Technische Universität München
TUM School of Computation, Information and Technology

# Studies on the impact of operator-designed cobot's behaviors for industrial use-cases

**Matteo Pantano**

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

**Doktors der Ingenieurwissenschaften (Dr.-Ing.)**

genehmigten Dissertation.

**Vorsitz:** Prof. Dr.-Ing. Matthias Althoff

**Prüfende der Dissertation:**

1. Prof. Dr. Dongheui Lee
2. Prof. Dr. Cristina Piazza

Die Dissertation wurde am 21. Dezember 2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 10. Juni 2024 angenommen.

## Abstract

Small and medium enterprises in Europe face pressure to produce goods with competitive pricing. Additionally, they are facing constant changes in production volume, product variety, and labor shortages. Collaborative robots are a promising technology for helping small and medium enterprises overcome these challenges as long they can help with dull and repetitive tasks. However, their adoption is now in a market sobering phase, and three main adoption barriers have been identified: safety, design, and interfaces. Overcoming these barriers can enable personnel at small and medium enterprises to reprogram these systems with minimal effort, thus enabling non-experts to perform such tasks on multiple occasions. This work addresses the three barriers by proposing and studying the effects of methods to simplify the programming of collaborative robots, mainly using spatial interactions through the users' movements of 3D sensors or workpieces. First, when experts and non-experts program robots by the placement of workpieces, non-experts select workpieces' locations that do not endanger their physical ergonomics. In contrast, experts set locations that could endanger their ergonomics mainly because they were twice as close to the robot than non-experts during the interaction. Second, when non-experts program the grasping location of a robot, conventional systems do not provide easy-to-use interfaces, and a spatial interaction can improve usability by 38% while improving the robustness of the taught grasp location. Third, when non-experts program robot trajectories via spatial interactions, they create trajectories that require 5% more separation distance between the user and the robot to ensure safety. Last, workflow-based programming for integrating the spatial interactions improves non-experts' usability by 51% when compared with similar programming methods. Therefore, the studies reveal that when users are empowered to program the robot behaviors through these interactions, different levels of robotic expertise impact the task and the operator differently. More specifically, the methods developed in the course of this work enable simplified programming of robots and lead to an improvement in workload compared to the state of the art. This work thus emphasizes the importance of user-friendly methods for robot programming. Using the presented methods and other algorithms could empower European small and medium-sized enterprises to produce goods at competitive prices with high product variety and frequently changing product volumes.

## Zusammenfassung

Kleine und mittlere Unternehmen in Europa stehen zunehmend unter Druck, Waren zu wettbewerbsfähigen Preisen herzustellen. Zusätzlich müssen sie mit ständigen Veränderungen im Produktionsvolumen zurechtkommen, eine große Teilevielfalt bedienen und sehen sich einem Fachkräftemangel ausgesetzt. Kollaborative Roboter sind eine vielversprechende Technologie, um kleinen und mittleren Unternehmen bei der Bewältigung dieser Herausforderungen zu helfen. Sie können eingesetzt werden, um bei monotonen und repetitiven Aufgaben zu unterstützen. Die Nutzung von kollaborativen Robotern befindet sich derzeit jedoch in einer Phase der Marktabkühlung. Dabei verhindern folgende drei Hauptbarrieren eine weitere Verbreitung der kollaborativen Roboter: Sicherheit, Design und Schnittstellen. Das Überwinden dieser Barrieren ermöglicht Mitarbeitern in kleinen und mittleren Unternehmen, solche Systeme mit minimalem Aufwand trotz fehlender Robotikexpertise bei neuen Aufgaben zu verwenden. Diese Arbeit befasst sich mit der Überwindung der genannten Herausforderungen. Sie stellt Methoden zur Vereinfachung der Programmierung kollaborativer Roboter vor und untersucht deren Auswirkungen. Dabei liegt der Fokus auf räumlichen Interaktionen des Benutzers mit dem Werkstück oder einem 3D Sensor zur Programmierung des Roboters. Erstens zeigte sich, dass Nicht-Experten bei der Programmierung von Robotern durch die Platzierung von Werkstücken Arbeitsplatzpositionen auswählten, die ihre physische Ergonomie nicht gefährden. Im Gegensatz dazu wählten Experten Positionen aus, die im Hinblick auf ihre Ergonomie als kritisch bewertet werden können, da der Abstand zwischen Mensch und Roboter während der Interaktion halb so groß war wie bei Nicht-Experten. Zweitens kann eine räumliche Interaktion zur Programmierung des Greiforts eines Roboters die Benutzerfreundlichkeit um 38% verbessern und gleichzeitig die Robustheit des gedachten Greiforts erhöhen. Drittens erstellen Nicht-Experten, wenn sie Robotertrajektorien über räumliche Interaktionen programmieren, Trajektorien, die einen um 5% erhöhten Sicherheitsabstand zwischen Benutzer und Roboter erfordern. Zuletzt steigert die workflowbasierte Programmierung zur Integration räumlicher Interaktionen die Benutzerfreundlichkeit für Nicht-Experten um 51% im Vergleich zu ähnlichen Programmiermethoden. Diese Arbeit unterstreicht damit die Bedeutung von benutzerfreundlichen Methoden zur Roboterprogrammierung. Die vorgestellten Methoden ermöglichen die vereinfachte Programmierung von Roboter und führen zu einer Verbesserung der Arbeitsbelastung im Vergleich zum aktuellen Stand der Technik. Die Nutzung der vorgestellten Methoden, auch in Kombination mit weiteren Algorithmen, könnte kleine und mittlere Unternehmen in Europa dazu befähigen, Waren zu wettbewerbsfähigen Preisen bei hoher Produk-

tvielfalt und sich häufig ändernden Produktvolumina herzustellen.

# Contents

## List of Abbreviations

**AM** Additive Manufacturing
**API** Application Programming Interface
**BB** Bounding Box
**BPMN** Business Process Model and Notation
**CAD** Computer Aided Design
**CAS** Computer-aided-safety
**CB** Context Broker
**CK** Chroma Key
**CNN** Convolutional Neural Networks
**Cobots** collaborative robots
**CV** Computer Vision
**DHM** Digital Human Models
**DL** Deep Learning
**DLOs** Deformable Linear Objects
**DLO-WSL** Deformable Linear Objects - Weakly Supervised Labeling
**EE** End Effector
**EU** European Union
**FOF** Factory Of the Future
**GQ-CNN** Grasp Quality Convolution Neural Network
**GRP** Generic Robot Programming
**HCD** Human-Centered Design
**HG** Hand guiding
**HH** Human-High
**HL** Human-Low
**HMI** Human Machine Interface
**HRC** Human Robot Collaboration
**I4.0** Industry 4.0
**IEC** International Electrotechnical Commission
**IK** Inverse Kinematics
**IoU** Intersection over Union
**ISO** International Organization for Standardization
**IT** Information Technology
**KPIs** Key Performance Indicators
**LERP** Linear Interpolation
**LfD** Learning From Demonstration

**M** Mean
**MSDs** Musculoskeletal Disorders
**MTM** Methods Time and Measurement
**NLP** Natural Language Processing
**NN** Neural Network
**NoC** Number of Clicks
**OLP** Off-Line Programming
**OT** Operational Technology
**PFL** Power and Force Limiting
**PTP** Point To Point
**RH** Robot-High
**RITM** Reviving Iterative Training with Mask Guidance for Interactive Segmentation
**RL** Robot-Low
**ROS** Robot Operating System
**RTM** Robot Time and Motion
**RULA** Rapid Upper Limb assessment
**SD** Standard Deviation
**SDG** Sustainable Development Goals
**SHOP4CF** Smart Human Oriented Platform for Connected Factories
**SLR** Structured Literature Review
**SMEs** Small-Medium Enterprises
**SPL** Spline
**SRMS** Safety-Rated Monitored Stop
**SSM** Speed and Separation Monitoring
**SUS** System Usability Scale
**TLX** Task Load Index
**TCP** Tool Center Point
**UGM** User Grasp Metric
**UI** User Interface
**UML** Unified Modeling Language
**UN** United Nations
**UR** Universal Robots
**VBA** Visual Basic for Applications
**VC** Visual Components
**VR** Virtual Reality
**VSD** Value-Sensitive Design
**VSM** Value Stream Mapping

**WAAM** Wire Arc Additive Manufacturing
**WDQ** Work Design Questionnaire
**XML** Extensible Markup Language
**YOLO** You Only Look Once

## List of Tables

## List of Figures

Chapter 1

# Introduction

## 1.1 Motivation

Robotics is set to play a tremendously important function in achieving the United Nations (UN) Sustainable Development Goals (SDG) [1]. For example, robotics can help reduce pollution with solutions for sorting or collecting trash [2], thus allowing humanity to flourish in the digitization age. Such expectation is widely shared, and news about robots can spark interest worldwide to create "viral" content, especially when impressive features are shown [3]. Such technological advancements are the offspring of numerous research activities in the last two decades. One embodiment of the progress achieved in robotics is the new robotics breed of collaborative robots (cobots). Cobots are robots designed with human interaction in mind. They integrate technical elements that allow them to work in a shared workspace with humans, thus removing the need to cage them in specific work areas as with standard industrial robots. For their peculiarities, cobots have been warmly accepted in the market, and, since their first appearance as finished products, their market share grew up to 5.7% in 2021, showing always a yearly increase [4]; yet, the addressable market is much broader. One appealingly customer segment for cobots are Small-Medium Enterprises (SMEs). SMEs are companies with a staff headcount lower than 250 and annual turnover lower than € 50 million as defined by the European Union (EU) [5]. SMEs are compellingly for cobots because robots' adoption in these enterprises is much lower when compared with larger ones [6]. Therefore, there is a large, highly untapped room for growth. Additionally, SMEs are known to be highly relevant to countries' economic growth. For example, only in the EU during 2018, SMEs represented more than 99% of the businesses, and created more than half of the European Gross Domestic Product (GDP) [6–8]. Lastly, a literature study showed that SMEs that adopted robots achieved higher productivity, efficiency, and employment when compared to the ones that did not [9]. In light of these points, it is visible that both cobots manufacturers and SMEs can benefit from

cobots' adoption. However, their diffusion could be faster, and several barriers hinder cobots' growth. A prominent literature research article has identified three main obstacles when introducing cobots, which are: applications, intuitive interfaces, and safety [10], where applications refers to the need to tailor cobots for a well-defined set of use-cases. Intuitive interfaces refers to the need that end-users of cobots must be enabled with simple-to-use interfaces. Safety refers to the need for transparent, documented methodologies to deal with the risks of cobots and human interaction. A summary of cobots' history and actual developments is useful to review for understanding how these needs affect cobots' technological developments. The following sections underline the main drivers behind specific barriers and the challenges to address.

## 1.2 Cobots' History and Challenges

### Origins

Cobots origins date back to the 1990s. The first cobot systems were part of a small family of passively actuated mechanisms that guided human operators' movements in the General Motors production line [11]. Specifically, these devices were actuated through human-exerted forces, and using software-programmed variable transmission elements, they guided a particular device and the operator on a well-defined path, thus reducing the operator's workload [12]. However, this idea of cobots as passively actuated devices lasted briefly, and the concept moved toward a lightweight robotic system. The first designs of a light robot appeared in space robotics research. Such a design was meant to address the needs of robotics operations without gravity. Therefore, a 1:1 weight-to-payload ratio, to avoid joint control motors' overloading, was embraced [13]. Out of this design, the German robot manufacturer KUKA released the first commercial product in 2008 under the name of iiwa. Iiwa's peculiarities were threefold. First, iiwa integrated a walk-through programming that allowed the robot to move through user-defined positions. Second, compliance with current legislation on robot machinery and safety through sensors enabled the robot to stop in case of collisions. And, finally, a low weight-to-payload ratio [14]. Similarly, the Danish start-up Universal Robots (UR) released its first cobot, the UR5, during the same decade. However, unlike the iiwa, the main features of the robot were a simple programming user interface (UI), safety through work-zone monitoring, and database structures for saving programming commands [15]. Therefore, the first decade of the 2000s ended with two fundamentally different approaches to cobot design and human interaction. On one side, the KUKA iiwa lightweight robot with embedded collision detection and

walk-through programming meant for direct interaction with humans. Conversely, a lightweight robot without embedded collision detection but with a UI meant to be used by non-robotic experts.

With the emergence of cobot systems in the market, a need for standardization was essential to guarantee customers' safety and compliance with regulations. Once cobots reached the market, only general standards, which do not describe how a cobot could be made safe, like the type A ISO12100:2010 [16] standard, were available. Therefore, the 2020s began with a revision of a crucial robot standard from the International Organization for Standardization (ISO), the ISO 10218. ISO 10218 is a type C standard that dictates common design features that should be adopted when designing or integrating industrial robots, and it is divided into two parts. Part 1, also referred to as ISO 10218-1 [17], describes which features should be integrated into the design of a safe industrial robot by the manufacturer. Part 2, also called ISO 10218-2 [18], defines safety requirements that should be followed while integrating a robot designed according to ISO 10218-1 in an industrial robot cell by a system integrator. Considering that the standard is a type C, which provides detailed safety design features for a specific type of machinery, accurate guidance on how safety should be ensured is also given for cobots. This standard has thus advocated for four types of safety modalities. However, due to the field novelty, the specification of such modalities did not limit itself to safety but recommended four types of collaborations considered inherently safe. Those modalities are known as Safety-Rated Monitored Stop (SRMS), Hand Guiding (HG), Speed and Separation Monitoring (SSM), and Power and Force Limiting (PFL). An illustration of those modalities is shown in Fig. 1.1. The first, SRMS, is the most elementary methodology for ensuring safety. With this approach, the robot and the human can work in a shared workspace, but the utilization of such an area is restricted to only one participant. This means the robot is driven to a halt when the human enters the shared space until the human leaves. However, to allow a natural operation's continuation, the robot stops through a Category 2 stop according to IEC 61800-5-2 [19]. Therefore, the drive power is not detached, and the machine stays in a secure interruption monitored through redundant software and electronics. The second, HG, allows the human to drive the robot in different positions by exerting forces on the machine. To guarantee safety, the robot operates in SRMS, and whenever the human enters the shared space, the hand-guiding device is switched on, and the human can guide the robot to defined positions. The third, SSM, is similar to SRMS but allows for expanded usage of the shared workspace. Specifically, the work zone is divided into three areas on the relative distance between the human and the robot, as shown by the color-coded Fig. 1.1. The robot operates at full speed whenever the human is in the

farthest area, denoted with green in the illustration. The robot reduces its speed once the relative distance decreases, denoted in yellow in the illustration. Finally, once the relative distance is small, the robot goes to Category 2 stop as in SRMS. A set of sensors or vision systems to detect the human and compare it with the robot's actual position is necessary to enable the relative distance measurement. Moreover, the zones must be computed with proper equations considering the robot's relative speed and stopping time, as illustrated by [20]. Last, PFL allows the robot to operate continuously in the shared zone, but its movement should be limited in motor power and force, allowing controlled collisions to happen. To enable PFL, the robot must embed dedicated hardware and software to deal with such controlled collisions. Additionally, the force and motor limits must be computed according to the collision area and the involved blunt element as described by [21, 22]. This process of calculating limits has been standardized under a technical specification (TS) released by ISO in 2016, the ISO TS 15066 [23]. Therefore, this first part of the decade ended with several guidelines on integrating cobots safely. In summary, those guidelines stated that whenever a cobot has to be introduced, a risk assessment should be conducted, and according to the level of the measured risks, proper mitigations following the collaboration modalities described above had to be introduced.

However, despite this effort for standardization, in nowadays robotic applications, technology adopters prefer not using cobots due to misunderstandings of the safety procedures [24], probably also driven by the cobots manufacturers campaigns which advocated for no requisite for proper safety analysis [25].

#### Nowadays

With several guidelines available on how to make and introduce safe cobots. The last five years of the 2020s saw a dozen new cobots released to the market [10], and for example, 2018 reported a cobots' year-on-year growth of 67% [4]. With this sharp increase and numerous new players in the market, cobot manufacturers started looking into ways to differentiate their products. One clinging idea across the manufacturers was to seek ease of use for their products as long this demonstrated to be a winning factor for the UR company [26]. More specifically, the ease of use targeted by these companies relates to the cobots' programming as long it is the most cognitively intensive task [10].

Initially, cobot programming followed the same programming methods as in standard industrial robots. This method, also known as lead-through programming, requires the human to use a teach pendant to program precisely defined robot motions

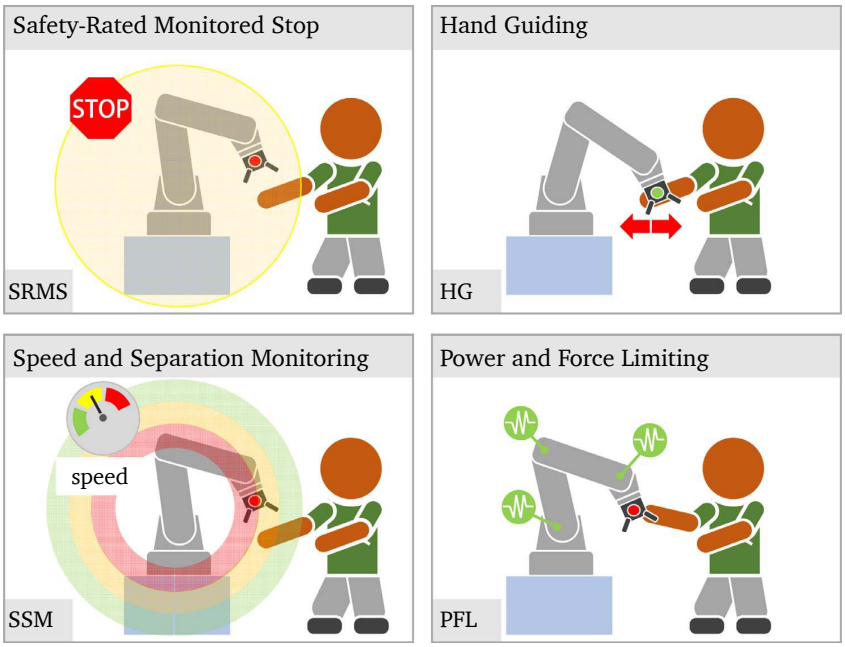



**Figure 1.1:** Representation of the four safety collaboration modalities described in ISO 10218-1 and ISO 10218-2. Reprinted from Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications, Mechatronics, Vol 55, Villani V. et al. [10], Copyright (2018), with permission from Elsevier.

as shown in Fig. 1.2. Specifically, the human has to jog the robot, i.e., move the robot in one or more of its axes and save positions or trajectories that satisfy the application's needs. Afterward, such information is played back by the robot controller to achieve robot autonomy. However, lead-through programming is considered dull, time-consuming, and must halt the machine for a long time. This method is best economi-

**Figure 1.2:** Traditional lead-through programming with the human holding and interacting with the teach pendant to program robot sequences or positions.

cally justified for production lines with large amounts of products (e.g., automotive). Therefore, cobots integrated variations of lead-through programming, alleviating some of the process's tediousness [27]. The first is off-line programming (OLP). In OLP, a software engineer in the design office can simulate robot behaviors through the digital representation of the robotic work cell. By these means, the engineer can walk through all the robotic steps, ensuring the task and its constraints are satisfied. Then, the simulation software generates the robot program, which can be downloaded to the machine. Although this method can reduce robot downtime, it requires expensive software. This method has been confined to robots acting as machine tools in specific processes (e.g., welding, polishing); one example is in the context of Wire Arc Additive Manufacturing (WAAM) [28]. The second is walk-through programming. Walk-through programming foresees a human operator to physically move the robot end effector (EE) through the desired trajectory and save key points. This method makes the robot programming intuitive because constant trajectory feedback is given. However, proper care should be given to safety and usability when using this method as long the robot compensates for its weight to allow the user to move the manipulator quickly. This is usually implemented through impedance control. Due to its nature, such control can have different parameters, and the literature demonstrated that a trade-off between accuracy and usability has to be made upon the application [29]. Therefore, walk-through programming is used alongside lead-through programming for mitigating this trade-off [30]. The third is learning from demonstration (LfD). It is a natural extension of lead-through programming. However, it bears the remarkable difference that LfD can learn trajectories and generalize them to new scenarios, i.e., object location change, by an appropriate trajectory encoding [31]. However, due to LfD novelty, its application still brings challenges like efficiency [31]. Considering these innovations, it is easy to understand that none can solve programming in all situations; therefore, there is still a need to bring different programming features together, thus limiting their drawbacks and improving usability [30].

Aside from programming, in the same five-year period, experiences from implementing cobots in industrial scenarios have been well described in the literature. The main field where cobots have been introduced is in automotive [10], mainly because this sector often introduces robots across the product life cycle [32]. Within this sector, cobots have been allocated to simple but repetitive dull tasks in the assembly line. More precisely, cobots have been used in the robot-as-tool scenario. Therefore, the machine is employed in use cases where it can help the operator lift heavy objects or avoid hazardous postures, which could lead to ergonomic hazards [10]. As a matter of fact, in this sector, cobots are used with a low level of autonomy [33]. In other manu-

facturing sectors, however, cobots are used slightly differently. Instead of introducing cobots as tools to relieve humans from hazardous tasks, cobots are also adopted to achieve higher productivity and flexibility. This is often the case for SMEs in different manufacturing sectors. More precisely, SMEs are searching for highly adaptable manufacturing systems, and cobots are an excellent asset for answering those needs [34]. Cobots introduced by SMEs are employed between different workstations to perform several tasks. The most famous examples are machine tending and material handling. In these applications, the cobot is either placed in front of a machine (e.g., lathe) or at a general-purpose workspace, and its task is to move raw materials and finished products, thus increasing the machine utilization rate. However, these applications require extensive programming knowledge, and SMEs are still looking for cobot technologies that can sense the environment and adapt to its changes to satisfy volatile production demands [35] without requiring robot experts for their reprogramming [36].

#### High-level Challenges

In the current landscape of cobots, adoption barriers need to be addressed to make them more accessible to a broader customer base, particularly SMEs. The previous overview of the current state of cobots highlighted the necessity for streamlined, user-friendly programming approaches that seamlessly incorporate various cobot programming techniques (*Challenge 1*). This approach should also simplify safety assessments, ultimately easing the adoption of cobots (*Challenge 2*). Furthermore, it should allow SMEs to deploy cobots that adapt to changing work environments and production demands (*Challenge 3.1*). However, these methods must ensure ease of use because SMEs' primary obstacle when using robot programming methods is the lack of access to robotic experts (*Challenge 3.2*).

In light of these challenges, this research proposes a comprehensive suite of intuitive programming methods grounded in spatial interactions, which non-experts can efficiently utilize to streamline robot programming. Additionally, such programming methods are designed to seamlessly incorporate safety considerations into the workflow. This is achieved by adhering to a standardized programming process that facilitates documentation and robust safety assessment tools.

However, proposing intuitive programming methods is complex, requiring an in-depth examination of their usability and accessibility. Therefore, this work places a particular emphasis on thoroughly investigating and refining these aspects to ensure that the programming solutions are innovative but also practical and user-friendly.

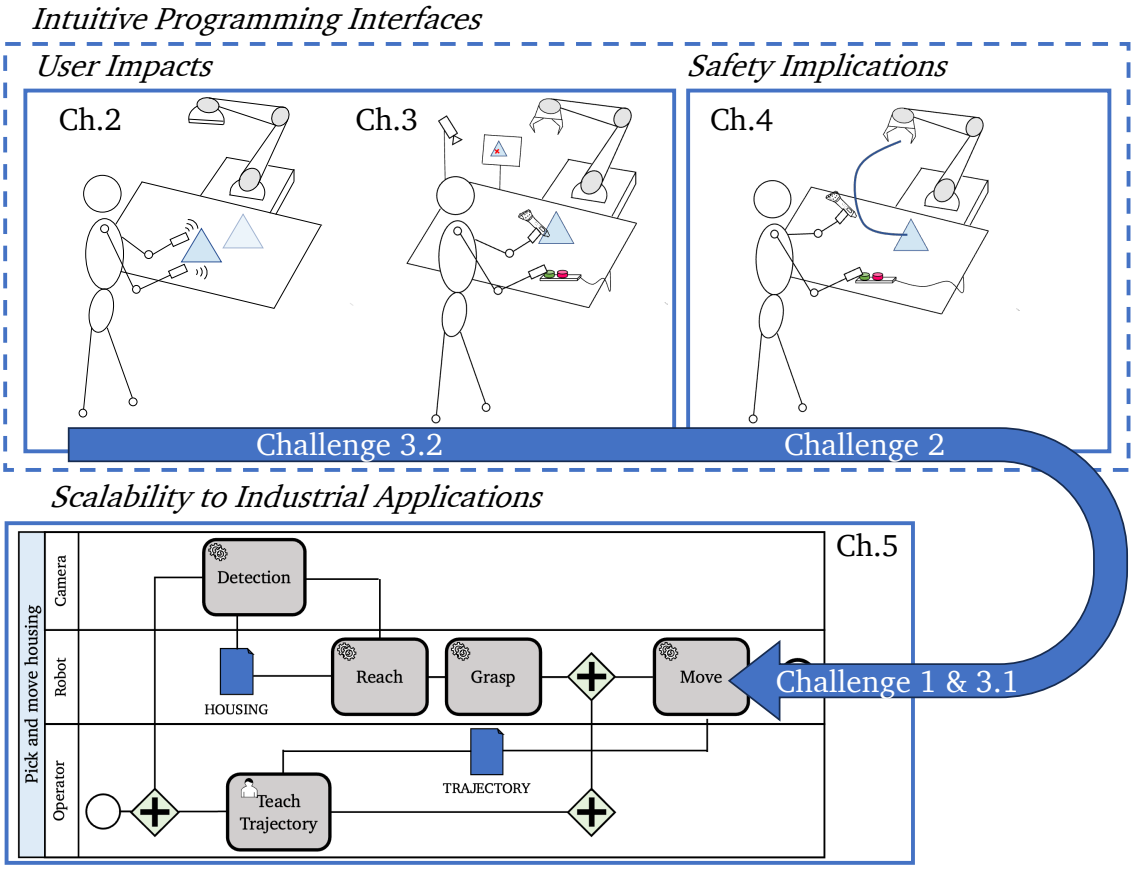



**Figure 1.3:** Thesis structure with relevant chapters and addressed challenges. The thesis follows the workflow a user should adopt for the programming of a collaborative robot, initially starting from Chapter 2 with the personalization of the workspace. Then, the user defines the grasping location as defined in Chapter 3. Next, the trajectory that the robot has to follow is programmed as defined in Chapter 4. Finally, the complete workflow of the robot is described and assessed in terms of safety as defined in Chapter 5.

