task itself, while real-world production scenarios offer very high potential for optimization in the context of IoT / I4.0. The combination of technologies from various domains further complicates this process. This paper presents an approach for the equivalent modeling and analyzsis of skills from the domains of robotics and industrial automation to support the design of such integrated systems. The developed approach builds on an existing domain-specific language. It incorporates hardware, software, and functional perspectives, enabling a semi-formal problem description and providing support for analyzing system designs prior to the commissioning phase, thereby aiming to save time during the testing process and facilitating system validation. The DSL supports the design phase of distributed control system architecture by making system design requirements and properties formalizable and comparable. The main contribution of this paper is the detailed definition of skills and their integration into the DSL of the referred latency modeling approach. This semi-formal definition enlarges the approach from [5] to enable the modeling and integration of heterogeneous software skills from different domains. It formulates a skill's internal

		structure and its relation to other system components on an abstraction level, which allows the modeling of robotic skills and skills from industrial automation. Integrating skills from different domains is essential in modeling modern CPPS, as they use various technologies partly adapted from consumer domains. The proposed concepts were evaluated through a use case combining robotics and industrial automation. Different scenarios were tested under normal operating conditions as well as in fault recovery states, providing comprehensive insights into their applicability. Future work will focus on exploring suitable metrics to describe the interaction between hardware and software to further automate the latency analysis of CPPS. Additionally, enabling dynamic and automated skill distribution within the overall system of the distributed networked control system will be a key objective. The long-term goal is an integration platform based on the developed DSL, enabling the design and ramp-up of a plant's distributed control system based on requirements derived from the defined technical process.
#R1-7	Einzelne Formulierungen ("left y-axis", "is most likely caused by the fact", "resulting in the observed 0 values") deuten darauf hin, das der Autor ebenfalls Probleme bei der Darstellung, Beschreibung und Interpretation seiner Ergebnisse hat	Der Text wurde präzisiert. Weitere Erklärungen, wie die Modellierung bei der Latency Analyse helfen kann, wurden angefügt. Weiterhin wurden die sprachlichen Unsicherheiten angepasst: […] Still, the call in the init sequence is not redundant and cannot be left out, since the higher level skill does not make assumptions about the state of the system and first transfers the robot into a well-defined initial state. A significant result derived from Fig. 13 is, that the execution time of a single skill can vary based on the use case that is operated. By modeling the problem in Fig. 11, the measured times of skills and communication can be summed up, enabling an evaluation of the use case. The *MaxExecutionTimes* of the robotic skills from Fig. 13 and the *MaxInnerCycleTime* of the automation skills from Fig. 12 are stated in the model as metrics. For the cyclic behavior of the PLC, it can be analyzed whether the defined cycle time is still being met or if a redistribution is necessary (*MaxExecutionTimes*). The experimental study focused on measuring the latency of skill execution within the system. While inter-skill communication latency between subsystems was not explicitly captured in the experiments, the latency introduced by the non-deterministic behavior of the MQTT protocol, used for communication between the PLC and the robot, has been considered in the model. To account for the impact of this latency, an estimate for the order of magnitude has been included in the visual representation of the use case (cf. Fig. 11). When defining system requirements, such as a minimum reaction time, the feasibility of the approach can be determined, as described in [5], by summing the corresponding communication times, the sequentially executed skills on the Franka robot, and the cyclically executed PLC skills. The result of the consolidated model is a DSL in which a holistic overview of the latency within and between the different systems can be captured for a problem description. […] […] For the cyclic execution of the PLC (cp. Fig 11, ControlerBox CX2040), the summed-up execution time of the shown excerpt would be CE_S31 = 2500ns + CE_C32 = 200ns + Driver_S31 = 300ns + Driver_C32 = 500ns equals 3500ns or 0,0035 ms, which is way smaller than the cycle time of 10 ms. This result was expected as the excerpt of the control code was small compared to a complete control program. The analysis of the sequence behavior can be done by summing up all skills involved in the task execution, from the triggering of a sensor in the PLC skills to the full execution of robotic skills. This includes two cycles of PLC task (20ms = Worst-Case Execution Time), the MQTT communication (420µs), once the skill HomeGripper (6,05s), once MoveToSavePose (5,23s), four times MoveCartesian (4,01s), once MoveGripper (1,27s), once Grasp (1,47s), and once MoveCartesianWHGT (4,01s). This rounds up to 34,1s. As discussed in Section 5, the process should not take longer than 20s to be efficient and not more than 50s to avoid safety issues. The safety requirement can be addressed, but the robot skills are executed too slowly for efficiency. In future work, this analysis should automatically be done by a tool that considers all possible input and output combinations of the process to provide a supportive overview of the system behavior.

#R1-8	Introduction: refer [3] earlyer: "a 2021 McKinsey study [3]"	Die Referenz wurde an die vorgeschlagen Stelle versetzt: Additionally, the industrial automation domain inherently requires fulfilling specific challenges, such as timing, determinism, reliability, availability, and security. According to a 2021 McKinsey study [3], IoT holds a value potential of up to $13 trillion by 2030, with $3.3 trillion attributed to the manufacturing industry alone.
#	**Reviewer 2 comments**	**Response and revision by the authors**
#R2-1	The measurement of the skill latency focusses on the skill execution itself. From an overall system perspective the communication between the subsystems (PLC and RC) also introduces latency. In the example, MQTT is used for that communication that, due to its nondeterministic behaviour introduces latency. A short remark stating that the experiment leaves the inter skill communication out of scope might be helpful for then reader	Thank you for the helpful comment on improving the paper. A remark has been added to the last paragraph of chapter 6. Explicitly mentioning the limitation that the latency introduced by the MQTT communication has not been captured during the experiments for measuring skill execution latency. And that a (separately measured) estimate has been included in the model of the presented use case: The experimental study focused on measuring the latency of skill execution within the system. While inter-skill communication latency between subsystems was not explicitly captured in the experiments, the latency introduced by the non-deterministic behavior of the MQTT protocol, used for communication between the PLC and the robot, has been considered in the model. To account for the impact of this latency, an estimate for the order of magnitude has been included in the visual representation of the use case (cp. Fig. 11).