## 1.3 Overview of the Thesis

As highlighted in the section above, several challenges exist for introducing an integrated workflow-based programming method for cobot programming with spatial user interactions. More precisely, such interactions must be easy for a broad customer base. In order to present the contents of this research, this work is structured in three main areas following the workflow a user would need to adopt for programming a cobot, as shown by Fig. 1.3. This section describes each of the three main areas and their contributions in detail.

### 1.3.1 Intuitive Programming Interfaces and User Impacts

As highlighted above, when robot programming should be integrated, it should be easy for a vast pool of users and allow simple modifications without requiring deep robotics expertise as outlined in *Challenge 3.2*. To present the research work behind this line, chapters 2 and 3 present novel, simple programming modalities to allow the operator to program behaviors of cobots when dealing with pick and place operations.

Chapter 2 introduces a programming method based on computer vision (CV) to allow the operator to define where the parts shall be located to increase the level of robotic acceptance, as long as task personalization is essential [37]. However, the chapter investigates how operator decisions influence the robot and the operator's well-being to understand how personalization should be introduced as a programming method. Some of the material presented in this chapter is also published in:

- Pantano, M., Yang, Q., Blumberg, A., Reisch, R., Hauser, T., Lutz, B., Regulin, D., Kamps, T., Traganos, K., and Lee, D., "Influence of task decision autonomy on physical ergonomics and robot performances in an industrial human–robot collaboration scenario," 2022, Frontiers in Robotics and AI, Volume 9.

- Pantano, M., Curioni A., Regulin D., Kamps, T., and Lee, D., "Effects of Robotic Expertise and Task Knowledge on Physical Ergonomics and Joint Efficiency in a Human-Robot Collaboration Task," 2023, IEEE-RAS International Conference on Humanoid Robots.

Once the task has been personalized, the next step is to allow users to define locations where objects should be grasped. Chapter 3 introduces a spatial programming method for the definition of such locations. More precisely, considering that industrial users must be able to define such locations considering task requirements and reliability [38], the chapter proposes a novel metric to fuse an initial grasp guess given by an operator with a state-of-the-art neural network (NN) to ensure reliable grasps which respect users' suggestions. The section investigates how this definition influences grasp point robustness and user acceptance. The material presented in this chapter is partially published in:

- Reisch, R. T., Pantano, M., Janisch, L., Knoll, A., and Lee, D., "Spatial Annotation of Time Series for Data Driven Quality Assurance in Additive Manufacturing," 2022, 17th CIRP Conference on Intelligent Computation in Manufacturing Engineering (ICME).

- Caporali, A., Pantano, M., Janisch, L., Regulin, D., Palli, G., and Lee, D., "A Weakly

Supervised Semi-automatic Image Labeling Approach for Deformable Linear Objects," 2023, IEEE Robotics and Automation Letters (RA-L), Volume 8.

- Pantano, M., Klass V., Yang Q., Sathuluri A., Regulin D., Janisch L., Zimmermann M., and Lee, D., "Simplifying Robot Grasping in Manufacturing with a Teaching Approach based on a Novel User Grasp Metric," 2023, 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM).

### 1.3.2 Intuitive Programming Interfaces and Safety Implications

Once the user has partly personalized the task, it is essential to understand how this will impact the level of safety to understand the impacts for *Challenge 2*. More precisely, the safety standards describe how robot movements should be considered when a cobot can move outside a caged environment. In this regard, analyzing the robot's trajectories is crucial for defining safe behaviors. Chapter 4 studies how integrating operator programming affects a robot's safety. This chapter is partly based on the following publications:

- Pantano, M., Regulin, D., Lutz, B., and Lee, D., "A human-cyber-physical system approach to lean automation using an industrie 4.0 reference architecture", 2020, Procedia Manufacturing, Volume 51.

- Pantano, M., Blumberg, A., Regulin, D., Hauser, T., Saenz, J., and Lee, D., "Design of a Collaborative Modular End Effector Considering Human Values and Safety Requirements for Industrial Use Cases," 2022, Human-Friendly Robotics 2021, Springer Proceedings in Advanced Robotics, Volume 23.

- Pantano, M., Schmidt, M., Bolano G., Schulenburg, E., Regulin, D., and Saenz, J., "Analyzing the Influence of Self-defined Trajectories on Safety and Task Ownership: An Empirical Study," 2023, industrial paper at the 19th IEEE Conference of Automation Science and Engineering (CASE).

### 1.3.3 Scalability to Industrial Applications

Finally, to ensure that the proposed personalized, safe workflows can be successfully integrated into SMEs considering the requirement for a single programming workflow, chapter 5 studies how the presented novel interaction methods can be brought together as required by *Challenge 3.1* and *Challenge 1*. More precisely, this chapter outlines

a proposal for standard nomenclature to describe robot motions and perform a risk assessment. Additionally, the user acceptance of a programming method based on the standard nomenclature is investigated. This chapter is partly based on the following publications:

- Pantano, M., Eiband, T., and Lee, D., "Capability-based Frameworks for Industrial Robot Skills: a Survey," 2022, 18th International Conference on Automation Science and Engineering (CASE).

- Pantano, M., Pavlovskyi, Y., Schulenburg, E., Traganos, K., Ahmadi, S., Regulin, D., Lee, D., and Saenz, J., "Novel Approach using Risk Analysis Component to Continuously Update Collaborative Robotics Applications in the Smart, Connected Factory Model," 2022, Applied Sciences, Volume 12.

## 1.4 Contributions

As previously highlighted in this work methods to reduce barriers of adoption of cobots for SMEs are addressed. More precisely the contributions of this work can be listed as follows.

#### Intuitive Programming Interfaces and User Impacts

In chapter 2, a method for allowing users to program robot end-locations is proposed, and its effects are analyzed to tackle *Challenge 3.2*. Therefore, the main contributions of this chapter are the in-depth analysis of how the user's robotic background can influence human-robot collaboration and an evaluation framework for estimating physical ergonomics via camera inputs. This contribution shows that user background must be considered when allowing users to design their workspace to ensure that different robotic expertise levels are supported.

In chapter 3, a method to allow operators to propose candidate grasp locations via a spatial sensor is proposed, and its impacts are examined to tackle *Challenge 3.2*. Consequently, the main contributions in this chapter are the NN based methods to fuse spatial sensor data with visual data accounting for user and calibration errors. These methods show that neural network-based techniques to fine-tune user initial label points on visual data can be beneficial and that the quality of the selected labels is improved to automatically grasp location selections while reducing the need for robotic expertise.

#### Intuitive Programming Interfaces and Safety Implications

In chapter 4, a method to program trajectories using a spatial sensor and its impacts on safety and user acceptance are presented to tackle *Challenge 2*. Therefore, the main contributions of this chapter are a method to correct inaccuracies of a spatial sensor based on photodiode sensors and, additionally, insights on how user-defined trajectories might influence the safety of the applications considering different levels of user robotic background. This chapter shows that robotics expertise plays a significant role in programming trajectories that are considered safe and that a spatial sensor based on photodiode sensors should not be calibrated only with conventional calibration techniques.

#### Scalability to Industrial Applications

In chapter 5, a workflow-based programming method that allows integrating different programming modalities with safety evaluations is presented to tackle *Challenge 3.1* and *Challenge 1*. The main contribution of this chapter is a standard nomenclature and structure based on a business process model notation for robot programming. This chapter shows that having a standard nomenclature and structure in programming facilitates user programming when compared with state-of-the-art methods and also allows the creation of documentation relative to the safety evaluations.

## 1.5 Notes for the reader

#### Chapters' structure

This work comprises four main chapters, delineating the procedural steps users should follow when programming a robot in an SME. To enhance readability, each chapter adheres to a structured format. It begins with an exhaustive review of the prevailing state-of-the-art, which is succeeded by preliminary research. Subsequently, the methods necessary to address the identified challenges are outlined. Lastly, primary evaluations of these methods are presented to demonstrate their impact on overcoming the challenges. Thus, readers are encouraged to comprehensively analyze each chapter to understand various topics correctly.

#### Statistical analysis

Statistical analysis is often conducted when analyzing different results of this work. More precisely, when such analysis is performed, the primary evaluation often is to

ensure that two populations have a different average value with different statistical tests depending on the distributions. When the data is presented, the applied method is reported; if the method is omitted, the t-test should be considered. Aside from this, it is essential to underline that statistical results are also reported in the results graph. Asterisks are used to report the level of the $p$ value to report statistical significance. The selected convention is as follows: ns: $p > 0.05$, *: $p \leq 0.05$, **: $p \leq 0.01$, ***: $p \leq 0.001$, ****: $p \leq 0.0001$.

Chapter 2

# Influences of User-Defined Co-Assembly Locations on Operator Well-being and Task Performances

Contents



Personalizing the human-robot collaboration can improve the application intuitiveness and ownership for the operator. Therefore, there is a need for research on how to allow the integration of personalization and analyzing its effects on collaboration. This chapter studies how allowing operators to define co-assembly locations for robot motion influences the users' tasks and physical well-being. The contents of this chapter are partly based on the following publications:

- Pantano, M., Yang, Q., Blumberg, A., Reisch, R., Hauser, T., Lutz, B., Regulin, D., Kamps, T., Traganos, K., and Lee, D., "Influence of task decision autonomy on physical ergonomics and robot performances in an industrial human–robot collaboration scenario," 2022, Frontiers in Robotics and AI, Volume 9.
- Pantano, M., Curioni A., Regulin D., Kamps, T., and Lee, D., "Effects of Robotic Expertise and Task Knowledge on Physical Ergonomics and Joint Efficiency in a Human-Robot Collaboration Task," 2023, IEEE-RAS International Conference on Humanoid Robots.

The structure to present the contents of this chapter is as follows. Initially, the motivation behind this work is outlined. Second, an overview of the current state of the art concerning the personalization of robotic tasks is given, and the need for a higher degree of human-robot task personalization is discovered. Next, with the discovered need, a method for allowing users to personalize a robotic task by specifying co-assembly locations is presented. Third, the results of two studies that try to underline the relationship between users' background and satisfaction with the robot personalization are presented. Finally, the chapter gives conclusions and concluding remarks on how the identified results can benefit the future of personalized robotic applications.

## 2.1 Motivation

In human-robot collaboration (HRC), humans and robots should collaborate. However, the design of this collaboration can be challenging depending on the task or the human operator's background. To overcome some of these issues, including the human operator in the design can be beneficial [37]. By including the operator, the creation of good mental models of the application is encouraged [39–45]. One recent example proving the benefit of good mental models can be seen in the work of [46]. In this work, the authors found that greater satisfaction in human-robot interaction can be achieved if operators design their sequence of tasks. However, no investigation in industrial scenarios was performed in the study. Therefore, there is a need for further research in this sector. By looking at the adoption of HRC in industry, the applications that could benefit the most from those design suggestions are the ones known as cooperation and collaboration [47]. In these modes, teammates (i.e., human and robot) perform tasks in a shared workspace on different components or the same ones [48]. If these modes are successfully implemented, several benefits can be achieved. One of those is the improvement of operators' physical ergonomics [49]. Therefore, to better grasp the connection between mental models, task design, and performance in industrial scenarios, there is a need for more investigation.

## 2.2 Related Works

### Evaluation of Physical Ergonomics

When considering operators' physical ergonomics, an evaluation must be carried out. These assessments can be based on two principles. First, a posture-based analysis, in this assessment, deviations of joint angles from the natural standing pose are observed, and, upon their magnitude, scores representing the exposure to hazards related to musculoskeletal disorders (MSDs) are given. Second, a biomechanics-based analysis, in this assessment, repetitions of actions involving a load, namely pushing, pulling, carrying, lowering, or lifting, are counted to obtain the exposure to risks of MSDs [50]. However, those principles provide only methods to record measurements, and task observations must be made. Two classes of task observations are known. On one hand, simulations via digital human models (DHM) [51–53]. On the other hand, in-situ process studies [54]. In the former, digital representations for the task must be available [55]. Often, this happens when planning production lines that will run for several years, like an automotive assembly line [56, 57]. Consequently, commercial tools are available (e.g., Siemens Tecnomatix™ Process Simulate). However, they bear a high degree of complexity; therefore, they are not often used by SMEs [9]. In the latter, digital simulations are not used; instead, experts are studying the task and providing the level of exposure to MSDs by actual observations at the factory [58]. As a consequence, this type of evaluation can carry errors due to its nature [59]. However, it is easier to introduce in SMEs and does not require high levels of digitization.

### Physical Ergonomics in HRC

Posture-based assessments and in-situ observations have been widely adopted to estimate and improve HRC's physical ergonomics. Rahal et al. [60] propose a haptic control based on the human arm's inverse kinematics (IK) to derive user comfort and change the control strategy in a teleoperation use case. Through this approach, they achieve lower muscular loading. Shafti et al. [61] present an algorithm that reduces operators' armload by controlling the manipulator positioning following muscular and physical ergonomics assessments in hand-over interactions. Makrini et al. [62] demonstrate that task allocation based on physical ergonomics can improve the overall working conditions of the operators in an assembly use case. However, their outcomes were influenced by a human posture algorithm in sub-optimal operation. A similar approach, but with an improved visual algorithm based on OpenPose [63], is proposed by [64]. In this work, they demonstrate that a physical ergonomics assessment can be used to derive compliant robot motions that follow the postures of different operators, thus reducing the operator joint torque overloading in hand-over tasks. Such state-of-the-art

approaches demonstrate how exposure to ergonomics hazards can be reduced when the robot hands over tools to the operator. However, this appears barely in industrial scenarios. More precisely, as highlighted in the introduction, collaborative robots are being employed by SMEs for material handling. Therefore, handing over tools is not the primary concern for applications in SMEs. Therefore, investigations in other tasks are needed.

#### Task Design

As highlighted in previous sections, to improve cobot acceptance, users should be able to design their task sequence [46]. However, this study was performed with ideal fictitious scenarios. A following study involving real robots identified that such design of the task sequence is extremely dependent on the task and operator's subjective preference [65]. Therefore, operators should be empowered to choose the design of a human-robot collaborative task [65, 66]. Different robot programming modalities are available to empower operators to influence the task design. However, non-experts do not always accept conventional methods, and usability can strongly depend on the manufacturer interface [30, 67]. Interfaces that mimic natural behavior can decouple users from robots' requirements. Recent market examples are Tracepen™ from Wandelbots[1] and Mimic™ from Nordbo Robotics[2]. However, such systems focus more on the robot's trajectory than the objects' location. Therefore, there is a need to investigate other task design approaches with natural interfaces. One promising approach is using visual data coming from cameras [68]. However, no example is present in the literature to the author's knowledge.

## 2.3 Definition of Co-Assembly Locations via a Visual Interaction

As highlighted from the state-of-the-art, there needs to be more research in investigating how operator-designed tasks influence physical ergonomics. To study these scenarios, a method for allowing operators to program tasks via the specification of co-assembly locations is proposed. This approach is based on identifying components with an object detection algorithm and a parameterizable robot IK for component movements. Additionally, to study the influence of human exposure to MSDs, posture-based physical ergonomics evaluated with CV is introduced. This chapter describes these different components and their connection to the proposed study.

[1] https://wandelbots.com (accessed on 22/12/2022)
[2] https://www.nordbo-robotics.com (accessed on 22/12/2022)

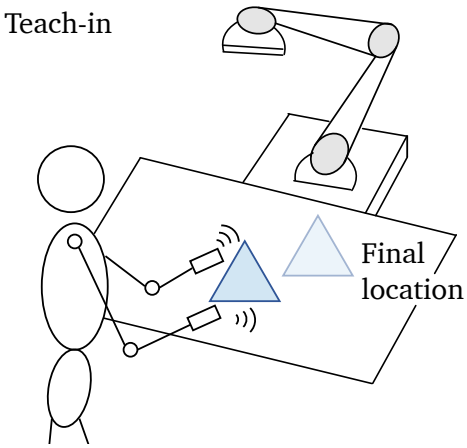



**Figure 2.1:** Illustration of the movement-based interaction during teach-in. The operator can personalize the task by physically moving a workpiece toward a final co-assembly location from which the robot and the operator will perform a subsequent collaboration.

#### Envisioned Interaction

A movement-based interaction is selected to allow a human operator to personalize the task. More precisely, to personalize the task, the human has to first position the objects in a workspace location that best suits the task and themselves in an operation also denoted as teach-in as shown in Fig. 2.1. Afterward, a camera records the object's location, and the robot collaborates with the human for task accomplishment. This movement-based interaction is selected as long it allows the operator to select a defined spatial organization of the workspace, which proved beneficial for the perceived work efficiency in [69].

#### Object Detection and Adaptive Robot Inverse Kinematics

A CV pipeline based on an object detector is introduced to enable the robot to sense the environment and locate the position of objects in the workspace. The pipeline goal is to obtain parts' location in robot coordinates through a camera, as shown from Fig. 2.2. To solve this problem, the You Only Look Once (YOLO) NN algorithm [70] has been selected due to its good performance with few training data. This algorithm can predict a Bounding Box (BB) describing where an object in an image is through its pixel coordinates, thus allowing an operator to place parts at preferred positions. However, these coordinates in 2D space must be transformed into 3D ones to allow the robot to reach the detected components. The camera was calibrated to the robot to compute this transformation using the method in [71], and the 2D-pixel coordinates were transformed into 3D ones with Eq. 2.1.

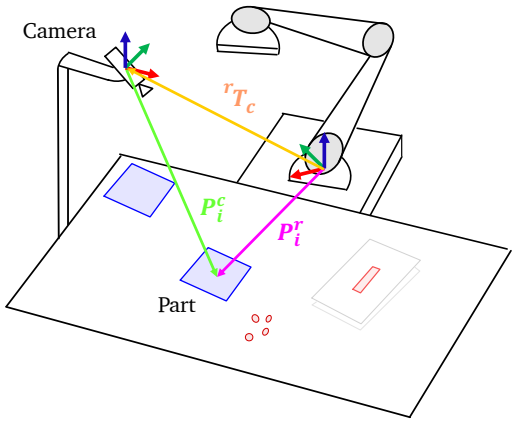



**Figure 2.2:** Illustration of the different coordinate frames in the use case. To enable the robot to handle a part in the workspace, its location computed in pixel coordinates, denoted with $P_i^c$, must be transformed to robot coordinate frames, denoted with $P_i^r$. This transformation is done through the homogeneous matrix ${}^rT_c$. Reprinted and adapted from Influence of task decision autonomy on physical ergonomics and robot performances in an industrial human–robot collaboration scenario, Pantano M. et al. [72], Copyright (2022), with permission from Frontiers in Robotics and AI.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = K^{-1\,r}T_c \begin{bmatrix} u' \\ v' \\ w' \\ 1 \end{bmatrix} \tag{2.1}$$

Where $[x, y, z]$ is the 3D point representing the position of a component in robot coordinate frames, $K$ are the intrinsic camera parameters, ${}^rT_c$ is the transformation between the robot coordinate frame to the camera coordinate frame and $u, v$ are pixel coordinates with $u = u'/w'$, $v = v'/w'$ and $w'$ as scaling factor. Henceforth, with the parts' coordinates in the robot system, the machine can interact with them for the envisioned task as described in the previous section.

### Physical Ergonomics Assessment

A posture-based assessment with in-situ observations with a camera is selected to calculate exposure to physical ergonomics hazards. More precisely, the rapid upper limb assessment (RULA) [73] is employed due to its superior capability to evaluate risks ranging from low to high when compared with other assessment methods [74]. However, according to this assessment, joint angles must be known for scoring postures. RULA is implemented with the human limb detection algorithm OpenPose [63] for this calculation. Therefore, angles between limbs are calculated with Eq. 2.2.

$$\theta = \cos^{-1}\left(\frac{\mathbf{a}\cdot\mathbf{b}}{\|a\|\|b\|}\right) \tag{2.2}$$

Where **a**, **b** are the vectors representing two limbs obtained through the human limb detection, and $\theta$ is the angle between those two limbs. Afterward, postures are scored via look-up tables, which assign ordinal values to each part of the body, knowing the angles' quantitative information (assuming that non-repetitive actions and loads under 0.5kg are present due to the task design). For clarity, an example of this conversion is reported here. Suppose a task where an operator has the upper arm with an angle between ±20°, the lower arm with an angle above 100° from the natural position of lower and upper arms aligned with the torso, and wrist without twisting. This leads to an intermediate risk level of 2 for the wrist-arm compound. Then, if the other limbs are considered where the neck and trunk are straight and the leg well supported, an intermediate risk level of 3 for the neck, trunk, and legs compound is obtained. Therefore, by considering both levels and looking at the second row and third column of the RULA evaluation tables, a final risk level of 3 for this posture can be obtained. Therefore, this posture assessment can be used in HRC tasks as long the human operator is monitored with cameras.

## 2.4 Effects of User-Defined Co-Assembly Locations on the Well-being and the Task

Two sets of studies are performed to investigate the effect of this visual programming method on human well-being. The first is a preliminary study that tries to gather some first insights into the first-order effect of the method. The second details how different levels of operator experience influence the task design and how these considerations can be considered for future studies.

### 2.4.1 Preliminary Study on the First-Order Effects

In this study, the goal is to identify if leaving decision autonomy freedom can benefit the operators. Therefore, in this first study, two hypotheses are made. The first concerns the operator's perceived level of autonomy. The second is the measured physical ergonomics. More specifically, the hypothesis can be stated as follows:

- *Hypothesis 1 (H1)* If the operator can decide the object location, the operator perceives a higher task decision autonomy

- *Hypothesis 2 (H2)* if the operator decides by itself the object location, the operator's physical ergonomics is the best

This section presents the experiment conducted to find answers to these questions.

#### Industrial Use Case and Envisioned Interaction

For the scope of this study, a use case concerning a stack preparation for a sintering process is considered. In this task, an operator must stack different materials onto each other to create a pile, which will go into a furnace for heat treatment. This task has been selected for two reasons. First, nowadays, tasks are performed manually. Second, in the manual process, operators displace objects at their will during the operation. Therefore, by adopting the proposed interaction, operators can, on the one hand, exert their will on part displacement. On the other hand, they can be helped by the robot with some dull, repetitive actions. Hence, they complement each other. Therefore, the placement of the parts is customizable. This customization allows the human to select the material's co-assembly locations on an x-y plane before the interaction. Afterward, the collaboration unfolds, and each actor (i.e., robot, human) performs its duties in the use-case. This collaboration unfolds with five sub-steps. In step 1, the robot moves the fragile part towards the stack. In step 2, the user places three separators on the stack. In step 3, the robot places a separating sheet on the stack while the user exposes the following fragile part. In step 4, the robot moves the fragile part while the user ensures the stack is straight. In step 5, the user exposes another fragile part so the process can start again. For clarity, this is shown in Fig. 2.3.

#### Experiment Design

A study with two levels of interaction has been envisioned to investigate how the freedom of placing parts influences the operator's well-being. This design ensures that the investigation compares the effects of the co-assembly teach-in proposed in this section against a typical interaction. With this experimental design, two variables are manipulated. On the one hand, the level of personalization (user versus apriori defined). On the other hand, the level of collaboration familiarity (first versus second interaction). The following abbreviations are used for these interactions: *usr* for the user-defined interaction, *std* for the apriori-defined interaction, *Interaction I* for the first interaction in the experiment, and *Interaction II* for the second one. Using these abbreviations, the experiment procedure was as follows. At first, users were introduced to the task by showing an explanatory video with the task. Then, any doubts in the collaboration were answered, and the familiarization phase ended with the users replying to

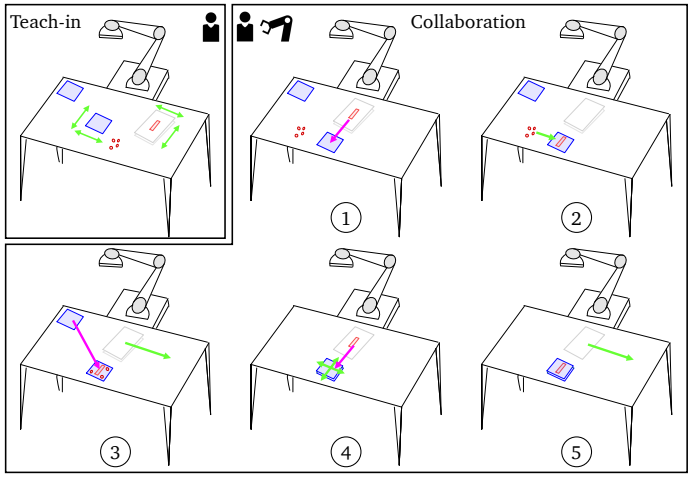



**Figure 2.3:** Representation of the unfolding interaction. The cooperation is composed of the teach-in phase and collaboration phases. The human operator takes over the actions marked with light green. The robot takes over the actions marked in pink. Reprinted and adapted from Influence of task decision autonomy on physical ergonomics and robot performances in an industrial human–robot collaboration scenario, Pantano M. et al. [72], Copyright (2022), with permission from Frontiers in Robotics and AI.

a questionnaire regarding their demographics and informed consent. Next, the user performed the collaborative task with the two methods. After each interaction, the level of decision autonomy was measured through two subjective measures from the Work Design Questionnaire (WDQ) [75] with a Likert scale. Additionally, during the interaction, the robot and operator were monitored to calculate the objective measures of exposure to ergonomic hazards and the task time. For clarity, the study design is shown in Fig. 2.4.

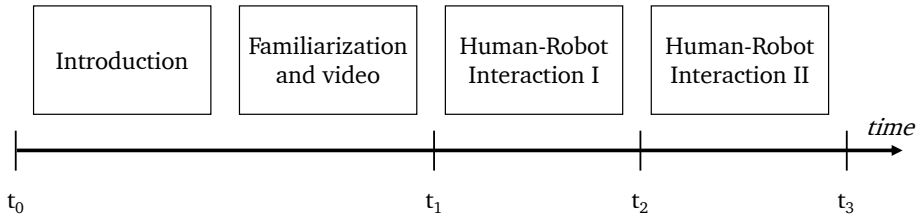
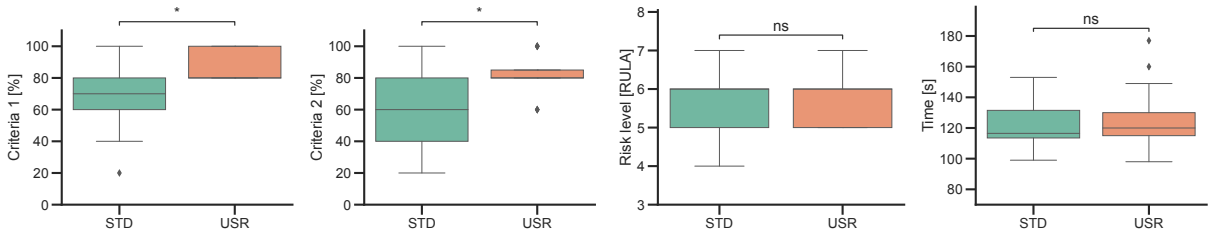



**Figure 2.4:** Experiment design schema. At first, the users were introduced to the task, and a video of the interaction was shown. This initial phase ended at $t_1$, where informed consent and general demographics were collected. Then, the two interactions unfolded, and during them, the RULA and the task times were collected. Finally, at $t_2$ and $t_3$, the perceived level of decision autonomy with WDQ was gathered. Reprinted and adapted from Influence of task decision autonomy on physical ergonomics and robot performances in an industrial human–robot collaboration scenario, Pantano M. et al. [72], Copyright (2022), with permission from Frontiers in Robotics and AI.




**(a)** Autonomy criteria I **(b)** Autonomy criteria II **(c)** RULA **(d)** Time

**Figure 2.5:** Results of the experiment investigating the physical ergonomics. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed. The first two graphs report the outcomes of the WDQ criteria. The third graph shows the recorded physical ergonomics. Finally, the last graph shows the time used for the task execution. From the evaluation, it is possible to denote that *usr* achieved a higher level of decision autonomy while maintaining the same level of physical ergonomics and required time.

### Results

As highlighted in the introduction, this study aims to identify how the definition of object location influences decision autonomy and the operator's physical ergonomics. In the study, 17 individuals not trained in physical ergonomics participated, age M = 33.05 yrs, SD = 12.89, height M = 180.11 cm, SD = 9.26. One did not perform the task correctly, and the data was discarded from the analysis. Regarding task decision autonomy, responses to two WDQ criteria were compared. For the sake of clarity, the WDQ questions were respectively: "The system gives me a chance to use my personal initiative or judgment in carrying out the work," and "The system provides me with significant autonomy in making decisions" from now on denoted as criteria one and criteria 2. In this evaluation, the recorded results for criteria 1 were M=68.75%, SD=23.06 and M=86.25%, SD=9.57 for *std* and *usr*. For criteria 2, the results were M=62.50%, SD=27.20 and M=81.25%, SD=13.60 for *std* and *usr*. Therefore, considering that the results proved statistically significant, allowing the part placement can improve the level of task decision autonomy. The measured levels of physical ergonomics via the RULA assessment were compared. The results were M=5.74 ergonomic hazard, SD=0.56, and M=5.78 ergonomic hazard, SD=0.64 for *std* and *usr*, respectively. However, considering that no statistical difference was found, it is possible to conclude that the part placement did not influence the level of physical ergonomics in this experiment. Finally, regarding the taken time, the outcomes were M=123.56 s, SD=17.29 for *std* and M=125.56 s, SD=21.44 for *usr*, which was not found statistically different. Therefore, part displacement did not influence the required task time. For clarity, the results with statistical significance are displayed in Fig. 2.5.

### Discussion

**Table 2.1:** Correlation analysis of the metrics related to the user group which saw *std* as *Interaction I* calculated with the Pearson correlation coefficient. Significant correlations ($p < 0.05$) are marked with an asterisk (*). Metrics definitions have been abbreviated with the following. Erg. for risk level, Aut. for autonomy representing the average response to criteria 1 and 2.

| | M | SD | Erg. *std* | Erg. *usr* | Time *std* | Time *usr* | Aut. *std* | Aut. *usr* |
|---|---|---|---|---|---|---|---|---|
| Erg. *std* | 5.57 | 0.60 | - | | | | | |
| Erg. *usr* | 5.69 | 0.54 | 0.59 | - | - | | | |
| Time *std* | 129.13 | 17.63 | −0.16 | 0.53 | - | | | |
| Time *usr* | 123.50 | 23.29 | 0.52 | 0.86* | 0.59 | - | | |
| Aut. *std* | 2.75 | 1.06 | 0.49 | 0.02 | −0.17 | 0.29 | - | |
| Aut. *usr* | 3.50 | 0.72 | 0.02 | −0.36 | −0.38 | −0.05 | 0.70 | - |

**Table 2.2:** Correlation analysis of the metrics related to the user group which saw *usr* as *Interaction I* calculated with the Pearson correlation coefficient. Significant correlations ($p < 0.05$) are marked with an asterisk (*). Metrics definitions have been abbreviated with the following. Erg. for risk level, Aut. for autonomy representing the average response to criteria 1 and 2.

| | M | SD | Erg. *usr* | Erg. *std* | Time *usr* | Time *std* | Aut. *usr* | Aut. *std* |
|---|---|---|---|---|---|---|---|---|
| Erg. *usr* | 5.92 | 0.70 | - | | | | | |
| Erg. *std* | 5.83 | 0.52 | 0.09 | - | - | | | |
| Time *usr* | 127.63 | 20.80 | −0.62 | 0.30 | - | | | |
| Time *std* | 118.00 | 16.10 | 0.07 | 0.23 | 0.66 | - | | |
| Aut. *usr* | 4.41 | 0.44 | −0.22 | 0.13 | −0.21 | 0.11 | - | |
| Aut. *std* | 3.91 | 0.88 | −0.05 | 0.40 | 0.15 | 0.13 | 0.60 | - |

From the results, it is possible to understand that allowing for user-defined co-assembly locations can increase the level of task decision autonomy while not altering the task time and the exposure to ergonomic hazards. Therefore, considering in more detail the hypotheses stated before, it is possible to accept *H1* saying that allowing the operator to decide about part placement increases task decision autonomy. Unfortunately, such positive results cannot be recorded for *H2* as when the operator decided the co-assembly locations, the physical ergonomics was not the best. More precisely, the statistical difference across the two samples was not found for the ergonomic hazard. Therefore, *H2* must be rejected as long the risk level is similar in the two interactions, and no reduction of the ergonomic hazard is achieved by leaving the operators with a high level of task decision autonomy. A study of correlations between recorded measures was conducted to better understand how task decision autonomy influenced the interaction. Therefore, the Pearson correlation coefficient was calculated among the different metrics. Results of the analysis are shown in Tab.2.1 and 2.2. In line with

the previous findings, task decision autonomy has always been more significant in the *usr* interaction. Additionally, this analysis shows that the time taken to conduct the task has always been lower in *Interaction II*. This is probably related to the fact that repetition of an error-free interaction can increase familiarity and acquaintance, as discovered by [76]. Interestingly, in the case where *std* was the *Interaction I*, a significant positive correlation between time and risk level is recorded. Therefore, users in *usr* were driven to maintain a hazardous posture for an extended period. This might be related to the closer distance to the work table. Despite these additional insights, it is not possible to understand how data for *H2* was affected. Therefore, an additional analysis of the parts' positioning was performed. More precisely, the x and y displacement on the worktable was monitored, and the Euclidean distance between the position in *std* and *usr* was measured. The results are shown in Fig. 2.6. From the figure, it is possible to understand that user selection plays an important role. Each user selecting the displacement of a specific part chooses similar delta values for both parts. A correlation analysis among the Euclidean distances of the two parts is conducted to check if there is a correlation between the displacement of one part and the one of the second part by a single user. The correlation returned a significant value ($p < 0.05$) c = 0.82. Therefore, it is possible to understand that there is a specific co-assembly location selection depending on the user's background. During the test, users were requested with an open question format why the parts were displaced with such positions. Often, the recurring answers were, "I placed the part closer to the robot so I can be faster, although this leads to a bad position for me," or "I placed the parts closer to me so I can handle them better." Therefore, the user background could influence the robot's vicinity. However, further analysis on this topic needs to be done.

#### Limitations

This study analyzed the impact of user decisions in a human-robot collaborative task on user physical ergonomics. For such analysis, the users were in charge of displacing parts on the x-y plane according to their beliefs. However, during the study, some limitations were encountered. More specifically, those were as follows. Initially, when users were instructed on how to place the parts, no clear goal was given to them aside from completing the task itself. Therefore, the users had to consciously know what they should try to optimize. This led to high variability in parts' displacement, and the users randomly selected the parts' position. This might have led to unwanted results. The reported average RULA levels are also relatively high for normal standing operations. During the task, it was observed that the table on which users were working was

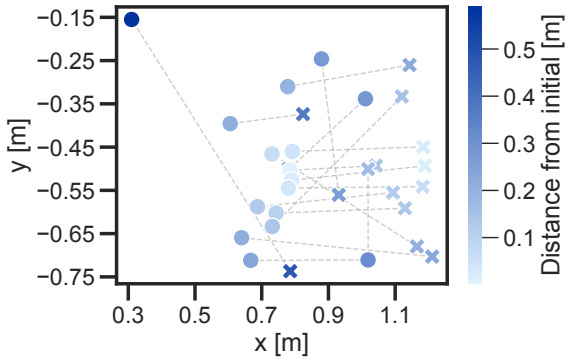



**Figure 2.6:** Parts' positioning on the work table with respect to the robot base. The plot displays the x-y displacement of the two parts, one denoted with X and the other with O, for each user participating in the study. Additionally, the Euclidean distance of the parts' position from the initial placement in *std* is shown through a colored hue. Last, dotted lines show how a single user selected the placement of the two parts in the conducted test. From this figure, it is possible to understand that a correlation might be present between the delta values of the two parts given by the user preference due to the similar hue colors between connected points.

low compared to typical working tables, and this led to elbow angles higher than 90 degrees, which in the RULA evaluation leads to high scores. Therefore, these limitations should be addressed to better understand the impact of user decisions on physical ergonomics and task performance. The following section presents another study where these limitations were addressed.

### 2.4.2 Follow-up Study with Focus on the User Background

As highlighted in the previous study, no relation between task decision autonomy and improvement in physical ergonomics was found. However, the study identified that different groups of people are selecting positions with contrasting rationales. More precisely, some people preferred to place parts close to the robot as long they wanted to simplify the task for the machine. Others preferred to place the parts close to them to improve their well-being. In other words, some participants tried to minimize their effort by reducing travel distance, thus maximizing the robot's travel path. In contrast, others maximized their effort by incrementing the traveling distance, thus minimizing the robot's travel path despite the first group adhering to the findings of the literature, which states that humans try to minimize the traveling distance [77, 78]. The other group decided on a contradictory option, considering their travel path. From the previous study, additionally, it was observed that the people belonging to the group deciding on the contradictory option appertained to individuals with robotics experience. Thus, considering the scarce literature, a follow-up study is necessary to investigate

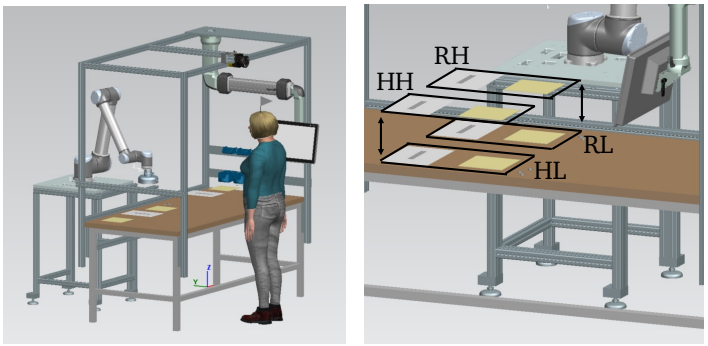



**Figure 2.7:** Configuration of the simulation engine. The simulation uses Tecnomatix™ Process Simulate Human Modeling and Simulation. To properly assess the collaboration, four levels of part displacement are defined (see figure on the right). Each level has a different impact on the human and robot effort. To highlight the difference between the four displacements, the configurations are named with the first letter stating to which actor they are closest and the second letter for the height from the table. Those configurations are then Robot-Low (RL), Robot-High (RH), Human-Low (HL), and Human-High (HH).

how robotics experts make spatial decisions. To better frame this study, one additional hypothesis is formulated considering the recent literature on joint-task efficiency, which states that when humans work together, they select options that minimize the effort of both the human individuals, i.e., maximize joint task efficiency [79, 80]. More precisely, the hypothesis is:

- *Hypothesis 3 (H3)* Robotic experts, due to their better understanding of the robot's behavior, will select positions that optimize the robot task, potentially jeopardizing their physical ergonomics for optimizing joint task efficiency

### Workplace Design

As previously highlighted, some limitations on the workplace design were present in the previous study. Therefore, a better understanding of the workplace according to the task description in Sec. 2.3 is necessary. To achieve this goal, an analysis in a simulation engine is performed. Siemens Tecnomatix™ Process Simulate Human Modeling and Simulation is used. Therefore, the robotic cell's Computer Aided Design (CAD)

**Table 2.3:** Qualitative representation of the four selected configurations for understanding how robot and human effort are allocated. The four configurations have been selected to have opposite efforts. The level of effort is displayed using ↗ (increasing) and ↘ (decreasing) effort. Erg. Risk stands for the physical ergonomics risk in the configuration.

| Effort / Conf. | Human | | | Robot | | |
|---|---|---|---|---|---|---|
| | Path | Erg. Risk | Time | Path | Extension | Time |
| RL | ↗ | ↗↗ | ↗ | ↘ | ↘ | ↘ |
| RH | ↗ | ↗ | ↗ | ↘↘ | ↘↘ | ↘↘ |
| HL | ↘ | ↘ | ↘ | ↗↗ | ↗↗ | ↗↗ |
| HH | ↘ | ↘↘ | ↘ | ↗ | ↗ | ↗ |

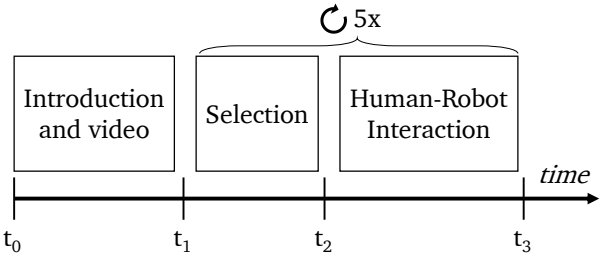



**Figure 2.8:** Experiment design schema. At first, the users were introduced to the task, and a video of the human-robot collaborative manipulation was shown. This initial phase ended at $t_1$, where informed consent and general demographics regarding robotics background were collected. Then, five equal repetitions of the task unfolded. The individual could choose a different configuration between $t_1$ and $t_2$ in each replica. However, once the best one was identified, the individual had to repeat the same interaction until reaching the fifth one. Thus, users could test all the levels first and then decide on the best one. At the end, $t_3$, the users were asked which configurations they preferred the most.

model is created respecting real-world dimensions. Afterward, two human models with anthropomorphic data accommodating 95% of the German female and male population were introduced to ensure that different individuals are considered as suggested by [81, 82]. Finally, four parts configurations with different levels of robot and human effort as shown from Tab. 2.3 were selected, and robotic programs were created. The final configuration of the workplace with the four configurations is shown in Fig. 2.7. With this simulation, it is identified that the previous study did not allow the individuals to optimize their physical ergonomics as long the parts displaced on the table surface were too low, thus leading to considerable elbow bending, which leads to high RULA values. Therefore, in this second study, configurations Robot-High (RH) and Human-High (HH) have been selected to have a z displacement from the table of around 0.30 m; in this way, the elbow angle is reduced, and the physical ergonomics is improved. Additionally, configurations Human-Low (HL) and Robot-Low (RL) were added to account for the robot's vicinity. By creating an experiment study with this configuration, individuals participating could choose to optimize for the robot or themselves with different magnitudes, thus allowing to gather data regarding *H3*.

### Experiment design

Having four different configurations, this second experiment was designed to understand the decisions of the individuals. The experiment schema is shown in Fig. 2.8. At first, between time $t_0$ and $t_1$, the users were introduced to the experiment, and a video of the interaction was shown. To mitigate their bias toward one configuration or another, the users were given the goal of deciding on a configuration that was the most efficient for the human-robot team; the implementation of the configurations is shown

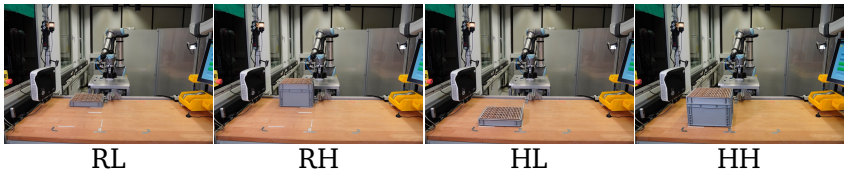



**Figure 2.9:** Implementation of the four configurations in the robotic cell. A box with markers for fixed part positions ensures that positions are constant between users. The box or boxes had to be moved to the respective position marked on the table to allow the user to select between different configurations.

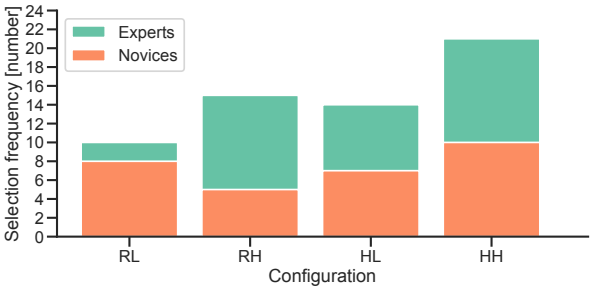



**Figure 2.10:** Outcomes of the decisions in experts and novices. The decisions are reported by displaying the number of users, divided upon expertise, that selected a specific configuration across the five trials.

in Fig. 2.9; therefore, leaving freedom to interpret the meaning of the word efficient. Afterward, at time $t_1$, informed consent and general demographics regarding robotics usage were collected. Having this information, two groups could be identified: the robotic experts and the novices. More precisely, considering the results of the previous study, robotic experts and novices were divided against a hard constraint: if they ever programmed one robotic manipulator. Next, between $t_1$ and $t_3$, the users had to perform five repetitions of the task, and at the beginning of each repetition, between $t_1$ and $t_2$, they could select the preferred configuration. However, once they identified the one satisfying the efficiency requirement, they had to repeat the interaction until they reached the fifth repetition. Finally, at time $t_3$, they were asked to select which configuration they preferred through a multiple choice questionnaire. During the experiment configurations, physical ergonomics, decision time, robot path, and human distances to the robot were collected.

### Results

In the user study, 16 users were invited, but four did not perform the test correctly; therefore, the sample was shrunk to 12. The analysis in this section thus concerns a user group of 12 male individuals age M = 26.08 yrs, SD = 2.24, height M = 183.00 cm, SD = 6.55. Half of them had medium to high experience with robotics; the other half had none.

The first outcome of the experiment regarded how the decision varied across repetitions. This is shown in Fig. 2.10. As the figure shows, the number of users selecting a specific configuration changed across the five trials. Interestingly, most users decided on configurations favoring good physical ergonomics (configurations HH or HL) in the first repetition. However, across the succeeding trials, this distribution changed. At the last trial, half of the user group decided on non-optimal configurations considering their physical ergonomics (configurations RH or RL). Therefore, starting from the initial distribution primarily skewed to the configurations close to the human (*RL*: 0, *RH*: 1, *HL*: 5, *HH*: 6) the users concluded with a more even distribution of selections (*RL*: 1, *RH*: 3, *HL*: 2, *HH*: 6). Yet, despite the redistribution, half of the user population preferred level HH, the best in terms of physical ergonomics. A correlation analysis with the Pearson coefficient between the different factors collected during the experiment is conducted to better understand these changes. Results for the metrics at the first trial are shown in Tab. 2.4. At the same time, results for the last trial are shown in Tab. 2.5. From the analysis, it is possible to understand that the robot experience influences just the first trial, meaning that the robot experts preferred HH and novices preferred more HL considering the variations in decisions in Fig. 2.10. However, this correlation disappears at the last trial.

Aside from the selected configuration, the physical ergonomics was monitored using the tool described in Sec. 2.3. The RULA risk level across the different configurations in all trials is checked to verify the correctness of the user study and the physical ergonomics level. The results are shown in Fig. 2.11. From the results, it is possible to see that the trend in the RULA respects what the simulator calculated. Therefore, if configurations are seen, the lower RULA levels are more frequent in HL and HH. Thus

| | M | SD | Exp. | Time | Config. |
|---|---|---|---|---|---|
| Exp. | 0.34 | 0.38 | - | | |
| Time | 22 | 10.52 | -0.57 | - | |
| Config. | 0.80 | 0.22 | 0.64* | -0.73** | - |

**Table 2.4:** Correlation analysis of the metrics related to the user group at the first trial. Exp. stands for robotic expertise and Config. stands for configuration. M of robotic expertise and configuration are adimensional and normalized between 0 and 1, whereas M of Time is expressed in seconds. The higher the configuration, the better the physical ergonomics. The statistical significance of the results is displayed with asterisks. In the case of the first trial, correlations are identified between configuration time and robotic expertise.

| | M | SD | Exp. | Time | Config. |
|---|---|---|---|---|---|
| Exp. | 0.34 | 0.38 | - | | |
| Time | 8.51 | 2.20 | 0.02 | - | |
| Config. | 0.69 | 0.36 | 0.03 | 0.45 | - |

**Table 2.5:** Correlation analysis of the metrics related to the user group at the last trial. Exp. stands for robotic expertise and Config. stands for configuration. M of robotic expertise and configuration are adimensional and normalized between 0 and 1, whereas M of Time is expressed in seconds. The higher the configuration, the better the physical ergonomics. The statistical significance of the results is displayed with asterisks. In the case of the last trial, the meaningful correlations disappear.

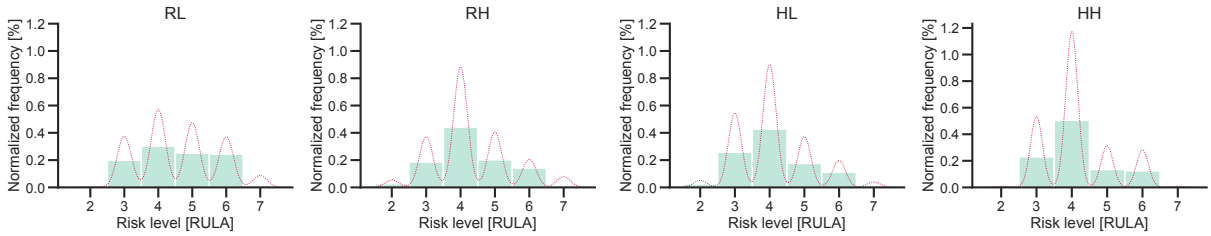



**Figure 2.11:** Results of the average RULA risk level across the four configurations. Results are displayed with normalized frequency of risk level. For clarity, the Kernel Density Estimation (KDE), in red, is displayed to give a better insight into the differences across the configurations. As the graph shows, the resulting risk levels reflect the results obtained in the simulator. The higher the RULA risk level, the worse.

providing insight into the experiment's being well designed. Additionally, to better a correlation analysis with the Pearson coefficient was performed as shown in Tab. 2.6. This analysis also yields the same result that the experiment was well designed as long as a strong correlation between the RULA and the configuration is found.

Thus, knowing that the design was well implemented due to the previous finding, the analysis of the results can continue. Regarding the physical ergonomics, the average RULA levels across the five trials were monitored as shown in Fig. 2.12. Upon analyzing the data, it becomes apparent that experts began with a lower average Rapid Upper Limb Assessment (RULA) score. However, experts were found to increase their susceptibility to physical ergonomic hazards throughout the trials. Conversely, novice operators of robotic systems initially exhibited a higher RULA score but exposed themselves more to physical ergonomic hazards.

Conversely, the novice robotic operators displayed a high RULA level during the first trials. However, their trend decreased over time, culminating in a significantly lower RULA level than the experts. More precisely, during the first trial, the measured RULA was M = 4.08, SD = 0.45, and M = 4.18, SD = 0.35 for experts and novices, respectively. However, during the last trial, the measured RULA was M = 4.13, SD = 0.35, and M = 4.03, SD = 0.49 for experts and novices, respectively. When comparing the RULA score of the last trial between experts and novices, the t-test shows a statistical difference. Moreover, when comparing novices' first and last trials, the t-test also shows statistical differences. However, this did not hold for experts.

The distance between the robot workplace and the operator's trunk was measured as mentioned. The results of these measurements across the experiments are shown in Fig. 2.13. From these results, it seems that experts tended to be closer to the robot when compared to novices. Additionally, experts tended to move closer to the robot across the trials, whereas novices kept a constant distance. More precisely, during the first trial, the measured distance was M = 1.59 m, SD = 0.80 and M = 2.42 m,

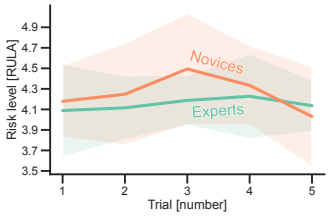



**Figure 2.12:** Measured RULA levels across the five experiments. Robotic experts show a trend of increased RULA levels, thus increasing exposure to ergonomic hazards. On the contrary, the novices tended to decrease their RULA level.

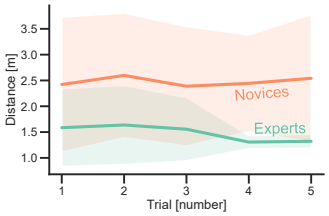



**Figure 2.13:** Measured distance between the operator trunk and the robot base. Experts positioned closer to the robot and moved even closer over time, while novices tended to move farther away. Distances were measured in meters from the camera.

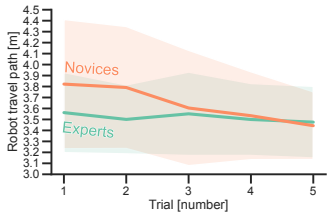



**Figure 2.14:** Measured robot travel path across the five repetitions. During the repetitions, robotic experts and novices tended to reduce the robot travel path. However, initially, experts tended to select a shorter robot travel path.

**Table 2.6:** Correlation analysis of the metrics across the five trials calculated with the Pearson correlation coefficient. Significant correlations are marked with asterisks. M of Robot. exp and configuration are adimensional and normalized between 0 and 1; time is expressed in seconds, and height is in centimeters. Strong correlations are found between the RULA level and the selected configuration.

| | M | SD | RULA | Height | Time | Configuration | Robot exp. |
|---|---|---|---|---|---|---|---|
| RULA | 4.15 | 0.39 | - | | | | |
| Height | 183.00 | 6.55 | -0.33* | - | | | |
| Time | 13.15 | 7.17 | 0.14 | -0.2 | - | | |
| Configuration | 0.59 | 0.37 | -0.5*** | 0.19 | 0.02 | - | |
| Robot exp. | 0.34 | 0.38 | -0.04 | 0.2 | -0.3* | 0.1 | - |

SD = 1.36 for experts and novices, respectively. However, during the last trial, the measured distance was M = 1.32 m, SD = 0.40 and M = 2.54 m, SD = 1.36 for experts and novices, respectively. When comparing the average distance to the robot in experts and novices in the last trial, the t-test shows significance. Moreover, the t-test performed between the first and last trials in experts showed significance. Considering these results, it is possible to deduce that experts effectively reduced their distance to the manipulator across the trials, whereas novices did not.

The third set of measurements regarded the resulting robot travel path given by the different selected configurations. The resulting travel paths divided upon expertise are shown in Fig. 2.14. In contrast to the observed trends in physical ergonomics and distances, the travel paths taken by experts and novices exhibited similar trends. Specifically, both groups displayed reduced robot travel paths across the repetitions. However, it is noteworthy that experts initially opted for a shorter travel path when compared to novices. More precisely, during the first trial, the measured travel path was M = 3.56 m, SD = 0.37 and M = 3.82 m, SD = 0.59 for experts and novices, respectively. However, during the last trial, the measured path was M = 3.47 m, SD = 0.33 and M = 3.44

m, SD = 0.31 for experts and novices, respectively. The t-test performed on the last trial between experts and novices did not report statistical differences. Therefore, it is possible to conclude that experts and novices selected similar robot travel paths.

The fourth set of measurements concerned the decision time required by participants to select a configuration before commencing the collaborative task or referring to Fig. 2.8 the time between $t_2$ and $t_1$. The outcomes are shown in Fig. 2.15. In a manner analogous to the observed trend in robot travel paths, both groups tended to decrease the time required for configuration selection. Nevertheless, the expert group displayed a faster initial decision-making speed, as shown by the significant correlation in Tab. 2.6. In detail, the measured decision time was M = 16.30 s, SD = 3.55 and M = 27.70 s, SD = 12.40 for experts and novices, respectively, at the first trial. However, in the last trial, the time decreased to M = 8.57 s, SD = 2.33 and M = 8.47 s, SD = 2.28 for experts and novices, respectively. The t-test confirmed that both groups had a significant difference in time between the last trial and the first one. However, no statistical difference was found between novices and experts in the last trial. Therefore, it is possible to conclude that experts and novices showed a similar decision time after the five repetitions. Across the repetitions, they reduced the time necessary for making a decision.

The last analysis concerned the execution time needed for the task, or referring to Fig. 2.8 the time between $t_3$ and $t_2$. The outcomes are shown in Fig. 2.16. The trend in this last set of measurements is similar to the one in the decision time, where experts and novices tended to reduce the time with repetitions. More precisely, the experts' execution time during the first trial was M = 58.92 s, SD = 9.50, whereas in the last trial, it was M = 44.72 s, SD = 3.83. In novices, the execution time was M = 76.03 s, SD = 15.86 and M = 52.32 s, SD = 17.12 for the first and last trials. A t-test confirmed that both groups significantly reduced the execution time between the first and the last trial. Additionally, the t-test confirmed no statistical difference between experts and novices at the last trial despite statistical differences in the first trial. Therefore, it is possible to conclude that experts and novices reduced the execution time across the trials at the end, reaching the same average execution time.

#### Discussions

This follow-up study investigated the relationship between task autonomy, physical ergonomics, and operator background after the results obtained in the first study. To design this study, the limitations identified in the previous one were addressed with three main differences. Firstly, the experiment required operators to choose between con-

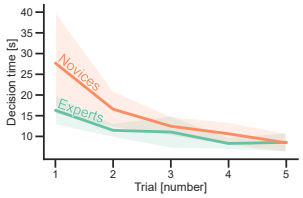



**Figure 2.15:** Variations of decision time ($t_2 - t_1$) for the different trials. At the end of the experiments, both groups opted for shorter decision times. However, at first, experts were faster in the process.

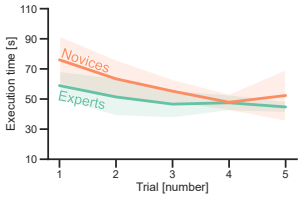



**Figure 2.16:** Execution time variations ($t_3 - t_2$) for the different trials. Experts show a constant execution time. However, novices did not. Interestingly, novices at the last trial slightly increased the execution time.

**Table 2.7:** Qualitative summary of the experiments' results for experts with differences between the first and last trials. The level of effort is displayed using ↗ (increasing) and ↘ (decreasing) effort. If the level of effort between trials maintains the same arrow, it is kept constant. Erg. Risk stands for the physical ergonomics risk in the configuration, conf. for the most preferred configuration, Exec. time for the execution time, dec. time for decision time, and H-R stands for human to robot.

| Effort / Trial | Expert | | | | | |
|---|---|---|---|---|---|---|
| | Conf. | Erg. risk | Exec. time | Dec. time | H-R distance | Robot path |
| 1st Trial | HH | ↘ | ↗ | ↗ | ↘ | ↘ |
| 5th trial | n.a | ↗ | ↘ | ↘ | ↘ | ↘↘ |

**Table 2.8:** Qualitative summary of the experiments' results for novices with differences between the first and last trials. The level of effort is displayed using ↗ (increasing) and ↘ (decreasing) effort. If the level of effort between trials maintains the same arrow, it is kept constant. Erg. Risk stands for the physical ergonomics risk in the configuration, conf. for the most preferred configuration, Exec. time for the execution time, dec. time for decision time, and H-R stands for human to robot.

| Effort / Trial | Novice | | | | | |
|---|---|---|---|---|---|---|
| | Conf. | Erg. risk | Exec. time | Dec. time | H-R distance | Robot path |
| 1st Trial | HL | ↗ | ↗ | ↗ | ↗ | ↗ |
| 5th trial | n.a | ↘ | ↘ | ↘ | ↗ | ↘ |

straining options, prioritizing either the robot or the operator's well-being. Secondly, the user group was divided into robotic experts and novices. Finally, the operators were exposed repetitively to the collaboration to account for different exposure rates. During the experiment, the type of configuration selected by the operator, decision time, execution time, average distance between the robot workplace and operator, and robot travel path were measured. Through these measurements, a better understanding of the complex decision-making process of operators was thus possible. For clarity, all the results divided upon robotics expertise are summarized in Tab. 2.7 and Tab. 2.8.

This second study identified that expertise and task knowledge significantly impact the ability to choose the best ergonomics. Experts identified the best ergonomic configuration at the first encounter of the task. However, over time, they tended to

incorporate the robot partner in optimizing the task, resulting in a slight tendency to deteriorate their physical ergonomics. On the other hand, novices developed a better task model over time and improved their physical ergonomics. Therefore, the development of task knowledge should be encouraged as it allows operators to optimize their interaction with the robot.

The distance between the operator and the robot was also affected by expertise and experience with the task. Novices kept a higher distance overall than experts, but over time, experts reduced their distance from the robot, potentially indicating that their risk perception was reduced by experience. This allowed experts to incorporate the robot into their action space, facilitating the development of a better model of the robot's behavior. It is still being determined whether, with more repetitions, novices would also show a similar effect.

Finally, experts and novices learned to reduce the robot's travel path, decision time, and execution time over repetitions. This indicates that experience with the specific task is needed to develop a more optimal task model. In summary, these findings suggest that experience and exposure are critical factors for operators to develop a model of the joint task and the robot behavior. Allowing operators to develop a task interaction history with the robot partner would allow them to optimize both individual and joint task design.

Considering these results, it is possible to conclude that the initial hypothesis *H3* is confirmed, suggesting that robot experts may optimize joint task efficiency better than novices as long they tend to reduce both the robot travel path and their distance to the robot system. Furthermore, considering recent literature [83], it is possible that, with longer habituation, robotic novices may understand how to optimize for robot travel path as their perception of the safety of the robot's behavior increases. However, further research is needed to better understand why robotic experts did not improve their physical ergonomics with exposure to the task, while novices did.

## 2.5 Conclusions

### Summary

Through this chapter, this work wanted to discover the user impacts of intuitive programming interfaces. More specifically, this section tackled *Challenge 3.2* (intuitive programming interfaces that do not require robotics expertise) by proposing a novel intuitive programming method based on co-assembly locations and studying its influences on the users.

To study such influence, the chapter presents two studies to analyze the impact of allowing operators to define co-assembly locations in physical human-robot collaboration scenarios. Initially, the chapter started with a literature overview concerning the benefits of allowing operators to configure tasks, discovering that physical ergonomics could be improved if users have task configurability. Next, the chapter presented how such interaction could be integrated using CV techniques to allow the definition of co-assembly locations and measure the level of physical ergonomics. Last, the chapter presented the two studies that tried to identify the decisions' impact on the task and the operators' well-being considering the physical ergonomics and other task parameters like execution time or travel paths.

From the studies, it was possible to measure that allowing operators to decide on the co-assembly locations can benefit the task decision autonomy as long the measured autonomy score with WDQ was on average 86.25% while the score for pre-defined locations was on average 68.75%. Additionally, it was discovered that operators selected co-assembly locations in different ways upon previous exposure to robotic programming. More precisely, robotic experts decided on locations that better optimized the efficiency of the joint task between robot and human by reducing the robot's travel path while jeopardizing the operator's physical ergonomics as long the average physical ergonomics in the last trial measured with RULA was 4.13, the average distance human to the robot was 1.32 m, and the average robot travel path was 3.47 m. In contrast, novices selected displacements that mainly benefited the task efficiency of the single individual by improving the operator's physical ergonomics as long the average physical ergonomics in the last trial measured with RULA was 4.03, the average distance from human to robot was 2.54 m, and the average robot travel path was 3.44 m.

#### Discussions and future work

Considering these findings, it is possible to underline that allowing operators to select co-assembly locations can also benefit task decision autonomy in industrial tasks and thus be an excellent user-friendly programming method initially identified by [46] but in non-industrial tasks. However, such a programming method can have extensive user impacts. The studies identified that background robotic expertise plays a role in how users define the co-assembly locations and how the human-robot interaction unfolds. The study hinted that experts might treat the robot as a teammate while novices did not, considering recent literature [79, 80]. Nevertheless, the intuitive programming method allowed for high decision autonomy for different robotic expertise; therefore, it can be an excellent tool to address the above-mentioned challenge and help SMEs

program robots without needing robotics expertise. Therefore, practitioners who want to integrate similar programming methods should ensure that previous robotics expertise is adequately addressed by either constraining choices or providing guidance to ensure that users do not select co-assembly locations that might lead to injuries or inefficiencies. Unfortunately, it is essential to underline that the studies presented in this chapter mainly targeted collaborative pick-and-place applications; therefore, the applicability of these results to other, more complex tasks requires further investigations. Future works should investigate if these results apply to other tasks.

Chapter 3

# Influences of User-Defined Labeling Points Through a Spatial Device on Robotic Grasp Points

Contents



When robots are dealing with tasks, they often fall within two scenarios. Either the robot moves in the space to reach a target location or directly interacts with the environment to perform some action through its EE. The previous chapter presented studies on the influences given by defining end-locations from the operators. This chapter studies how operators can define approaches of the robot EE with target objects. The contents of this chapter are partly based on the following publications:

- Reisch, R. T., Pantano, M., Janisch, L., Knoll, A., Lee, D., "Spatial Annotation of Time Series for Data Driven Quality Assurance in Additive Manufacturing," 2022, 17th CIRP Conference on Intelligent Computation in Manufacturing Engineering (ICME).

- Caporali, A., Pantano, M., Janisch, L., Regulin, D., Palli, G., and Lee, D., "A Weakly Supervised Semi-automatic Image Labeling Approach for Deformable Linear Objects," 2023, IEEE Robotics and Automation Letters (RA-L), Volume 8.
- Pantano, M., Klass V., Yang Q., Sathuluri A., Regulin D., Janisch L., Zimmermann M., and Lee, D., "Simplifying Robot Grasping in Manufacturing with a Teaching Approach based on a Novel User Grasp Metric," 2023, 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM).

The contents of this chapter are structured as follows. Initially, the motivation for this work is outlined. Second, an overview of the state of the art concerning robotic approaches with target objects is given, and a need for simpler interfaces to define grasping locations is discovered. Next, considering the discovered need, three preliminary studies to understand how interactions of the EE with the environment can be defined with spatial interactions are presented. Fourth, a novel method for defining the grasp position using a novel spatial interaction is described. Finally, the results of user experiments underlying the effectiveness of the proposed method are outlined together with remarks on its applicability in industrial use cases.

## 3.1 Motivation

Robots can be used at their best when they can perform direct interactions with the environment, meaning that the EE is used to perform operations on objects. This translates mainly into object grasping and un-grasping. However, these actions are not trivial and can hardly be solved without using external information [38]. Therefore, several research works have proposed methods to fuse sensor data with objects' properties to extract robust locations for grasping. The most exciting results in this topic have been presented at the Amazon Picking Challenge as extensively summarized in [84]. In this contest, robotic systems had to detect unknown objects and select the best grasping position to ensure a robust pick-up. The most successful teams proved that combining cameras, hybrid EE, and Convolutional Neural Networks (CNN) makes solving the task for several objects possible. Unfortunately, the proposed systems were optimized to ensure the best grasp success rate but not to satisfy task needs (e.g., grasp in a defined location) [38, 85]. Therefore, there is a need to research how task-specific knowledge can be integrated into those systems. More precisely, as long as operators know any task requirements, the question remains on defining constraints for the robot to pick parts.

## 3.2 Related Works

### Analytical Robotic Grasping

In the initial stages of robotics, grasp poses computations primarily relied on analytical solutions. This approach involved considering mathematical models of the object's dynamics and kinematics, focusing on estimating grasp poses by considering the object's equilibrium and stability, as discussed in [38]. While a comprehensive domain analysis is deemed unnecessary here, it is pertinent to note that analytical methods exhibit high complexity, a limited solution space, and often exclusively address grasp stability, neglecting location considerations and practical boundaries [38, 86]. However, the landscape has witnessed a paradigm shift over the past decade with the emergence of Deep Learning (DL). Data-driven approaches facilitated by DL have demonstrated outcomes surpassing those achieved by analytical methods.

### Data-driven Robotic Grasping

Data-driven, or learning, approaches exploit data from visual sensors mounted on the robot cell. Such data is then used to identify relevant regions on the target object to estimate grasp poses, hence not involving an explicit mathematical model that defines stability and equilibrium during inference. However, to identify the relevant grasp regions, labeled data for training is necessary. However, two approaches are used in learning-based methods. On the one hand, model-based approaches have a priori object knowledge (e.g., 3D model) during training like in [87, 88]. On the other hand, model-free methods have no prior object information like in [89, 90]. An in-depth review of these two methods can be found in this section in [38, 84, 91]; therefore, analysis of their performances will be redundant. Nonetheless, citing prominent technologies and approaches in model-based and free methods is required to understand their capabilities and inefficiencies. In model-based approaches, object information is used to generate training data, which can then be fed to a CNN to compute grasp reliability. A noteworthy example in this class is Dex-Net [88]. This algorithm samples possible grasp candidates on image data provided by depth cameras and then assesses their performances, i.e., quality, via a Grasp Quality Convolution Neural Network (GQ-CNN). Employing this approach, the latest version of this algorithm, Dex-Net 4.0, achieves a grasp success rate up to 95% with unknown objects [92]; thus, it is assessed as state-of-the-art model-based DL grasp computation network [38]. In model-free methods, training is driven by a lack of object knowledge due to application requirements; thus, grasp sampling is often skipped, as shown in [93, 94]. Therefore,

model-free approaches build on Fully Convolution Networks (FCN) rather than CNNs. A prominent example in this class is GG-CNN [89], which, despite the name, uses fully convolution networks to generate pixel-wise grasp poses. Therefore, GQ-CNN achieves faster grasp computation due to fewer network parameters when compared with model-based approaches and shows an 83% grasp success rate [89]. Despite the remarkable results achieved with these two methods, they still need solutions to select grasp poses that consider task constraints and not only object properties. In other words, the presented methods base the grasp quality computation only on the robotic EE and the target object, neglecting any other constraint that might appear after the grasp is executed. For example, if a hammer needs to be grasped and handed to a human, the network will compute the best grasp pose, generalizing on previous data; therefore, the solution of taking the hammer on the head might be disregarded. Thus, the hammer will be handed to the human with the head facing the human, which may cause higher hazards than handing the hammer with the handle facing the human. Additionally, when the robot has to take the hammer through the handle, it might fall off from the EE, as its center of mass needs to be accounted for correctly. An example of this problem can be seen in [85]. This problem has also been highlighted in one of the challenges for industrial applications of data-driven grasping by [38]. To the best of the author's knowledge, the only works that try to bridge this gap are proposed in [95, 96]. Nonetheless, these approaches require embedding a knowledge representation for the object, which becomes too cumbersome if the products to be picked change often. Therefore, a gap is still present in the solution to this problem. More precisely, no research has proposed how knowledge of the operator can be leveraged directly at the robotic cell to solve this issue. There is a need for more investigation on this topic.

#### Influencing Grasp Poses via Labeling

Labeling must be performed to influence the selection of grasping poses. Marking candidate grasps is often known to happen when creating datasets for training data-driven robotic grasping algorithms, and robotic experts have mainly performed this task. Therefore, visual data is obtained from a photo-realistic simulation or real-world data. Then, annotations in the form of pixel-wise information or oriented bounding boxes are inserted via point-clicks. Examples of datasets created with these processes are the Cornell Grasp Dataset [97] or the Dex-Net one [88]. However, this method might not be optimal if an operator would have to do it as it bears the disadvantages of any other point-click labeling approach [98]. Additionally, the operator might need more robotic experience to define a robust grasping position. Therefore, the research

community has proposed novel methods for simplifying the labeling. One example has been presented in [96] where the authors propose using LfD to generate objects' task constraints. However, no demonstration was given on its effectiveness with different users. Another method has been proposed in [99], where they try to infer the correct grasping pose by physically simulating the handled object instead of using operator demonstrations. However, the improvement in the success rate was minimal. In [100], the authors propose using a digital pen to mark the EE pose in relation to the target object via technical drawings. However, their approach was limited to welding applications. Therefore, to the author's knowledge, the only work that tried to integrate grasp labeling via natural interactions and considered the impact on test subjects has been proposed in [101]. In this work, the authors suggest that operators can gather information about grasp labeling if they point directly at the object that has to be taken. Unfortunately, their approach is limited to the target object rather than the exact grasping location. Therefore, there is a need to investigate how to integrate grasp pose labeling while considering both operator inaccuracies and natural interaction methodologies.

## 3.3 Preliminaries

As seen from the state-of-the-art, no tangible work suggest integrating grasp pose labeling while considering operator inaccuracies and natural interaction methodologies. However, a promising approach for gathering labeling information can be through spatial sensors, as suggested by [102]. To investigate the potential of using a spatial sensor, two preliminary studies have been performed in advance to assess the capabilities of such sensors. The Tracepen™ spatial sensor was used for these studies. The Tracepen™ is a spatial pen-shaped sensor based on the principle of reflective photodiode receivers. Thus, the travel time from the signal emission to its reception is measured to calculate the sensor position. However, the sensor can bear some inaccuracies. Therefore, this section describes the recorded outcomes when giving the sensor to an operator for labeling time series and image data. Additionally, to better understand how the spatial sensor can be used for industrial grasp labeling, a preliminary study comparing different industrial bin-picking systems is also reported in this section.

### 3.3.1 Usage of a Spatial Sensor for General Purpose Point Labeling

A first use case concerning anomaly detection has been used to investigate the possibility of labeling features via a spatial sensor. Parts produced by the additive manufactur-

ing process WAAM can present defects given by printing parameters or environmental factors. Anomaly detection algorithms can reduce the number of such defects, as illustrated in [103]. However, such algorithms require labeled anomaly data. In this preliminary study, using a spatial sensor, as shown in Fig. 3.1 was envisioned to generate such data. The choice of using a spatial sensor was driven by the need to reduce the time to label anomaly data, which nowadays is very time-consuming as long the data to be labeled is a time series.

#### Method

To ensure that the spatial sensor can produce labeled information that can be correlated to the part. Transformations of labeled points in the part coordinate system are necessary as spatial sensors have their reference frame. This problem is shown in Fig. 3.2. Therefore, the transformation problem can be reformulated using the figure nomenclature. First, the transformation between the reference frame of the spatial sensor and the origin of the part ${}^{t}T_{p}$ had to be computed. For this calculation, a method similar to [104] was employed. Therefore, the spatial sensor was used to sample three points on the part coordinate frame: the origin $P_1$, one on the x-axis $P_2$, and one on the x-y plane $P_3$. Then, Eq. 3.1 was solved.

$$ {}^{t}T_{p} = {}^{p}P_{i}\,{}^{t}P_{i}^{-1}, \forall i (i = 1,2,3) \tag{3.1} $$

In this equation, ${}^{p}P_{i}$ denotes points in the part coordinate system, while ${}^{t}P_{i}$ represents points in the spatial sensor coordinate system. Subsequently, with the obtained ${}^{t}T_{p}$, the positions of anomalies can be calculated using Eq. 3.2.

$$ A_{i}^{p} = {}^{t}T_{p}\,{}^{t}A_{i}^{-1}, \forall i (i = 1,2,3) \tag{3.2} $$

In this formula, $A_{i}^{p}$ represents the anomalies in the part coordinate system, and ${}^{t}A_{i}$ represents the anomalies in the spatial sensor coordinate system. Subsequently, these identified anomalies, recorded as points in the part coordinate system, are utilized for marking specific regions in the time series data of the part through a proximity search.

#### Performances and User Effects

Two tests were performed to investigate some of the first-level effects of using a spatial sensor. The first experiment evaluates the accuracy of the spatial sensor, and the second one evaluates the accuracy of labeled points with the device. For the first evaluation, a test bench with mechanical mechanisms to induce controlled translations was used

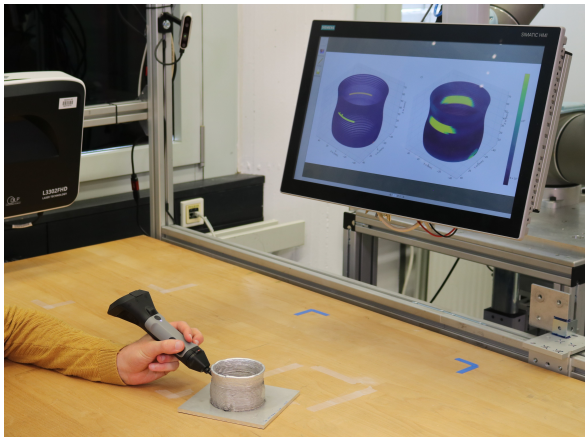


**Figure 3.1:** Example of part labeling using the Tracepen™ spatial sensor. The human operator stands in front of the part and marks anomalies. Any marked anomaly point is then displayed back to the user with a display.

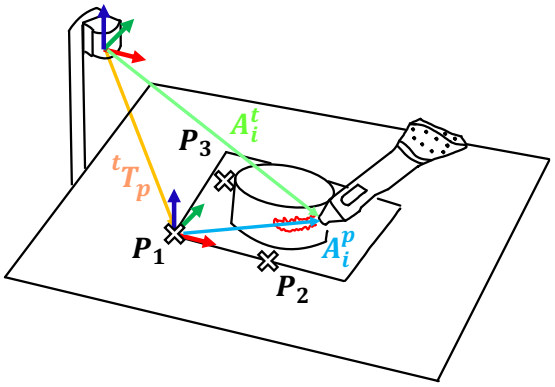



**Figure 3.2:** Transformations involved in labeling anomalies. ${}^tT_p$ represents the transformation from the sensor coordinate system to the part one calculated with $P_i$. ${}^tA_i$ are the anomalies in the sensor coordinate system, and ${}^pA_i$ are the anomalies in the part system.

as long as relative movements from the part origin have been used in the described method. Therefore, small translations were applied, and sample points were collected before and after the transformation to calculate the delta change with Eq. 3.3.

$$\delta = P_i^s - P_i^e, \forall i (i = 1,2,3) \tag{3.3}$$

Where $P_i^s$ denotes the measured position by the spatial sensor before the transformation, and $P_i^e$ denotes the measured position by the spatial sensor after the transformation. This test was repeated three times for each main axis, and the results were as follows. For the x-translation $\delta = 0 \pm 1$mm. For the y-translation $\delta = 0 \pm 1$mm. For the z-translation $\delta = 0 \pm 1$mm.

For the second test evaluation, an exploratory user study was conducted. In the user study, users had to label points with two methods. On the one hand, with the method proposed here. On the other hand, with a 3D computed tomography scan. In this test, the users had 10 min for labeling as many anomalies as possible, and at the end, labeling performances in terms of recall ($tpr$), precision ($ppv$) and balanced accuracy ($acc$) were calculated. For this study, two users with age M = 28.70 yrs, SD = 2.23 were invited to perform the labeling, and the results yielded as follows. For the method with the spatial sensor, $tpr = 30.5 \pm 27.6\%$, $ppv = 11.5 \pm 4.9\%$, and $acc = 60.0 \pm 11.3\%$. For the method with the computed tomography scan, $tpr = 42.0 \pm 1.4\%$, $ppv = 43.5 \pm 0.7\%$, and $acc = 68.5 \pm 2.1\%$.

Thus, it is possible to extract two outcomes from the testing. First, the spatial sensor is unreliable for precise measurements as long it has accuracy in relative movements

of ±1mm. Additionally, by allowing an operator to use the spatial sensor for marking features, the quality of the labeling points can be highly dependent on the user, as shown by the recorded high standard deviation values. Therefore, from this study, the following primary outcome is drawn. If this spatial sensor should be used for labeling, additional algorithms should be integrated to increase the precision of the marked features to ensure stable outcomes despite users' variances.

### 3.3.2 Refinement of Labeling Candidates

As seen from the previous section, accessory algorithms must be integrated to improve user accuracy if the spatial sensor is used for point labeling. A use case with small components has been selected to investigate a novel method for integrating labeling fine-tuning algorithms. More precisely, a use case concerning deformable linear objects (DLOs). DLOs are objects with an elongated cylindrical form that lack relevant features; electrical wires are the most prominent examples of DLOs. DLOs are present in several manufacturing domains, and robust DLO detection via CV can enable their manipulation with robotic solutions. One common approach for DLOs detection is via data-driven methods. However, their performances are heavily affected by the size and quality of the datasets. To solve this issue, datasets with a mix of real-world and synthetic data can be used. However, there is a need to investigate how real-world data can be efficiently generated and mixed with synthetic one without burdening the operator [105]. Therefore, this section proposes a novel approach for generating real-world data for DLOs with operators' error correction methods, and its effect on the operator is analyzed.

#### Background on Image Labeling

Several methodologies for labeling real-world data are available to reduce the operator's burden. The most prominent methods are weakly supervised labeling approaches. With these techniques, the person in charge of labeling provides a few pixel-wise points marking some features, and the underlying algorithms predict pixel-wise masks that mark the whole feature. A state-of-the-art example of this approach is proposed in [106]. However, despite the low effort in labeling one image, these systems can only quickly scale to multiple images if they require the user's input for each image. Therefore, to the author's knowledge, the only method to create large datasets with a low number of clicks is Chroma Key (CK). However, CK is limited to single-color backgrounds; therefore, it can be challenging to adopt in real-life scenarios (e.g., control

cabinets).

### Method

An approach based on the Tracepen™ spatial sensor is proposed to address the above-mentioned issues. This method aims to create several annotated images that a data-driven algorithm can use to learn DLO features. Therefore, the spatial sensor's labeling points in 3D coordinates must be transformed into pixel coordinates for the images. Therefore, coordinate transformations must be considered. The problem is shown in Fig. 3.4. Hence, the problem can be expressed as follows using the figure nomenclature. First, the transformation between the sensor coordinate system and the robot coordinate system (${}^{t}T_{r}$) is calculated by mounting the sensor on the robot flange and sampling three points which are then used to solve Eq. 3.4.

$$ {}^{t}T_{r} = (P_i^r + M)\,(P_i^t)^{-1},\ \forall i\ (i = 1,2,3) \tag{3.4} $$

In this equation, $P_i^r$ represents the sampled point coordinates in the robot coordinate frame, $M$ is the fixed transformation between the robot flange and the sensor mounting point, and $P_i^t$ denotes the sampled point coordinates in the sensor coordinate system. Subsequently, by utilizing ${}^{t}T_{r}$, the sampled points can be projected onto pixel coordinates using Eq. 3.5.

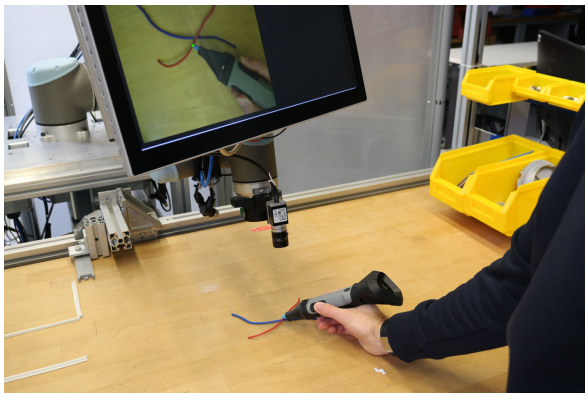


**Figure 3.3:** Example of DLO labeling using the Tracepen™. The user can mark points via the device and immediately receive feedback on the point location via the display, showing a live feed of the DLO from the robot's perspective.

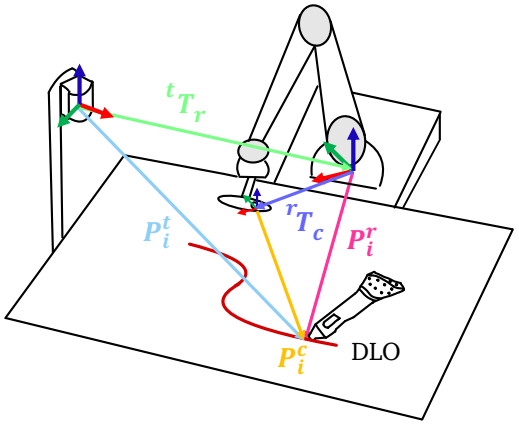



**Figure 3.4:** Illustration of the different coordinate frames involved in labeling a DLO with the spatial sensor. A point marked by the spatial sensor($P_i^t$) must be transformed to a point in the image coordinate frame ($P_i^c$) through ${}^{r}T_{c}$ and ${}^{t}T_{r}$.

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = K\, {}^{r}T_{c}\, {}^{t}T_{r}\, P_{i}^{t},\ \forall i\,(\, i \in C) \tag{3.5}$$

Where $u, v$ are pixel coordinates of a point $i$ which belongs to a DLO $C$ with $u = u'/w'$ and $v = v'/w'$, $w'$ is the camera scaling factor, $K$ is the intrinsic camera matrix, ${}^{r}T_{c}$ is the camera extrinsic matrix calculated as in [71] and $P_{i}^{t}$ is the sampled point via the spatial sensor belonging to the cable $C$. Next, an error correction method for the user input was integrated with the point in the pixel coordinates. Considering that the points are in the pixel coordinates, a method based on CNN error computation is employed. The approach consists of having the network calculate a pixel error between the input point $u, v$ and the center of the DLO by evaluating an image crop of $96 \times 96$ pixels around the input point $u, v$. Therefore, a ResNet-18 feature extractor [107] is used along two Fully Connected linear layers $FC_{512,256}$ and $FC_{256,96}$. In this way, the network outputs an array of dimension 96 representing the probability of an image column containing the DLO center. In this way, the offset between the input $u, v$ and the column with the highest probability is obtained with Eq. 3.6.

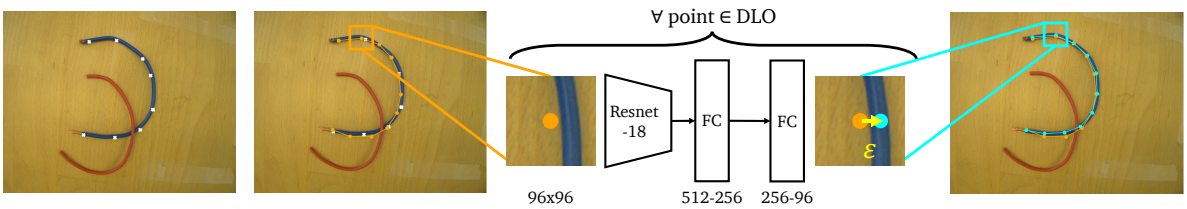
$$\varepsilon = v - v_{max} \tag{3.6}$$

Where $v$ is the column index for the labeled point and $v_{max}$ is the column index with the highest probability for the DLO center computed by the network. An example of how this pipeline works is shown in Fig. 3.5. A synthetic dataset generated with a photo-realistic simulator in Blender [108] is employed to train the network mentioned above. Thus, 40000 randomly extracted crops with simulated user error from 0 to 8 pixels were used, and the network was trained with Adam optimizer for 50 epochs with a batch size of 128 and learning rate of $5 \times 10^{-5}$.

#### Performances and User Effects

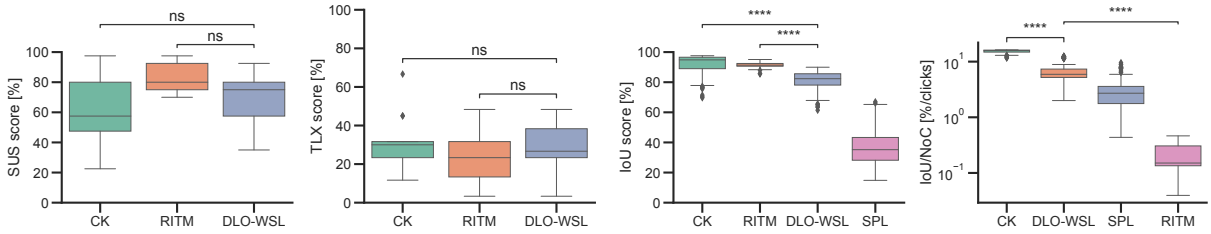
To evaluate and determine if fine-tuning user input via a CNN can be beneficial, a user test is envisioned to compare the proposed labeling approach (from now on denoted with *DLO-WSL*) with state-of-the-art ones. The user test comprises three randomized subsequent interactions with three different labeling tools. Therefore, *DLO-WSL* is compared against CK to evaluate its scalability with big datasets. Additionally, the method is compared against Reviving Iterative Training with Mask Guidance for Interactive Segmentation (*RITM*) [106] for comparison against state-of-the-art weakly supervised labeling known to reduce the number of clicks (NoC) for single image labeling tasks.

In the test, the users had to label 10 images; after the interaction, the usability was measured with the System Usability Scale (SUS) [109], and the workload with the NASA-TLX [110] scale was measured. Additionally, the NoC and the Intersection over Union (IoU) were recorded during the interaction. A total of 13 users not experienced with labeling age M = 32.70 yrs, SD = 9.23, participated in the study. All of them performed the task correctly, and no data sample was discarded. Concerning usability, the recorded SUS scores were M = 60.38%, SD = 21.00, M = 69.61%, SD = 16.26, and M = 82.30%, SD = 9.54 for *CK*, *DLO-WSL* and *RITM* respectively. Concerning workload, the recorded NASA-TLX scores were M=30.51%, SD=15.08, M=29.74%, SD=12.96, and M=22.31%, SD=13.12 for *CK*, *DLO-WSL* and *RITM* respectively. Concerning the performance, IoU and IoU/NoC ratios calculated using the three methods are used. A comparison with raw user input labels connected with a spline without CNN fine-tuning is added and denoted with SPL. For the former, recorded performances were M=91.68%, SD=6.56, M=91.32%, SD=1.52, M=81.05%, SD=6.22 and M=36.88%, SD=12.45 for *CK*, *RITM*, *DLO-WSL* and *SPL* respectively. For the latter, recorded performances were M=15.31%/clicks, SD=1.09, M=6.54%, SD=2.78, M=3.16%/clicks, SD=2.05 and M=0.21%/clicks, SD=0.13 for *CK*, *DLO-WSL*, *SPL* and *RITM* respectively. The results with statistical significance are shown in Fig. 3.6. Analyzing these results, it is possible to understand three primary outcomes. First, the usability of *DLO-WSL* is comparable with state-of-the-art tools. Second, *DLO-WSL* is an excellent tool to reduce the NoC for labeling a dataset while maintaining a high average IoU score. Third, fine-tuning with a CNN-based network can improve the quality of user labels. Therefore, it is possible to conclude that labeling with the Tracepen™ spatial sensor enabled via an NN-fine tuning algorithm is a valuable method that can provide good usability while minimizing the necessary number of clicks.




(a) User labels points (b) Spline is created (c) Fine-tuning of points (d) Spline with refined points

**Figure 3.5:** Example of the method for refining user inputs. At first, the user labels all the points regarding one DLO instance (a); second, a spline is created to obtain a line describing the DLO (b). Third, the spline direction is used to obtain a rotated crop around each label point with dimension $96 \times 96$, and a CNN evaluates the image for computing the offset $\varepsilon$ (c). Finally, each fine-tuned point is used to obtain a refined spline, which, with the DLO diameter, gives a DLO mask (d).




**(a)** Usability **(b)** Workload **(c)** IoU **(d)** IoU/NoC

**Figure 3.6:** Results of *DLO-WSL* evaluation with the user test. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed, or no statistical marking is shown. *CK* stands for Chroma Key, *RITM* stands for Reviving Iterative Training with Mask Guidance for Interactive Segmentation [106], and *SPL* stands for raw user input points without CNN correction. From the evaluation, it is possible to see how *DLO-WSL* can reduce the NoC while maintaining a good IoU.

### 3.3.3 Grasp labeling in Conventional Bin-picking Systems

Knowing that labeling with the Tracepen™ spatial sensor can help improve usability and reduce NoC, a last study is necessary to understand how grasp pose labeling is performed in modern industrial applications. Therefore, this section provides the outcomes of a preliminary study conducted to identify the pain points of these methods. In industrial applications, grasp candidates are defined in bin-picking 3D vision systems, as shown from [84]. Therefore, this study focused on their evaluation focusing on three market-ready systems.

#### Method

For evaluating the different bin-picking systems, considering the goal of identifying pain points related to grasp pose labeling, a between-subject design evaluating time and user experience was conceived. Therefore, three bin-picking systems were selected on the market. Three main criteria were used to select such systems: first, the bin-picking provider provided testing without system purchase; second, the underlying vision sensors differed between the systems. Third, the bin-picking kit's price range differed from the already selected systems. Thus, Roboception™ rc_viscore (in short *Rc*), Photoneo™ Phoxi M (in short *Pt*), and Mech-Mind™ Mech-Eye Pro S 1000M (in short *Mm*) were used for the evaluation. More precisely, a user study was envisioned in which users had to label one or more grasp points for a suction EE. The study involved 21 male subjects with age M = 27.3 yrs, SD = 3.40, and medium to high experience in robotic grasping divided evenly across the different systems. Each of them performed the task of grasp labeling with an eye-to-hand configuration, as shown in Fig. 3.7. At the interaction's end, usability, workload, and autonomy were measured with SUS,

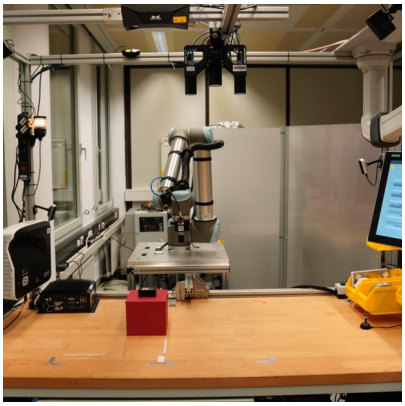
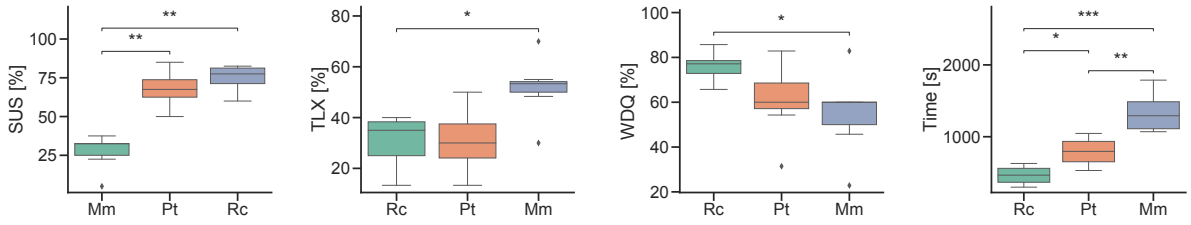


**Figure 3.7:** Example of set-up with conventional bin-picking systems. Cameras for the evaluation were mounted on the top of the work cell. Afterward, a part was placed under the camera, and the users were instructed to label a grasp point using the interface provided by the system. In this picture, the Mech-Mind™ Mech-Eye Pro S 1000M and the Roboception™ rc_viscore are shown on the top.

NASA-TLX, and WDQ. Additionally, the time for executing the task was monitored.

### Results

All 21 users performed the task correctly; the results are shown in Fig. 3.8. The results show that different outcomes have been recorded from the three systems. Concerning usability, the SUS scores were M = 27.41%, SD = 10.84, M = 67.85%, SD = 11.76, and M = 75.00%, SD = 8.16, for *Mm*, *Pt* and *Rc* respectively. Statistical difference is found between Mech-Mind™ and the other two systems. Concerning workload, the TLX scores were M = 51.66%, SD = 11.78, M = 30.92%, SD = 12.46, and M = 30.71%, SD = 10.62, for *Mm*, *Pt* and *Rc* respectively. Statistical difference is found only for Roboception™ and Mech-Mind™. Third, concerning autonomy the WDQ scores were M = 55.10%, SD = 18.12, M = 60.81%, SD = 16.22, and M = 75.91%, SD = 6.35, for *Mm*, *Pt* and *Rc* respectively. Statistical difference is found between Roboception™ and Mech-Mind™. Finally, concerning the time taken the results were M = 1335.28 s, SD = 286.70, M = 792.14 s, SD = 193.64, and M = 473.78, SD = 120.20, for *Mm*, *Pt* and *Rc* respectively. Statistical difference is found for all the pairs. Out of this comparison, therefore, it is possible to understand that different bin-picking systems with their interfaces might have different impacts on the users. A higher level of usability and a lower workload can reduce the system programming time, as shown from the results of *Rc*. Additionally, the results show that the *Mm* system is largely underperforming in terms of usability compared to the other two. Moreover, *Mm* and *Rc* seem to have comparable user acceptance, yet with *Rc*, users were able to complete the task the fastest. Despite these results, which show the better performance of *Rc* in




**(a)** Usability **(b)** Workload **(c)** Autonomy **(d)** Time

**Figure 3.8:** Results of the conventional bin-picking evaluation with the user test. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed, or no statistical marking is shown. *Mm* stands for Mech-Mind™ Mech-Eye Pro S 1000M, *Pt* stands for Photoneo™ Phoxi M, and *Rc* stands for Roboception™ rc_viscore. From the evaluation, it is possible to see how Roboception™ rc_viscore can often outperform the other systems.

terms of usability. It is important to underline the replies to SUS questions four and ten, which are: "I think that I would need the support of a technical person to be able to use this system" and "I needed to learn many things before I could get going with this system" reported ratings above two for all systems. More precisely, using the inverse Likert rating, the recorded scores for question four are M = 4.28 pt, SD = 0.75, M = 2.71 pt, SD = 1.38, and M = 2.57 pt, SD = 0.79, for *Mm*, *Pt* and *Rc* respectively. Whereas for question ten are M = 4.14 pt, SD = 0.69, M = 2.71 pt, SD = 1.25, and M = 2.28 pt, SD = 0.95, for *Mm*, *Pt* and *Rc* respectively. Therefore, this underlined that learning to use any conventional bin-picking system involves several challenges and steps. Therefore, first-time users might be overwhelmed by such complexity; thus, the goal of programming and testing grasping points becomes hard. Therefore, as also highlighted in the introduction, there is a need to allow the definition of a grasping point in a simpler way, which allows the users to speed up the process of grasping point definition and its testing. In this way, the cycle time can be reduced. Additionally, if the grasping point definition is intuitive, users with lower skills in bin-picking can be up-skilled to configure such a complex system.

## 3.4 Definition of User-Defined Grasp Points

From the previous analysis, it is possible to identify that the interaction approach of the tested bin-picking systems has some flaws, as both the time required for the teaching and the usability could be improved. As seen from state-of-the-art, natural interaction approaches, like the ones provided with the Tracepen™, can improve the grasp point location definition. However, as shown from Sec. 3.3.1, such a natural interface must

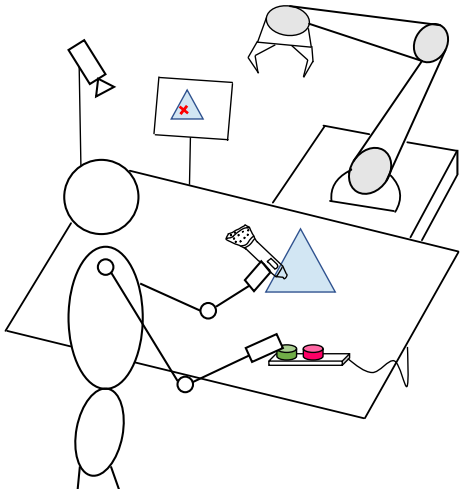


**Figure 3.9:** Schema of the user interaction. The user is in front of the target object to allow the user to define the grasping location. Then, using the Tracepen™ sensor, the user points to a position in an object satisfying the grasping requirements for the application. To confirm the location, the individual pushes an external button to avoid transmitting unwanted movement to the tracked sensor. Finally, the marked point is transmitted to the user via a Human Machine Interface (HMI) via a picture taken with a camera mounted on the top of the cobot cell.

be associated with additional refinement layers to reduce user errors. Therefore, this section presents a method to refine a user's grasp candidates by exploiting existing data-driven robotic grasping algorithms.

#### Envisioned Interaction

To achieve the goal of fine-tuning a grasp candidate, the following novel envisioned interaction is thus proposed. For clarity, the interaction is shown in Fig. 3.9. The user first places the part on the worktable center under a visible camera. Afterward, the individual grasps the spatial sensor and moves the device's tip to an appropriate object position for the robot to grasp, considering the task constraints. Finally, the decision is confirmed with an external button to avoid unwanted movements of the spatial sensor while saving the position. To conclude, the marked position is displayed to the user through an HMI, which shows a picture of the object taken by an overhead camera overlaid with the marked position. If the user is unsatisfied with the selection, the position can be canceled, and the procedure can start again. A movement to the grasp location is also initiated if the user wants to test the position with an actual robot. Upon the outcome, the grasp location can be redefined. Once the grasp position is finalized, it can be used program existing conventional bin-picking systems through their Application Programming Interface (API). Such set-up in the work cell is realized as shown in Fig. 3.11.

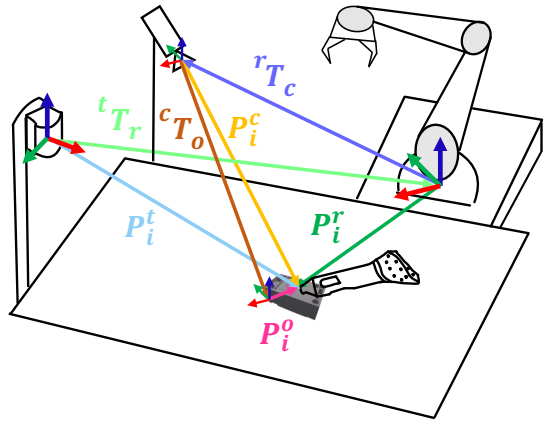



**Figure 3.10:** Illustration of the different coordinate frames involved in labeling a grasp point candidate with the spatial sensor. A point marked by the spatial sensor ($P_i^t$) must be transformed to a point in the image coordinate frame ($P_i^c$) through ${}^{r}T_{c}$ and ${}^{t}T_{r}$. Afterward, the point must be transformed in the object coordinate system ($P_i^o$) through ${}^{c}T_{o}$ to ensure that conventional bin-picking systems can use it.

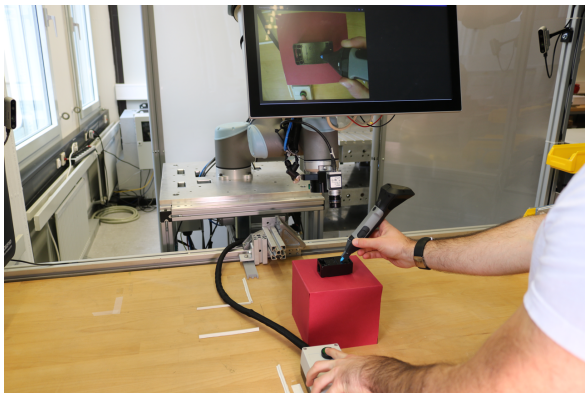


**Figure 3.11:** Example of grasp point candidate labeling using the Tracepen™. The user can mark points via the device and immediately receive feedback on the point location via the display, showing a live feed of the grasp point and the object from the robot's perspective. To avoid unwanted vibrations when labeling the grasp point, the user has an additional teach pendant with buttons for saving and deleting labeling points.

### Grasp Point Labeling and Coordinate Transformations

However, several technical hurdles need to be resolved to allow the integration of the target interaction. The first challenge regards the coordinate transformations between the robot, the camera, and the spatial sensor. To solve this issue, an approach similar to Sec. 3.3.1 and Sec. 3.3.2 is employed. However, some modifications have been added as the camera is mounted in an eye-to-hand configuration, as common in bin picking systems, and the object pose had to be accounted for. The problem is shown in Fig. 3.10. Therefore, first the (${}^{t}T_{r}$) is calculated by with Eq. 3.4. Then, ${}^{r}T_{c}$ is calculated with the algorithm proposed in [71]. Finally, by knowing ${}^{c}T_{o}$ through a previously trained 6D pose detection algorithm, it is possible to transform the labeled grasping point in the object coordinate frame with Eq. 3.7.

$$P_i^o = {}^{c}T_o \, {}^{r}T_c \, P_i^r, \; \forall \mathbf{i} \tag{3.7}$$

In this equation, the $i$ represents all the grasp-labeled candidates the user has sampled. Therefore, by employing this method, the labeled candidates can be utilized to program traditional bin-picking systems.

### Definition of a User Grasp Metric

Taking raw labeling pints from the user, however, might lead to unsatisfactory results as demonstrated in Sec. 3.3.1 and Sec. 3.3.2. Therefore, to ensure that robust grasp labeled points are selected, a refinement step after the user input is added. Instead of developing a new CNN as in the previous section, however, this refinement wants to exploit the existing algorithms in the state-of-the-art. Therefore, Dex-Net 4.0, one of the best-performing data-driven robot grasping algorithms, is selected for this purpose. As outlined before, Dex-Net 4.0 can select grasp candidates by applying computational steps. First, possible grasp candidates are sampled on a depth image using an analytical method described in [111]. Second, scores are given to the sampled grasps based on a GQ-CNN network. Therefore, if user input must be considered, constraining rules must be applied to one or more of the computational steps in Dex-Net 4.0. However, a definition of a constraining rule is necessary. As pointed out, the goal is to allow an operator to define a region where the grasp should be performed. In other words, a user has defined a point $\mathcal{P}_{\prime} \in \mathbb{R}^3$ on a surface $\mathcal{S}$ of an object $\mathcal{O}$, $\mathcal{S} \in \mathbb{R}^3 \wedge \mathcal{S} \in \mathcal{O}$, where the grasp should be executed. However, considering that $\mathcal{P}_{\prime}$ does not represent a precise position of a grasp due to human inaccuracies, it is possible to define that a neighborhood of such a point does. This can be then expressed with Eq. 3.8.

$$\mathcal{B}(\mathcal{P}_{\prime}, \epsilon) = \{\mathcal{P}_{\prime} \in \mathbb{R}^3 \text{ t.c } |\mathcal{P} - \mathcal{P}_{\prime}| < \epsilon\}, (\epsilon \in \mathbb{R}^3 \wedge \epsilon \in \mathcal{S}) \tag{3.8}$$

Therefore, to integrate the human initial guess, it is necessary to identify a grasp $\mathbf{u} \in \mathcal{U}$, $\mathbf{u} \in \mathbb{R}^3 \times \mathcal{S}^1$, which is contained in the boundary defined above for ensuring vicinity to the user initial guess. Additionally, such grasp should be the one with the highest probability of success, or adopting the nomenclature of Dex-Net 4.0, the one with the highest grasp quality or robustness, robustness or grasp quality is defined as $\mathcal{Q}(\mathbf{u}) \in [0,1]$. Thus, considering these additional boundaries, a novel metric denominated User Grasp Metric (UGM), $\pi(\mathcal{P}_{\prime}, \mathbf{u}), \pi \in [0,1]$, is proposed. The scope of the metric is to find the grasp $\mathbf{u}$ with maximum quality $\mathcal{Q}$ in the neighborhood $\mathcal{B}(\mathcal{P}_{\prime}, \epsilon)$ of the initial labeling candidate $\mathcal{P}_{\prime}$. To regard both terms equally, the UGM thus uses the geometric mean for two terms. However, as the UGM is $\pi \in [0,1]$, a definition of the boundary $\mathcal{B}(\mathcal{P}_{\prime}), \epsilon \in [0,1]$ is necessary. For this goal, the boundary is expressed as the inverse of the normalized Euclidean distance between the labeling candidate $\mathcal{P}_{\prime}$ and the grasp $\mathbf{u}$. This yields the following mathematical formulation for the UGM represented in Eq. 3.9.

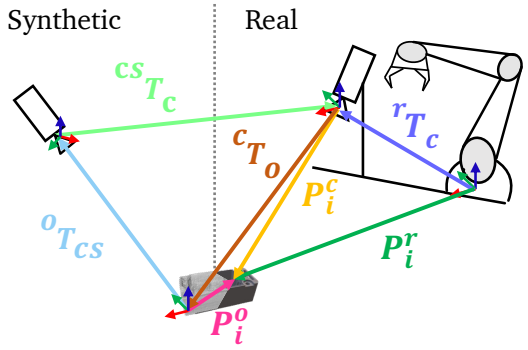



**Figure 3.12:** Transformations involved using a synthetic camera for sampling grasps with DexNet 4.0. First, the object pose is obtained via a 6D pose detector. This information is then used to project the object mesh in the synthetical world. Next, the synthetical camera, with camera parameters such as the one used to train Dex-Net 4.0, is used to sample grasp candidates and find the one with the highest UGM. In the end, exploiting the information of the object position in the synthetical $^{o}T_{cs}$ and real-world $^{c}T_{o}$ the selected grasp is projected back in the real world by using $^{cs}T_{c}$. Finally, the robot executes the grasp by knowing the camera position $^{r}T_{c}$.

$$\pi(\mathcal{P}_\prime, \mathbf{u}) = \sqrt{\mathcal{Q}(\mathbf{u})\left(1 - \frac{d(\mathbf{u}, \mathcal{P}_\prime)}{\epsilon}\right)} \tag{3.9}$$

Therefore, the grasp $\mathbf{u}$ to best fulfill robustness and vicinity of the user input should be the one with the highest $\pi_f(\mathcal{P}_\prime, \mathbf{u})$. This can be written as in Eq. 3.10.

$$\pi_f(\mathcal{P}_\prime, \mathbf{u}) = \text{argmax}_{\mathbf{u}\in\mathcal{U}}\pi(\mathcal{P}_\prime, \mathbf{u}) \tag{3.10}$$

#### Implementation of the User Grasp Metric

Having a mathematical formulation of the UGM, the final step is its integration of the cobot cell. As pointed out previously, the sampling of the grasp candidates is performed via the Tracepen™ as long it allows simple integration of natural interaction and sampling points in $\mathbb{R}^3$. However, for the computation of the quality $\mathcal{Q}(\mathbf{u})$, it is important to denote the generalization capabilities of Dex-Net 4.0. As pointed out in [90, 92], Dex-Net 4.0 performances can suffer if the camera used for obtaining depth images is placed in a different position than the set-up used by the authors for generating training data. To overcome this issue, a novel depth generation pipeline is thus proposed to guarantee maximum compatibility with the existing network without retraining. The pipeline is based on the concept of synthetical data. More precisely, the idea behind this pipeline is to generate synthetical depth data that represents the target object with a camera configuration similar to the one used in Dex-Net training. However, a real-world picture must be transformed into a synthetical one to achieve this goal. A novel data generation approach based on transformations is thus proposed. The concept be-

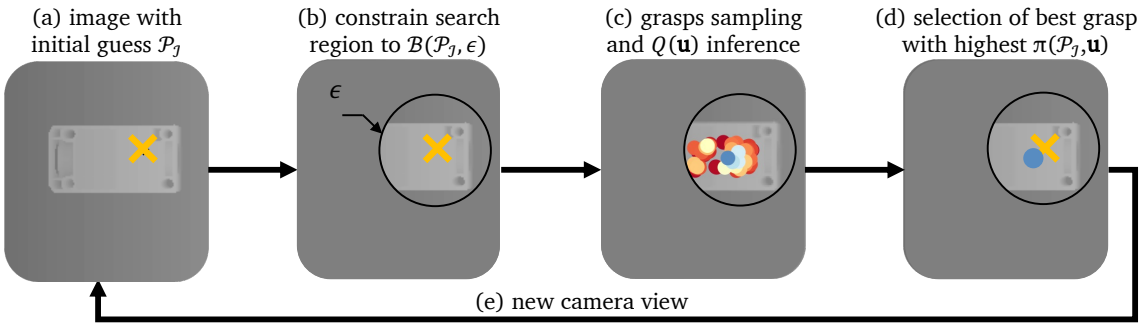



**Figure 3.13:** Qualitative representation of the sampling process using Dex-Net 4.0. First, a synthetic camera image is taken alongside its projection of the user grasped labeled point (a). Second, to speed up the computation time, the image is masked to the search region close to the user input $\mathcal{B}(\mathcal{P}_\prime, \epsilon)$ (b). Third, Dex-Net 4.0 is used to sample grasps and compute their raw quality $\mathcal{Q}(\mathbf{u})$ (c). Fourth, the grasp $u$ with highest $\pi(\mathcal{P}_\prime, \mathbf{u})$ is selected and saved (d). Finally, the process is reiterated for multiple camera views to ensure the best $\pi_f(\mathcal{P}_\prime, \mathbf{u})$ in the boundary $\mathcal{B}(\mathcal{P}_\prime, \epsilon)$(e).

hind the approach is shown in Fig. 3.12. To apply this method, first, the object pose with respect to the real camera ${}^{c}T_{o}$ must be sampled via a 6D pose detector by the conventional bin-picking system. Afterward, the object is projected in the synthetical world origin alongside the grasp labeled point $P^{o} = \mathcal{P}_\prime$. Then, the synthetical camera samples a depth image, and Dex-Net 4.0 is used to sample grasps $\mathbf{u}$ in the image. Finally, the grasp $\mathbf{u}$ with highest $\pi(\mathcal{P}_\prime, \mathbf{u})$ according to Eq. 3.10 is selected. Having this grasp, its pose is used for robot execution via the transformation ${}^{cs}T_{c}$ computed with Eq. 3.11.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = K^{-1} {}^{r}T_{c} \, {}^{c}T_{o} \, {}^{o}T_{cs} \begin{bmatrix} u' \\ v' \\ w' \\ 1 \end{bmatrix} \tag{3.11}$$

Where $[x, y, z]$ are the grasp coordinates in the robot coordinate frame, $K$ are the intrinsic camera parameters, and $[u', v', w']$ are the pixel coordinates of the grasp with highest $\pi(\mathcal{P}_\prime, \mathbf{u})$. However, denoting three points for applying this method and its speed computation is essential. First, the parameter $\epsilon$ had to be selected. For this, the maximum allowable object size of the robot EE is chosen. To speed computational complexity, thus, when Dex-Net 4.0 searches for all the possible grasps $\mathbf{u}$, the area of search is constrained in $\mathcal{B}(\mathcal{P}_\prime, \epsilon)$. Second, for using the synthetical camera sampling and a 6D pose detector, the object digital representation, i.e., object mesh, is necessary. Third, as long as the camera is in the synthetic world, several camera views are evaluated within the boundaries of Dex-Net 4.0 to select the best grasp out of several views. This whole process with qualitative images is shown in Fig. 3.13.

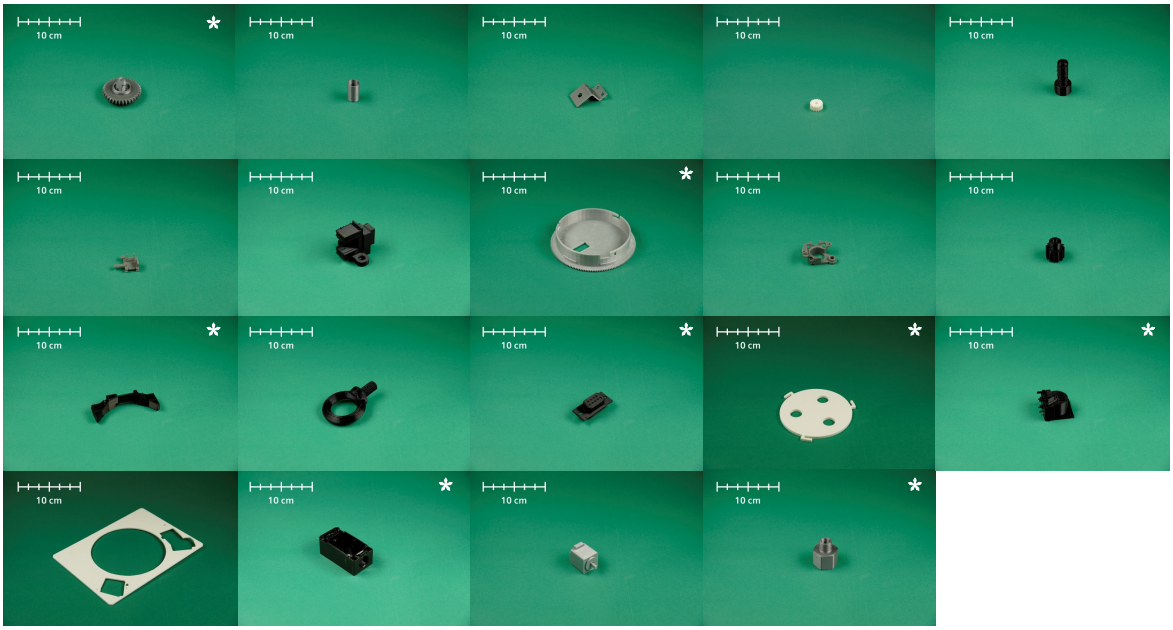


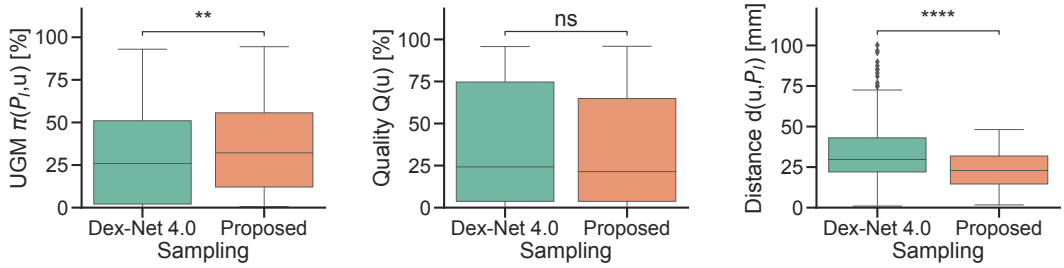
**Figure 3.14:** Objects selected for the virtual and physical evaluations. The objects are selected as long they represent objects common in industrial scenarios. The objects marked with an asterisk on the top right are also used for the physical evaluations.

## 3.5 Exploring User-Defined Grasp Points: Benefits and Performances

Having a method now to refine grasp labeling candidates given by a user with the best UGM $\pi_f(\mathcal{P}_\prime, \mathbf{u})$ the next step is to evaluate the accuracy and performances of this method. For doing so, three sets of experiments are thus conceived. The first set is formed by virtual experiments conducted in the simulator to asses robustness and differences of UGM $\pi_f(\mathcal{P}_\prime, \mathbf{u})$ again the raw quality of Dex-Net 4.0 $\mathcal{Q}(\mathbf{u})$ in a simulation environment. The second set of experiments is constituted by physical evaluations conducted with a real robot to assess the successful picking rate of UGM versus raw quality as suggested by [87]. Finally, the last experiment is a user study to evaluate the usability and workload using the proposed natural interaction.

### 3.5.1 Virtual Experiments

For this set of experiments, a set of different objects from three industrial datasets [112–114] are selected, summing up to a total of 19 objects as shown in Fig. 3.14. Then, the quality calculated with the two methods with random grasp label points is assessed. More precisely, depth images for all the objects were generated, and afterward, for each object, grasp label points with a uniform distribution on the object surface were

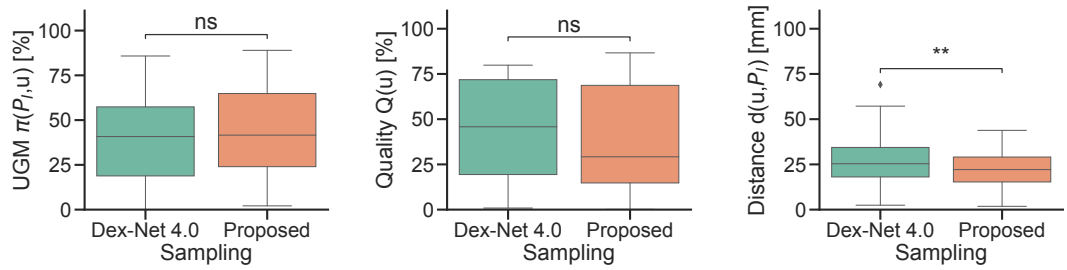



**(a)** User Grasp Metric **(b)** Raw quality **(c)** Distance

**Figure 3.15:** Results of the three involved metrics for evaluating a sampled grasp with either raw Dex-Net 4.0 or with the proposed metric UGM for a suction EE. To the left UGM value $\pi_f(\mathcal{P}_l, \mathbf{u})$, in the center raw quality $\mathcal{Q}(\mathbf{u})$, and to the right, distance from sampled grasp and user labeled grasp $d(\mathbf{u}, \mathcal{P}_l)$. For the suction EE, the proposed sampling method selects grasps with the same raw quality as Dex-Net 4.0 but simultaneously closer to the user-labeled grasp.

generated. Finally, the UGM of the final grasp, either with Dex-Net 4.0 or with the presented approach, is compared. The results for a suction EE are shown in Fig. 3.15, whereas the results for an antipodal EE are shown in Fig. 3.16. Concerning the suction EE, it is possible to understand that adopting the proposed metric UGM grasps, which are closer to the user-labeled grasp point but have the same probability of success as Dex-Net 4.0, can be selected. More precisely, looking the UGM score $\pi_f(\mathcal{P}_l, \mathbf{u})$ yielded M = 29.16%, SD = 27.14, and 34.05%, SD = 24.66 for Dex-Net 4.0 and the proposed approach respectively. Second, looking at raw grasp quality $\mathcal{Q}(\mathbf{u})$ the results are M = 35.98%, 33.41, and M = 32.43%, SD = 32.69 for Dex-Net 4.0 and the proposed approach respectively. Last, looking at the distance between the final grasp and the user-labeled grasp point $d(\mathbf{u}, \mathcal{P}_l)$ the results yielded M = 34.44 mm, SD = 20.89, and M = 23.00 mm, SD = 11.41 for Dex-Net 4.0 and the proposed approach respectively. Therefore, the statistical analysis finds a difference in the distance of grasps sampled by the two methods, but no statistical difference is found in the raw quality of the grasps. The proposed UGM can be well used for a suction EE to better select grasps that best fit the user's initial label while guaranteeing grasp robustness.

UGM score $\pi_f(\mathcal{P}_l, \mathbf{u})$ yielded M = 38.85%, SD = 23.19, and M = 42.76%, SD = 22.85 for Dex-Net 4.0 and the proposed approach respectively. Second, outcomes for the raw grasp quality $\mathcal{Q}(\mathbf{u})$ yielded M = 44.14%, SD = 28.86, and M = 39.09%, and SD = 28.24 for Dex-Net 4.0 and the proposed approach respectively. Last, evaluation of the distance $d(\mathbf{u}, \mathcal{P}_l)$ are M = 27.17 mm, SD = 12.95, and M = 21.76 mm, SD = 9.52 for Dex-Net 4.0 and the proposed approach respectively. Thus, considering that statistical significance is found only for results regarding the distance, the UGM score was not good enough for selecting grasps in the case of an antipodal EE. Therefore, raw Dex-Net 4.0 would perform as the proposed approach. Thus, the UGM is not beneficial




**(a)** User Grasp Metric **(b)** Raw quality **(c)** Distance

**Figure 3.16:** Results of the three involved metrics for evaluating a sampled grasp with either raw Dex-Net 4.0 or with the proposed metric UGM for an antipodal EE. To the left UGM value $\pi_f(\mathcal{P}_l, \mathbf{u})$, in the center raw quality $\mathcal{Q}(\mathbf{u})$, and to the right, distance from sampled grasp and user labeled grasp $d(\mathbf{u}, \mathcal{P}_l)$. For the antipodal EE, the proposed UGM does not lead to any meaningful differences when compared with the baseline Dex-Net 4.0.

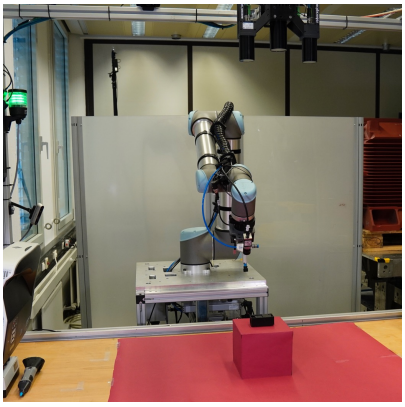
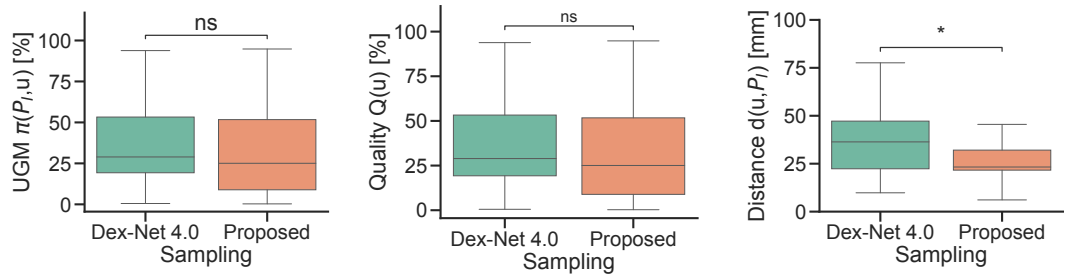


**Figure 3.17:** Example of set-up for the physical experiments. The robot was configured with a suction EE with a random part of the set in front of it. Afterward, a random user point close to the object's center of mass was inserted, and the system executed the calculated grasp with UGM.

for these cases.

### 3.5.2 Physical Experiments

The virtual evaluation highlighted that the proposed UGM for selecting candidate grasps could be beneficial in case of a suction EE to select robust grasps while guaranteeing they are in the vicinity of an initially labeled grasp guess given by an inexperienced operator. However, to ensure the correctness of these findings in the virtual experiment, a physical experiment with a real robot with a suction EE was envisioned. The experiment setup can be seen in Fig. 3.17. Therefore, a subset of eight objects out of those used for the virtual evaluation was selected, as shown from Fig. 3.14, and grasps were tested. More precisely, to guarantee results that might respect real-




**(a)** User Grasp Metric **(b)** Raw quality **(c)** Distance

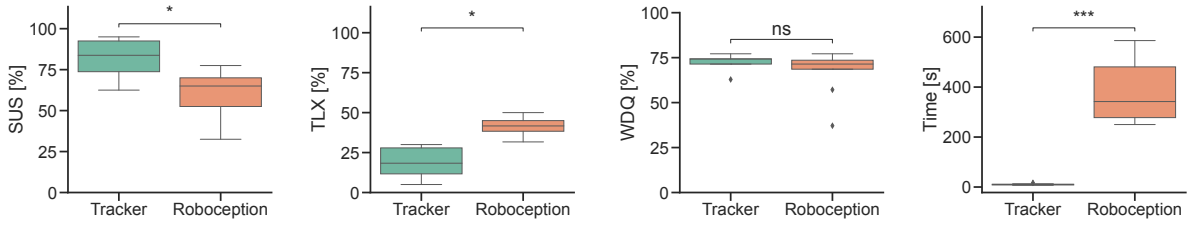
**Figure 3.18:** Results of the three involved metrics for evaluating a sampled grasp with either raw Dex-Net 4.0 or with the proposed metric UGM for the suction EE in physical experiments. To the left UGM value $\pi_f(\mathcal{P}_l, \mathbf{u})$, in the center raw quality $\mathcal{Q}(\mathbf{u})$, and to the right, distance from sampled grasp and user labeled grasp $d(\mathbf{u}, \mathcal{P}_l)$. For the suction EE, the proposed UGM does not lead to any meaningful differences when compared with the baseline Dex-Net 4.0. However, the distance between the final selected grasp and the user grasp label point is meaningful.

world usage, three input grasp label points at the center and edges of each object were marked, and evaluation was performed accordingly for a total of 24 trials. The outcomes of these trials are shown in Fig. 3.18.

Differently from the virtual evaluations, no meaningful difference is found in the value of the UGM for the physical experiments; however, results for the raw quality $\mathcal{Q}(\mathbf{u})$ and distance $d(\mathbf{u}, \mathcal{P}_l)$ are comparable to the previous experiments. More precisely, results for the UGM score $\pi_f(\mathcal{P}_l, \mathbf{u})$ yielded M = 29.29%, SD = 29.37, and M = 36.72%, SD = 26.29 for Dex-Net 4.0 and the proposed approach respectively. Second, results for the raw grasp quality $\mathcal{Q}(\mathbf{u})$ yielded M = 39.38%, SD = 33.54, and M = 35.95%, and SD = 35.63 for Dex-Net 4.0 and the proposed approach respectively. Last, results for the distance $d(\mathbf{u}, \mathcal{P}_l)$ are M = 36.23 mm, SD = 18.15, and M = 25.94 mm, SD = 9.54 for Dex-Net 4.0 and the proposed approach respectively. To shed more light on these results, the successful grasping success rate defined by [87] was monitored during the physical evaluation. This yielded 83.3% for Dex-Net 4.0 and 87.5% for the proposed approach. Therefore, it is possible to deduce that, despite not having found a statistical difference for the UGM value, the proposed method for selecting the grasp position can improve the grasp success rate while guaranteeing selecting grasps that are closer to the user grasp label when compared with Dex-Net 4.0.

### 3.5.3 User Study

Having identified that the proposed approach can select grasps that are closer to an initial guess given by an inexperienced operator, the next step is to assess the proposed method in terms of usability and compare it with conventional systems as outlined




**(a)** Usability **(b)** Workload **(c)** Autonomy **(d)** Time

**Figure 3.19:** Results of the user experiment comparing the proposed method based on the tracker (i.e., spatial sensor) and the best performing conventional bin-picking system Roboception$^{TM}$ rc_viscore. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed, or no statistical marking is shown. From the evaluation, it is possible to see how using a tracker for grasp labeling positively affects usability, workload, and time. No difference is found in the level of autonomy.

in Sec. 3.3.3. Therefore, a user study experiment with one object was conducted for this scope. In the study, differently from the previous investigation, users had the goal of labeling a suction grasp point with two different methods: the proposed approach with the spatial sensor and the best performing conventional bin-picking system Roboception$^{TM}$ rc_viscore. After each interaction, users were administered the SUS, TLX, and WDQ. Additionally, the time taken was monitored. In the user study, ten male users with low to medium experience in robotic grasping, age M = 30.2 yrs, SD = 11.78, participated. All of them performed the task correctly, and no data was discarded. The outcomes of the user test is shown in Fig. 3.19.

The results show that using a tracker (i.e., spatial sensor) can benefit usability, workload, and time but not on the level of autonomy. More precisely, the results yielded the following. For SUS, the scores are M = 59.50%, SD = 17.62, M = 82.25%, and SD = 11.87 for Roboception and tracker, respectively. For TLX, the scores are M = 41.33%, SD = 6.91, M = 19.17%, and SD = 9.34 for Roboception and tracker, respectively. For WDQ, the scores are M = 67.14%, SD = 11.84, and M = 72.00%, SD = 5.50 for Roboception and tracker, respectively. For the time, the outcome is M = 374.70 s, SD = 123.27, M = 10.40 s, and SD = 3.03 for Roboception and tracker. Thus, considering that statistical significance was found for SUS, TLX, and time, it is possible to conclude that using the tracker can be better perceived than conventional bin-picking systems. Additionally, looking at questions four and ten of the SUS questionnaire, the results for tracker are, using an inverse Likert scale, M = 1.50 pt, SD = 0.53, and M = 1.30 pt, SD = 0.48 for questions four and ten, respectively. Therefore, considering the results obtained in Sec. 3.3.3, it is possible also to add that the tracker system results less cumbersome and less information is needed by a person to use it, thus allowing any user to quickly learn and use the system.

| Type | mAP(IoU=0.5) | Pos. Error [mm] | Rot. Error [deg] |
|---|---|---|---|
| CAD | 0.94 | 0.27 ± 0.91 | 1.47 ± 14.40 |
| SIM NeRF | 0.74 | 0.73 ± 1.15 | 4.58 ± 25.31 |
| REAL NeRF | 0.85 | 1.39 ± 3.12 | 9.63 ± 38.11 |

**Table 3.1:** Outcomes of the evaluation of the CADMatch 6D pose algorithm trained with different kind of source data (± denotes the standard deviation).

## 3.6 Conclusions

### Summary

Through this chapter, this work wanted to discover further impacts given by intuitive programming interfaces. More specifically, this section tackled *Challenge 3.2* (intuitive programming interfaces that do not require robotics expertise) by proposing and studying the impacts of an intuitive method for defining robotic grasping points via spatial interaction.

To present these investigations, initially, the current state-of-the-art and challenges identified in defining grasping locations according to user needs were introduced, and the need for more accessible programming of grasps considering task constraints without robot experts was discovered. Second, the chapter presented an approach for labeling points based on raw data from a spatial sensor, but limited sensor accuracy was discovered. Therefore, the third section presented improved labeling by fusing visual and spatial sensor data to correct the inaccuracies. This approach resulted in more success. Henceforth, using the knowledge of the previous two experiences, a final method to best fuse an initial grasp guess and a state-of-the-art visual NN based on a novel user grasp metric for grasp selection is presented. This last approach presents studies to evaluate the method's accuracy regarding absolute positioning and user impacts compared with conventional market-ready solutions.

The proposed approach for grasp labeling showed that the usability and workload are improved with the proposed method. More precisely, the usability measured with the SUS scored an average of 82.25%, and the average workload measured with TLX scored 19.17%. Aside from the higher usability, the investigations yielded results that showed higher or equal grasp reliability when comparing grasps programmed with the proposed method against conventional approaches, as the grasp success rate was 87.5%.

### Discussions and Future Work

Based on the usability results, it is possible to conclude that a broad user pool can accept the proposed method for defining grasps [115]. Additionally, when comparing test results with traditional bin-picking systems, which showed equal or worse performance, the barrier for grasp definition considering task constraints has been lowered. This reduction in required knowledge makes it possible for non-experts to program robotic grasp points. Beyond the enhanced usability, improvements in accuracy were also observed. Therefore, the suggested approach, incorporating the user grasp metric, proves effective in achieving both usability and accuracy improvements. Furthermore, despite testing with a single state-of-the-art neural network and hardware setup, the method's applicability can extend to various use cases and applications. This adaptability is attributed to the approach based on a synthetic image pipeline, allowing it to go beyond current research or specific hardware setups as highlighted by [90, 92]. Last, being the approach usable with low robotics expertise as the user does not need to know in detail about the constraints of grasping the benefits of the approach, and it is possible to conclude that with this tool, SMEs can partially program robots without the need for robotic experts thus answering to the aforementioned challenge.

Unfortunately, despite this approach's benefits, it is essential to underline that this method requires the CAD model of the object to train 6D pose detectors for the object. Therefore, if an SME does not have this available due to manufacturing limitations, this method cannot be applied anymore. To overcome this issue, the creation of CAD artifacts of the target object is needed. Neural Radiance Fields (NeRF) [116] can be used for this task as suggested by [117]. To ensure this approach could be used in future work, a small set of experiments to start understanding the limits of this method was conducted. For such testing, one object out of our dataset was selected, and the Roboception$^{TM}$ CADMatch 6D pose algorithm was trained with CAD data, NeRF from simulation, and NeRF from real-world data generated with the Eyecan$^{TM}$ scanning station. Afterward, a test set of 68 images from the simulation was taken, and the metrics of mAP at IoU $= 0.5$ and errors in rotational and positional accuracy were measured. The scores are shown in Tab. 3.1. From the results, it is possible to draw two outcomes. First, the detection rates are slightly lower in the case of NeRF reconstructions, and in the case of bin-picking applications, this might result in a not-complete depletion of the bin. Second, the positional error is not too different between the different models. However, the rotational error is still considerable for the parts scanned. The results on the performances of the pose detection are not as promising as the user tests; however, fair results have been achieved. Therefore, future works should further investigate

how to improve the scanning capabilities of NERF to ensure the integration of objects without CAD data.

Chapter 4

# Influences of User-Defined Trajectories on Robotic Safety

Contents



The previous chapters presented two studies on how defying part locations influences the robot and the user. However, the studies did not account for the entire motion of the robot but just for final locations, thus limiting the operator's level of control of the robot's motion. Considering that this might be a crucial acceptance factor considering literature, there is a need to investigate this matter further. This chapter looks into how trajectories defined by operators affect the interaction. This chapter is partly based on the content of the following publications:

- Pantano, M., Regulin, D., Lutz, B., and Lee, D., "A human-cyber-physical system approach to lean automation using an industrie 4.0 reference architecture", 2020, Procedia Manufacturing, Volume 51.

- Pantano, M., Blumberg, A., Regulin, D., Hauser, T., Saenz, J., and Lee, D., "Design of a Collaborative Modular End Effector Considering Human Values and Safety Requirements for Industrial Use Cases," 2022, Human-Friendly Robotics 2021, Springer Proceedings in Advanced Robotics, Volume 23.
- Pantano, M., Schmidt, M., Bolano G., Schulenburg, E., Regulin, D., and Saenz, J., "Analyzing the Influence of Self-defined Trajectories on Safety and Task Ownership: An Empirical Study," 2023, industrial paper at the 19th IEEE Conference of Automation Science and Engineering (CASE).

To present the content of this chapter, first, the motivation behind the need to study the influence given by trajectories on human operators is presented. Next, the main related works are identified, together with two preliminary investigations necessary to identify which method is suitable for teaching trajectories. Third, a novel method for teaching trajectories through spatial interactions is proposed, and last, the evaluation is presented with final remarks and discussions.

## 4.1 Motivation

In the previous chapters, two studies on how defying part locations influence the robot and the user. However, this definition did not account for the entire motion of the robot but just the final locations. Therefore, trajectories were not explicitly addressed by the users. However, the literature expressly identifies that trajectories strongly influence the perceived safety and human-robot collaboration [118]. One of the most known research works in this field is regarding the legibility of robot motion [119]. In this research, the authors discovered that the robot should make its intentions transparent for seamless human-robot collaboration, meaning the human collaborator can easily understand the robot's next steps. However, their studies identified that the observer can understand legibility differently. Therefore, there is a need to investigate the topic better, considering a human observer. Building on the previous works, which defines that humans better collaborate with robots if they define their task [46]. This section investigates how human-defined trajectories impact motion legibility and robot and human performances.

## 4.2 Related Works

### Trajectories Legibility

Humans have natural boundaries in defining acceptable distances for Human-Human interactions as described by the Proxemics model [120]. Later studies have found that this applies also to HRC. For example, in [121], the authors identified that once the cobot entered the personal space between 0.46 m and 1.20 m, humans backed up to increase their distance from the manipulator. Additionally, the study reported that the users' comfort level decreased when the robot moved closer. To improve human comfort, studies proposed several motion planning frameworks to reduce robots' disruption when entering personal space. One of the first ground research works on this topic was done by [122]. In this work, the researchers identified that any possibility of a robot crossing the path of a human reduced the level of human comfort. Additionally, robot paths were classified with a level of legibility for understanding their impacts on user comfort, and often, higher motion legibility results in higher human comfort. The authors then took a further step in a more formal definition of legibility in [119]. In this research, a legible motion was defined as a movement from which an observer can confidently infer its goal. The quicker the final goal deduction, the higher the motion legibility. Legibility is also known in the literature as readable motion [123]. Since this discovery, robots showing higher legibility are better perceived in different use cases, like virtual reality (VR) applications [124]. Therefore, the uncertainty of robot movements must be avoided to ensure user well-being [125].

### Adaptive Robot Motions

Another approach to improve the comfort level in motions performed in HRC is to include adaptive robot movements rather than legible motions. One of the first research to report the benefit of adaptive trajectories is presented in [126]. In this work, the authors propose a robot control algorithm that plans motions that do not enter the personal space or cross the human path by monitoring the human position. Through this approach, they demonstrate that human-aware planning can increase human comfort and decrease task cycle time. Other works achieve the same goal by including a cost function in the motion planner to select paths without collision with humans. Notable examples are presented in [127, 128]. Similarly, [129] proposes a cost function that penalizes areas previously occupied by humans. However, despite being this useful for collaboration and modeling, these cost functions have yet to consider delays in the collaboration itself [130]. Therefore, in [130], a novel cost function for accounting for these stops has been proposed. The robot can maximize the cycle time through this modeling, even considering stops due to safety regulations; however, despite the

advantages of these planners based on a cost function. There is still a need to reduce the uncertainty of robot motions. In the previous work, this has been mainly addressed by predictable motion planners [118]. However, these follow trajectories often not defined by the operator working with the cobot. Therefore, the operator might feel excluded from the task, thus raising feelings of uncertainties [131]. So, similar to the above, an open question remains on how robot trajectories can be defined considering the task ownership of the observer to reduce uncertainties.

#### Teaching of Robot Motions

Different methodologies can be used to create robot trajectories, as seen in Sec. 1.2. However, once an operator must teach the trajectories, the number of methodologies decreases due to the need for simple interfaces. Out of the known programming methods, the best for this task is LfD [132, 133]. In this approach, the robot's motion should be obtained by capturing trajectories from demonstrators through different user interfaces [134]. The most known are kinesthetic teaching [135] and spatial-sensor demonstrations [134]. In the former, the robot's physical movement is recorded to obtain the movements, thus avoiding transforming the information through different coordinate systems. Such physical movement is obtained by having a human guiding the manipulator via pushing or pulling the EE and recording different sensor data from the manipulator (e.g., force, position) during the procedure. Then, the trajectory is reproduced, trying to mimic the recorded behavior. However, despite the method being intuitive to teach and has demonstrated to achieve sound movement reproduction and recovery [136]. Kinesthetic teaching user acceptance could be influenced by the force control algorithm that balances the robot weight [29]. The latter, spatial-sensor demonstration, is more complex in the coordinate transformations. However, it uses more natural methodologies for teaching, like motion-tracked gloves [137]. Commercial solutions based on motion-tracked pens are also available for different robot brands. Notable products are the Tracepen™ and the Mimic™. However, these solutions often base the recording on positions, and their accuracy has yet to be investigated in detail. Nevertheless, spatial-sensor demonstrations and kinesthetic teaching can help an operator teach robot paths. However, to the author's knowledge, there is no study on the effect of kinesthetic or spatial-sensor demonstrations on the user's acceptance or the robot's safety. Therefore, there is a need for further investigation on this topic.

#### Trajectories and Safety in HRC

As humans and robots collaborate closely within factory settings, an increasing emphasis is placed on ensuring human safety. Typically, safety considerations involve a thorough risk assessment (as exemplified by ISO 12100:2010 [16]) and a rigorous adherence to established international standards like ISO 10218-2 [18] and ISO/TS 15066 [23]. Within Europe, the Machinery Directive [138] outlines the indispensable health and safety prerequisites that a robotic application must meet. Among the four principal collaborative operation modes delineated in ISO/TS 15066, SSM is one of the most widely implemented choices for collaborative robotics applications. In this operational mode, a sensor is employed to constantly monitor the separation distance between nearby humans and the robot. Accordingly, adjustments can be made to the robot's speed and trajectory to uphold a minimum safe distance. Therefore, in SSM, the robot's trajectory is significant regarding safety as long it can define the separation or the robot's speed during the application. Earlier research has primarily concentrated on establishing a foundation for robotic applications employing SSM [20]. This has encompassed efforts to enhance its dynamism [139] and the development of sensor implementations [140] crucial for enabling this operational mode. Subsequent investigations have utilized simulation techniques to assist in planning such applications [141]. Additionally, they have showcased the utilization of key performance indicators (KPIs) in design, such as overall spatial requirements and the minimal separation distances necessary. These indicators have been instrumental in evaluating different possibilities [142], encompassing safety sensor selection and robot trajectory choices. Therefore, these techniques can be used to rate the safety level of an application once trajectories are made available.

## 4.3 Preliminaries

To ensure that users can program their trajectories safely, the robot EE should first undergo defined tests according to the safety guidelines; additionally, the users should have interfaces for teaching their trajectories. This section first presents how a robotic EE can be designed and tested safely for this collaboration, following the actual regulations for implementing safe human-robotic applications. The second part describes how a spatial sensor can allow users to define their trajectories using the Tracepen™.

### 4.3.1 Risk Mitigation Measures for Cobot Trajectories

Before applying SSM, to evaluate the current level of safety, each area that can cause a collision must be evaluated regarding the maximum allowable pressure exerted at a certain speed. In the case of cobots, the most dangerous area for the operator is the EE [143, 144]. The EE is the component of a robotic system in charge of directly interacting with the surrounding environment (e.g., moving, grasping) through the application of a specific physical principle (e.g., impactive) as outlined by ISO 20218-1 [145]. Therefore, in the case of collaborative applications, the EE should satisfy both safety constraints and application requirements. In the case of SMEs, a flexible EE that can grasp different parts should be conceived [38].

#### Design Methodology

To ensure a design considering all the requirements for both flexibility and safety, the iterative design methodology proposed in [146] was followed. First, the tasks were examined, and requirements were obtained. For this step, publications and standards targeting flexible grasping were targeted, namely [23, 84, 145, 147]. Second, several abstract design solutions were created by combining different design primitives. In this process, physical principles were evaluated and selected. Third, morphological structures (i.e., EE concepts) were created. Finally, concepts were evaluated from technical and user perspectives.

#### Users and Technical Evaluations

The evaluation method proposed in VDI 2225 [148] was selected, which suggests rating mechanical designs according to technical soundness and economic viability. However, the economic viability axis was substituted with a human value score to reflect the integration of human values to evaluate the different EE concepts created in the process. The method is based on two scales, rated from 0 to 4 (4 being the best rating). When evaluating concepts, scores should be given to each of the two scales. In the end, concepts with the highest compound scores are the ones that best suit the application needs. In this evaluation, the two selected scores were technical performance (e.g., cost, manufacturing efficiency) and human aspects (e.g., human wellbeing, universal usability). Therefore, a pool of 12 males with M = 9.5 yrs SD = 10 of experience in engineering were invited to reply to a questionnaire. In the questionnaire, each concept had to be evaluated on the different scores using a Likert scale [149]. The outcome of the questionnaire is shown in Fig. 4.1. The concepts with the highest scores

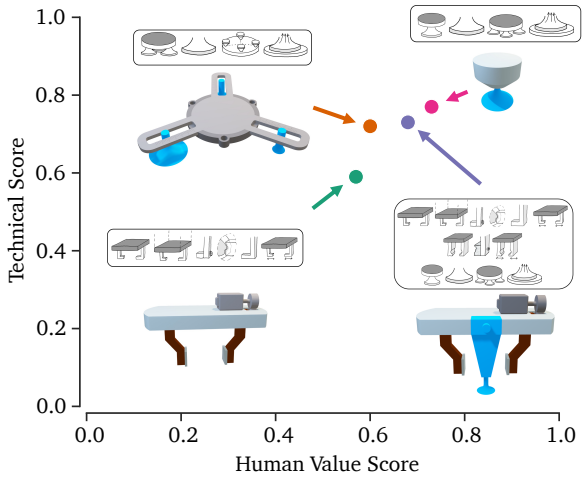



**Figure 4.1:** Outcomes of the evaluations of the four different prototypes. The evaluation methodology followed the method of VDI2225 adapted to include human values.

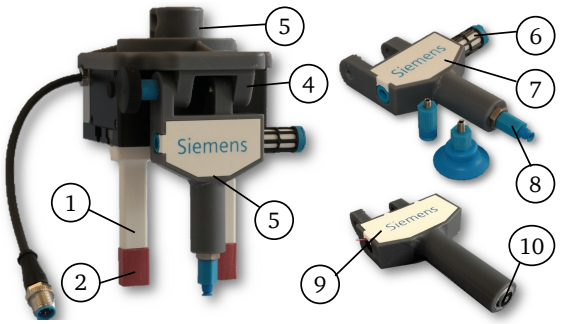



**Figure 4.2:** Manufactured hybrid EE with modules. Reprinted from Design of a Collaborative Modular End Effector Considering Human Values and Safety Requirements for Industrial Use Cases, Human Friendly Robotics 2021, Pantano M. et al. [114], Copyright (2022), with permission from Springer.

were the vacuum and hybrid gripper with vacuum and two-finger jaws. The hybrid gripper was selected for manufacturing due to the higher flexibility to adapt to several geometries. Therefore, the concept was manufactured using additive manufacturing (AM) technology and the integration of market-ready solutions. The final EE, shown in Fig. 4.2, was obtained through this process. Considering the numbering in Fig. 4.2, the EE can be described as follows. The main structure is connected to an adapter plate (3) with different mounting slots for an off-the-shelf two-finger jaw module and an angular adjuster (4) for additional modules is necessary. Then, to customize the grip of the two-finger jaw module, 3D-printed interchangeable jaws (1) with silicone tips (2) are added. Third, two additional modules, (7) and (9), are added to integrate different physical principles. On the one hand, vacuum, and the other hand, magnetic. These modules were primarily designed to maintain the Tool Center Point (TCP) of the robot constant despite tip (8) changes.

### Safety Testing

To evaluate the risk given by the EE design, a risk analysis had to be performed according to ISO 12100 [16]. Therefore, hazards had to be first identified, and then risk mitigation measures had to be integrated. In the case of an EE for a flexible cobot cell with close contact with humans, many hazards are either given by the robot motion or the actuation of one of the physical principles of the EE. To mitigate these hazards, it was then decided to program the robot with the PFL risk reduction method and limit the object mass handled by the cobot. However, for the integration of PFL, limit veloc-

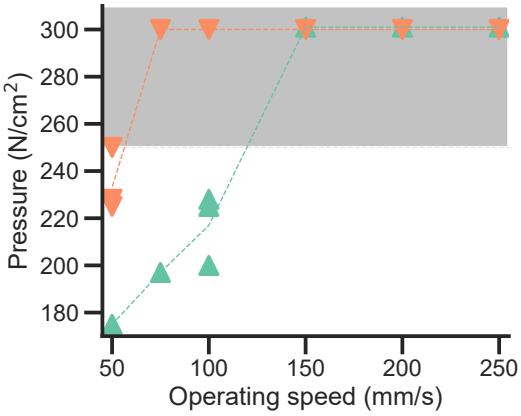



**Figure 4.3:** Outcomes of the safety evaluations. The two lines represent the measured contact pressure for two different gripper locations. Dangerous values are measured above 250 $N/cm^2$. Reprinted from Design of a Collaborative Modular End Effector Considering Human Values and Safety Requirements for Industrial Use Cases, Human Friendly Robotics 2021, Pantano M. et al. [114], Copyright (2022), with permission from Springer.

ities had to be identified according to the EE minimum contact area and the maximum allowable contact pressure. The COVR [150] GRI-LIE-1 protocol was followed to proceed with this evaluation. Therefore, the robot was programmed to travel at different speeds and collide with pressure foils. This procedure was repeated several times for two of the smallest EE contact areas until the limit contact pressure constraint was satisfied. The results of this study are shown in Fig. 4.3.

From this evaluation, considering the worst-case scenario, it is possible to deduce that the designed EE can be considered safe only when the robot moves at 50 mm/s. Therefore, whenever an operator is near the robot, the manipulator must be slowed down to this velocity to reduce hazards to acceptable levels.

### 4.3.2 Trajectory Teaching via Spatial Interactions

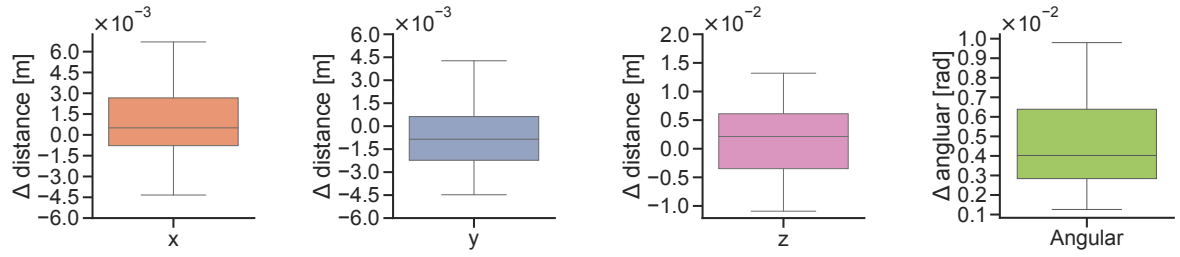
As highlighted from the state-of-the-art, different methodologies for demonstrating robot motions exist. The most prominent are kinesthetic teaching and spatial-sensor demonstrations. However, to the author's knowledge, there needs to be a comparison between these two methods from a user perspective. Therefore, before studying how self-defined trajectories affect the task, there is a need for preliminary studies to select which impact each method has. This section reports the findings of this study and the implications of using a spatial sensor for demonstration.

### Accuracy Performances

In the case of kinesthetic demonstrations, the accuracy of the recorded points is high as long as the robot's internal data is used, which often leads to errors in the millimeter range. However, as highlighted in chapter 3, if a spatial sensor is used to demonstrate trajectories, the robot needs to execute motions considering coordinate transformations between the spatial sensor and the robot. The reader is invited to review Sec. 3.3.2 to understand how such transformations are applied. However, for clarity, it is essential to underline that such transformation is accomplished by sampling three points while the spatial sensor is mechanically linked to the robot. Then, having such information, equation Eq. 4.1 can be solved.

$$^{t}T_{r} = (P_i^r + M)\,(P_i^t)^{-1},\ \forall i\ (i = 1,2,3) \tag{4.1}$$

In this formula, $P_i^r$ represents the sampled point coordinates within the robot's coordinate frame, while $M$ denotes the constant transformation between the robot flange and the location where the pen is mounted for tracking. Additionally, $P_i^t$ refers to the sampled point coordinates in the coordinate system of the tracked pen. However, it is important to denote that such calibration is not adaptive and may not account for sensor drifts or inaccuracies. Therefore, the sensor's accuracy was analyzed to understand the impact of such calibration. Therefore, the sensor was again rigidly connected to the robot, and the robot was instructed to move across the whole workspace. During the movement, both the sensor's calibrated position and the robot's positions were recorded with respect to a common coordinate frame; in this way, the accuracy could be compared as the robot's location could be considered the ground truth. The results of this evaluation are shown in Fig. 4.4. More precisely the average distance on the x-axis was M $= 9.74 \times 10^{-4}$ m SD $= 2.8 \times 10^{-3}$ m. For the y-axis it was M $= -5.41 \times 10^{-4}$ m SD $= 2.2 \times 10^{-3}$ m. Third, for the z-axis it was M $= 1.44 \times 10^{-3}$ m SD $= 7.0 \times 10^{-3}$ m. Finally, for the angular it was M $= 4.60 \times 10^{-3}$ rad SD $= 2.5 \times 10^{-3}$ rad. Therefore, looking at these values, it is possible to understand that the average accuracy is fine with the application because it is in the millimetric to sub-millimetric range. However, the main issue looking at the data is the rather large standard deviation, which could lead to a sudden loss of such precision. A 3D plot showing the differences between the sensor and the ground truth is created, as shown in Fig. 4.5. Interestingly, the graph shows that the error increases further outwards when the sensor is moved from the calibration location where the points for Eq. 4.1 were sampled. Therefore, if a user is using the sensor, good performances can be achieved in the vicinity of the calibration; however, the farther the sensor is moved away from the calibration location, the more




**(a)** x-axis **(b)** y-axis **(c)** z-axis **(d)** Rotation

**Figure 4.4:** Distance error of the measured positions recorded with the spatial sensor versus ground truth measured through the robot location as long the spatial sensor is connected rigidly with the robot. From the figure, it is possible to understand that the average accuracy is good, but the standard deviation can lead to sudden changes in the sensor's accuracy.

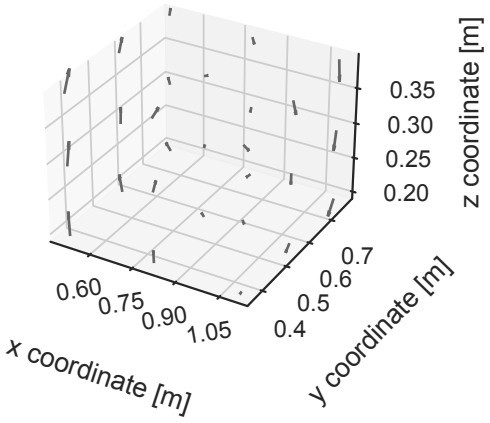



**Figure 4.5:** Visual representation of the differences between the ground truth and the measured location of the spatial sensor. The error is plotted with arrows, which, as starting points, have the ground truth data and, as the final point, the measured value of the spatial sensor. Interestingly, the error increases the farther the sensor is moved from the center. The center is the calibration location where points for Eq. 4.1 were sampled.

sudden changes can appear, leading to errors that could be in the centimetric range as the euclidean distance infinity norm is $\|x\|_\infty = 1.40 \times 10^{-2}$ m. Considering these results, if the spatial sensor needs to be used for teaching spatial trajectories, a method to account for this variability in its accuracy is mandatory to ensure precision in the whole workspace where the user could create trajectories.

### User Evaluations

Aside from the accuracy, the second set of experiments regarding the usage of the spatial sensor is to examine the impact of utilizing a spatial sensor in terms of usability. In this study, participants were required to demonstrate robot trajectories for a handover motion using two methods, namely kinesthetic demonstration and spatial sensor

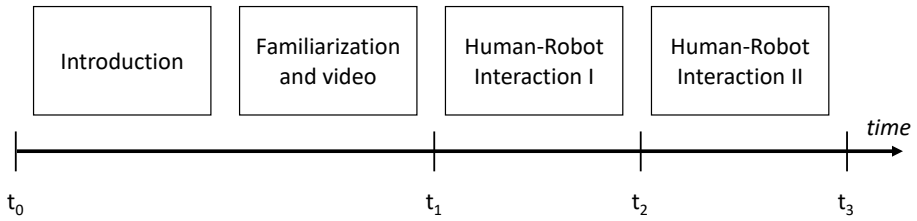
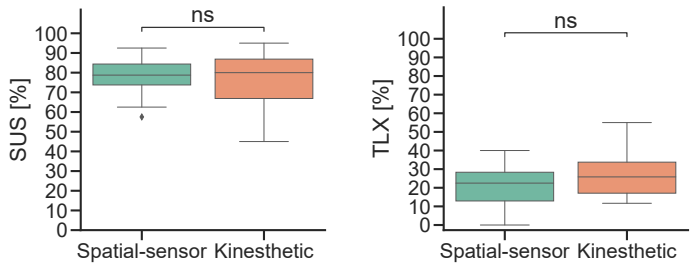



**Figure 4.6:** Experiment design schema. At first, the users were introduced to the task, and a video of the teaching methodologies was shown. This initial phase ended at $t_1$, where informed consent and general demographics were collected. Then, the two interactions unfolded, and at the end of each interaction, $t_2$ and $t_3$, the perceived levels of usability and workload were collected.




**(a)** Usability **(b)** Workload

**Figure 4.7:** Results of the user evaluation for the trajectory demonstration. To the left, usability is measured in percentage according to the SUS scale; to the right, workload is measured in percentage according to the NASA-TLX scale. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed.

demonstration. Therefore, after a brief introduction, they had to save the 3D positions along an imaginary trajectory indicating where the robot should move near the calibration point to avoid issues with the accuracy performances. Such demonstrations were performed with two subsequent interactions. At the end of each interaction, users had to rate the usability and workload of each method using the System Usability Scale (SUS) [109] and the NASA-TLX scale [110]. In the end, generated trajectories were also saved. For clarity, the experiment schema is shown in Fig. 4.6. For these studies, the Tracepen™ spatial tracked pen was used as a spatial sensor, and the default standard lead-through teaching of a UR10 robot was used as kinesthetic teaching. In the study, eight individuals participated, age M = 27.00 yrs, SD = 8.99, height M = 182.88 cm, SD = 11.61. All of them performed the task correctly, and no data was discarded. For evaluating usability and workload, the questionnaires' responses were compared. Regarding the former, results were M=75.63%, SD=17.31 and M=77.50%, SD=12.17 for kinesthetic and spatial-sensor teaching, respectively. Regarding the latter, outcomes were M=27.29%, SD=14.42 and M=21.67%, SD=13.45 for kinesthetic and spatial-

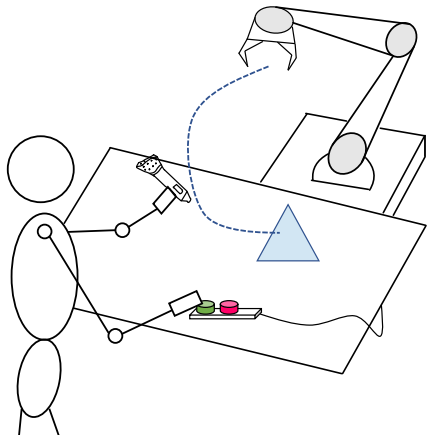


**Figure 4.8:** Illustration of the spatial-based interaction for recording points of a trajectory. The operator can specify where the robot should drive by moving the spatial sensor and confirming the recording or deletion of points via a teach pendant.

sensor teaching respectively. A Mann-Whitney U test was applied to check statistical significance as long the t-test preconditions did not hold and no statistical difference among the samples was measured. For clarity, Fig. 4.7 shows the outcomes of this evaluation. For such analysis, it is then possible to gather two primary outcomes. First, the force control algorithm of the UR10 robot does not have the same usability problems as the ones pointed out in the literature [29]. Second, the spatial-sensor demonstration is as usable as kinesthetic teaching and bears a low workload level. Considering this information, spatial-sensor teaching can record user-defined trajectories if the accuracy issues are resolved.

## 4.4 Definition of Trajectory Profiles Through a Spatial Interaction

As previously highlighted, more research is needed to understand how user-defined trajectories influence acceptance and robot safety. A method for allowing operators to program trajectories is proposed to study this topic further. This method is based on a spatial sensor as long as it was demonstrated to have a good usability level. However, the device data has to be corrected to reduce inaccuracies. Additionally, simulation-based evaluation is introduced to study the influence of trajectory teaching on safety. This chapter describes the various elements and their association with the suggested research.

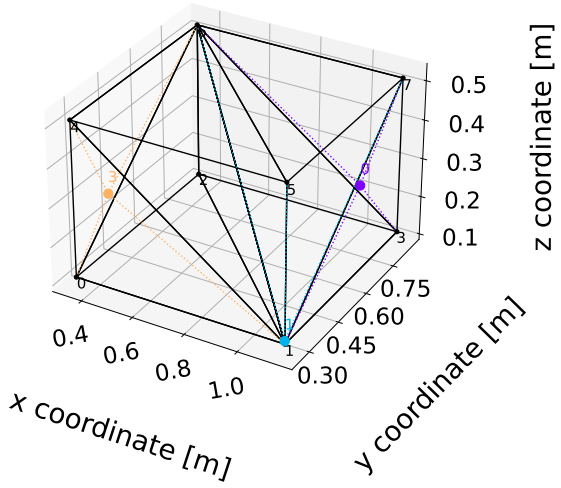



**Figure 4.9:** Clarified example of the barycentric coordinates using the Delaunay triangulation. The black numbered points are the center of the calibration locations where $U$ is calculated. The black lines are describing a tetrahedron in the cuboid grid. Finally, the colored points and lines show how the barycentric coordinates are used to calculate weights necessary for the translation and rotation interpolation.

### Envisioned Interaction

For allowing a human to record a trajectory, an interaction based on the movement of a spatial sensor is envisioned, as shown in Fig. 4.8. During this interaction, the user first places the objects involved in the interaction on the work table. Then, points where the robot EE should navigate, are recorded by moving the spatial sensor and clicking a button on an external teach pendant. If the user wants, points can be deleted using an additional button on the external teach pendant. In this way, the user can freely decide where the robot should navigate with minimum effort.

### Spatial Pose Recording

As previously identified, the spatial sensor's accuracy can be improved. However, to understand how this goal can be achieved, it is essential to underline the following results from the previous accuracy evaluation. First, the previously identified and described calibration method with Eq. 4.1 works well in a defined space. This means the spatial sensor is accurate in an area near the sampled points. Second, the accuracy of the spatial sensor seems to be proportionally decreasing with the distance from where the sensor was initially calibrated. Considering these findings, to improve the accuracy of the spatial sensor, it is possible to imagine that the initial calibration matrix ${}^{t}T_{r}$ needs to be corrected in different locations to account for the inaccuracies of the spatial sensor. In other words, if the spatial sensor is rigidly coupled with the robot flange while

performing the calibration, it is possible to define that an undistortion matrix $U$ must be applied to obtain the correct calibrated location as defined with Eq. 4.2.

$$^{t}T_{r} = (P_{i}^{r} + M)\,(P_{i}^{t})^{-1} * U^{-1},\ \forall i\ (i = 1,2,3) \tag{4.2}$$

However, if the $U$ is computed only once, then the undistortion matrix will mitigate the errors near the area where the points are sampled. Therefore, to ensure that the concept can be applied in the robot workspace, $U$ is calculated at different locations in a cuboid grid 3x3x3, resulting in 27 undistortion matrices. However, an interpolation-based method is introduced to ensure the correct selection of the suitable undistortion matrix in the workspace for a specific area. Therefore, the cuboid grid is divided into tetrahedrons using Delaunay triangulation [151] to allow for a barycentric coordinate system which can be used to calculate intermediate undistortion matrices upon distances to the vertices of the tetrahedrons. More precisely, the barycentric coordinate system allows the identification of the closest vertices in the tetrahedrons; therefore, such distances are then used to weight the interpolation between the different $U$ where the translation part of $U$ is interpolated linearly, and the rotational part is interpolated using linear interpolation (LERP) using its quaternion representation. For clarity, Fig. 4.9 shows how such distances are calculated for two sample points. In this way, intermediate $U$ can be calculated in the workspace to reduce errors of the spatial sensor. Therefore, following the idea of the envisioned interaction, the user will be able to move the sensor through the desired trajectory while holding down the recording button, and in this way, robot trajectories can be recorded for the latter execution.

#### Calculation of Safety KPIs

As mentioned in the introduction, safety KPIs can be calculated based on the trajectory used by the robot. In SSM, the most important KPIs are the distance between the robot and the human. To assess such safety parameters, the robot trajectories for both collaborations were recorded and subsequently simulated using the 3D-simulation tool Visual Components (VC) with a Computer-aided-safety (CAS) plugin [141]. This plugin computes and visualizes the separation distance following the ISO/TS 15066 formula. Among other factors, it considers the robot's joint states, velocities, and braking characteristics to ascertain the swept volume of a sphere when the robot halts suddenly at each point in time. The requirement for safety is that this volume must not intersect the space occupied by the human operator. To simplify matters, the 2D projections of safety volumes are aggregated onto the working area for the entire trajectory. To evaluate safety performance, we compare the sizes of these areas and the distance between

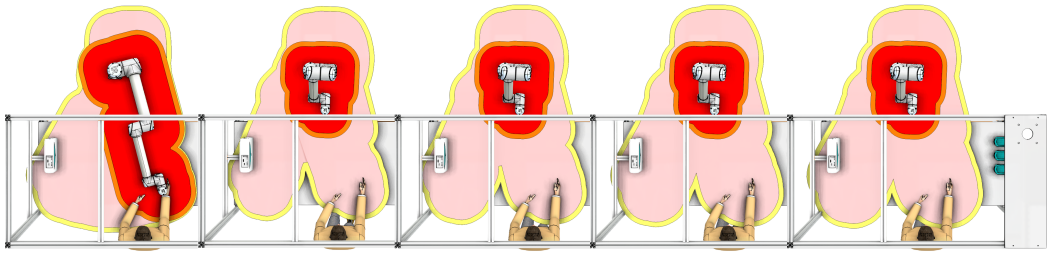


**Figure 4.10:** Qualitative representation of the safety area calculation in the CAS tool. The robot and human endpoints are used to calculate the area considering the speed and the robot extension. Afterward, the human end position calculates the distances between humans and robots.

the safety area and the human operator. A smaller safety area and a greater distance from the human operator help avert situations where the robot must stop to avoid collisions, enhancing the system's performance. One caveat of this approach is that the control algorithms utilized in the simulation differ from those of the actual control unit. Therefore, instead of reprogramming the authentic robot program to execute the simulated motions, the natural motions were duplicated in the simulation by employing the discrete robot joint angles recorded during the experiments at a sampling rate of 20 Hz. A qualitative figure shows this plugin's safety area in Fig. 4.10.

## 4.5 Evaluation of Safety Levels on User-Defined Trajectories

Having now a method to record precise robot trajectories via a spatial sensor, the next step is to understand which effects such a method has. To perform this evaluation, a user study is conceived. However, considering the previous results on the effect of expertise on teaching outcomes as defined in Chapter 2, a similar experiment design is implemented. This section reports the applied methods for this investigation.

### 4.5.1 Industrial Use Case

In manufacturing a gearbox for a healthcare product, see Fig. 4.11, variability is high in the preparation of the kitting box due to different materials employed in the gearbox. Therefore, the tasks involved in the process could change often depending on the bill of materials (BOM). Unfortunately, a fully autonomous solution is not economically viable. However, a method to reduce the mental effort of an operator could help reduce errors. More precisely, such a gearbox has the highest variability in the type and configuration of the base plate. Therefore, users are constantly pressured to select the

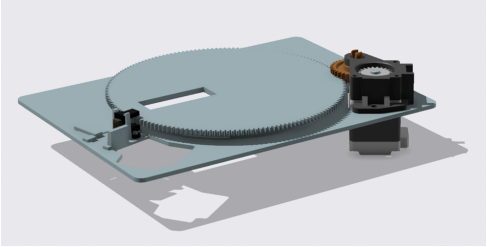


**Figure 4.11:** Rendering of the gearbox assembly.The assembly is composed of numerous gears and components affixed to a base plate. To assemble all the parts, several manual steps must be performed. Reprinted from Augmented Reality for Supporting Workers in Human–Robot Collaboration, Moya A. et al. [152], Copyright (2023).

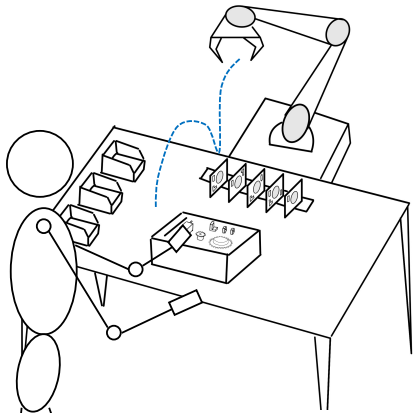


**Figure 4.12:** Representation of the industrial use case. The user assembles the kitting box with components stored on one side of the worktable. Meanwhile, the robot picks one of the base plates according to the production order and delivers it to the human operator with a particular trajectory.

correct base plate for the production order. Depending on the manufacturing schedule, an autonomous robot can pick and deliver the correct base plate configuration to simplify this procedure. However, to account for different operators of the robot's trajectory for the base plate delivery, it is suggested that the operators themselves devise the trajectory. In this way, they can better personalize the task to their needs. For clarity, Fig. 4.12 shows the main steps in the delivery trajectory teaching and the kitting box preparation.

### 4.5.2 Experiment Design

A within-subjects design with a randomized interaction sequence is conceived to analyze the influence given while allowing operators to devise their trajectory as shown from Fig. 4.13. In this design, the operators perform the interaction described in the industrial use case with two robot control methodologies. On the one hand, one interaction is through the proposed trajectory teaching through the spatial sensor; the other

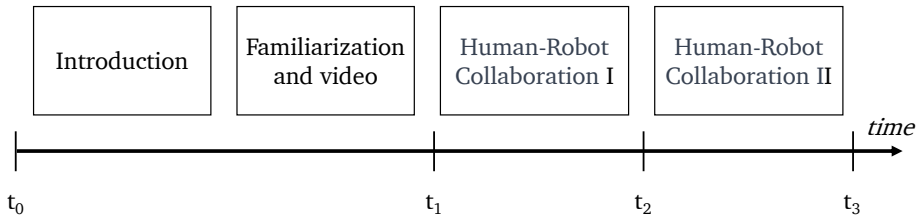



**Figure 4.13:** Experiment design schema to evaluate the impact of collision-free and self-defined trajectories. The users were exposed to two interactions after an initial introduction and familiarization with the two systems. At the end of each interaction, they had to express their level of trajectory acceptance according to the questionnaire of [126].

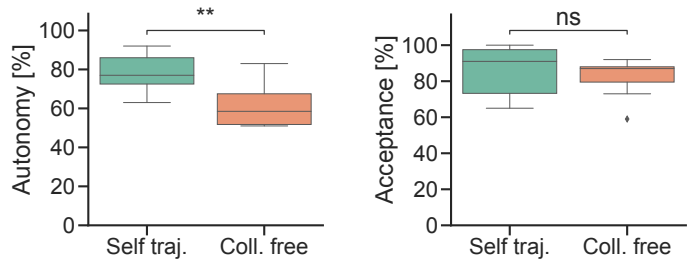
interaction is through a collision avoidance algorithm, which selects a collision-free trajectory for delivering the component. The selected collision avoidance algorithm is an implementation of previous work [153] to use a similar experiment design as [126]. This method relies on 3D cameras combined with the GPU-Voxels library to obtain a voxel map with 0.02 m resolution for a detailed representation of workspace occupancy. The robot collision model removes the manipulator from the voxel map and computes the swept volume of planned trajectories. This allows real-time replanning for collision-free trajectories during the interaction.

Therefore, using the nomenclature of Fig. 4.13, the users were initially given an introduction to the system and had time to familiarize themselves with it by looking at some videos between time $t_0$ and $t_1$. Then, the two interactions unfolded, and at the end of each interaction, namely at times $t_2$ and $t_3$, they had to express their level of trajectory acceptance according to the questionnaire of [126] and their level of task decision autonomy according to the WDQ [75]. Additionally, during the interactions, the executed robot trajectories and the position of the user's hands were monitored and saved for later analysis. Through this experiment, therefore, the following hypotheses are tested. One concerns the level of acceptance, and the other regards the level of safety. More specifically, the hypothesis can be stated as follows:

- *Hypothesis 1 (H1)* Users who are able to describe their trajectories feel a higher level of acceptance

- *Hypothesis 2 (H2)* Higher level of acceptance leads to safer interactions

This section presents the experiments conducted to find answers to these questions.




**(a)** Autonomy **(b)** Acceptance

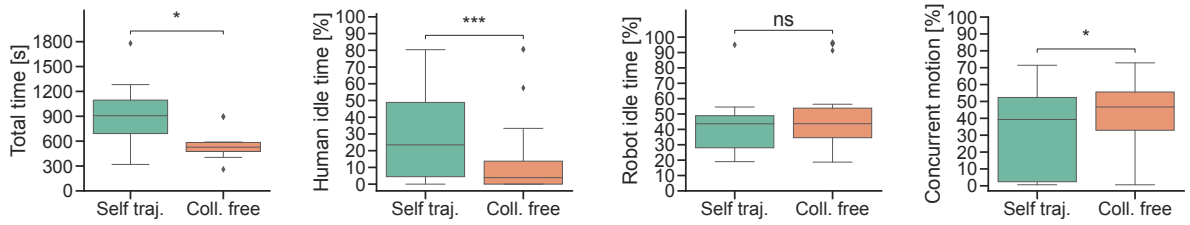
**Figure 4.14:** Results of the user evaluation for the trajectory demonstration. To the left, the level of task decision autonomy was measured with the WDQ, and to the right, the level of trajectory acceptance was measured with the questionnaire of [126]. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed.

### 4.5.3 Results

Having defined a method for users to create their trajectories using a spatial device with corrected accuracy, a user study designed as mentioned above was conducted. The user study aimed to identify the effects of user-defined trajectories on the users compared to collision avoidance trajectories. More specifically, the acceptability of the trajectory and its relative safety level are calculated through the CAS tool.

In the study, 12 users participated; however, two did not perform the test correctly, so results from 10 users were analyzed M = 27.00 yrs, SD = 8.99, height M = 182.88 cm, SD = 11.61. This group was subdivided between robotic experts and novices, considering the results of the previous chapter. The first set of results regarded perceived task decision autonomy and acceptance. These are shown in figure Fig. 4.14. As shown from the figures, the level of autonomy is improved when operators can define their trajectory. However, their level of acceptance is not. More specifically, the recorded measures for the autonomy were M = 61.60%, SD = 11.40, and M = 78.20%, SD = 9.22 for the collision-free and self-defined trajectories, respectively. On the other hand, the recorded levels of acceptance were M = 82.40%, SD = 10.03, M = 85.90%, and SD = 13.95 for collision-free and self-defined trajectories, respectively. A Mann-Whitney U test was applied to check statistical significance as long the t-test preconditions did not hold, and the statistical difference was found just for the level of perceived task decision autonomy.

The second set of measurements regarded the time taken during the task; the results are presented in Fig. 4.15. According to the measurements made during the task, the total time necessary to execute the task drastically reduced between the self-defined trajectory and the collision-free motion. More specifically, the recorded timings were

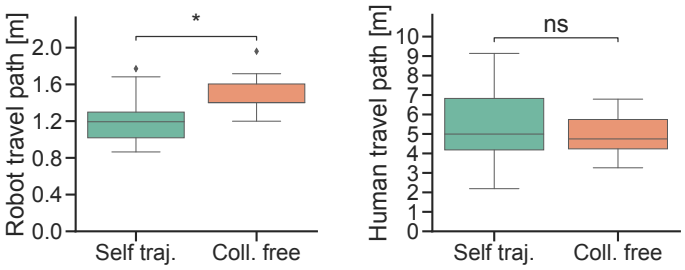



**(a)** Total time **(b)** Human idle time **(c)** Robot idle time **(d)** Concurrent motion

**Figure 4.15:** Time evaluations of the tasks. From left to right, the total time, the percentage of human idle time, the percentage of robot idle time, and finally, the percentage of concurrent robot and human motions are shown. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed.

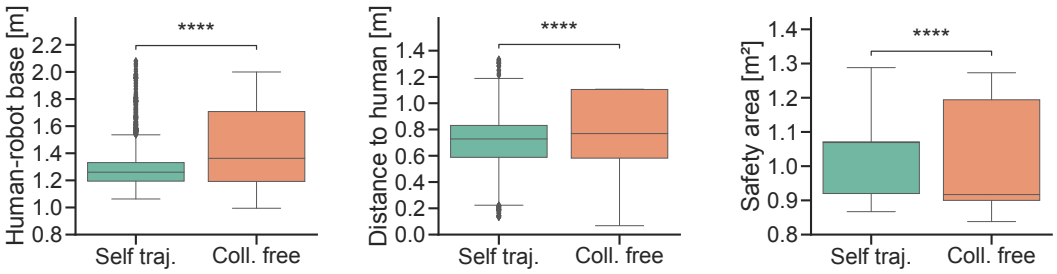
M = 563.00 s, SD = 198.00, M = 941.10 s, and SD = 400.60 for the collision-free and self-defined trajectories. A Mann-Whitney U test was applied to check statistical significance as long the t-test preconditions did not hold and a statistical difference was found. Additionally, to the total time, the percentage of the total time falling into three categories was measured as suggested by [126] to understand which percentages of the time were used either for performing concurrent robot-human movements or for idle times either for the robot or the human. Interestingly, the percentage of human idle time increased in the self-defined trajectory. Thus, the robot's idle time and the concurrent motion decreased compared to the collision-free. More specifically, the recorded values for the three measurements were as follows. Concerning the human idle time, the recorded values were M = 11.25%, SD = 19.12, M = 29.03%, and SD = 26.01 for the collision-free and the self-defined trajectory, respectively. Concerning the robot idle time, the recorded values were M = 47.63%, SD = 23.78, and M = 40.46%, SD = 14.66 for the collision-free and the self-defined trajectory, respectively. Finally, concerning the percentages of concurrent motions, the recorded values were as follows: M = 41.12%, SD = 22.69, M = 30.50%, and SD = 23.43 for the collision-free and the self-defined trajectory, respectively. A Mann-Whitney U test was applied to check statistical significance as long the t-test preconditions did not hold, and a statistical difference was found for the human idle time and the concurrent motion. Therefore, it is possible to understand that in the case of self-defined trajectories, operators waited for longer for the robot to execute motions, and they tended to move whenever the robot was not moving.

The third set of measurements concerned the travel paths of the human operators and the robot; the results are displayed in Fig. 4.16. Regarding the robot travel paths, the values were M = 1.50 m, SD = 0.22, M = 1.23 m, and SD = 0.30 for the collision-free and the self-defined trajectory, respectively. Regarding the human travel paths, the




**(a)** Robot travel path **(b)** Human travel path

**Figure 4.16:** Results of the travel paths in the two scenarios. To the left comparison of robot travel paths. To the right comparisons of the human travel paths. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed.

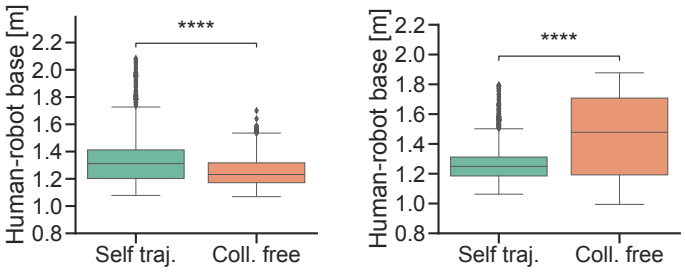



**(a)** Distance human robot base **(b)** Distance human to EE **(c)** Safety area

**Figure 4.17:** Recorded levels of distances and safety areas for the two scenarios. To the left is the distance between the human operator trunk and the robot base. The distance between the human operator's hands and the robot EE is in the middle. To the right is the recorded safety area. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed.

values were M = 4.95 m, SD = 1.10, M = 5.35 m, and SD = 2.30 for the collision-free and the self-defined trajectory, respectively. A Mann-Whitney U test was applied to check statistical significance as long the t-test preconditions did not hold, and a statistical difference was found for the robot travel path but not for the human travel path. Therefore, it is possible to understand that operators, when they could program robot motions, preferred program motions that were shorter compared to the collision-free ones.

The last set of evaluations regarded the distance between the human operators and the robot and the level of safety measured in the safety area. The results for both experts and non-experts are shown in Fig. 4.17. The recorded values regarding the distance between the human trunk and the robot base were as follows. The collision-free values were M = 1.42 m, SD = 0.24. For the self-defined trajectory, the values were M = 1.29 m, SD = 0.15. The recorded values regarding the distance between the human hands and the robot EE were as follows. The collision-free values were M



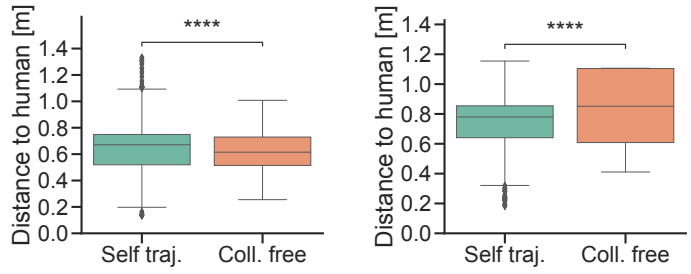

**(a)** Experts **(b)** Novices

**Figure 4.18:** Results of the human operator's trunk distances to the robot base. To the left are results for the experts, and to the right are results for the novices. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed.

= 0.81 m, SD = 0.24. For the self-defined trajectory, the values were M = 0.71 m, SD = 0.18. Regarding the safety area, the recorded areas were as follows. For the collision-free, the values were M = 1.01 $m^2$, SD = 0.13. For the self-defined trajectory, the values were M = 1.02 $m^2$, SD = 0.10. A Mann-Whitney U test was applied to check statistical significance as long the t-test preconditions did not hold, and a statistical difference was found for all three measures. Therefore, it is possible to understand that operators, when able to define the robot trajectories, tended to stay closer to the robot base. A similar result can be seen also in the distance between the human and robot EE. Therefore, the level of safety decreased as long as the average recorded safety area was larger than the one in the collision-free. However, considering the results of Ch. 2, it is crucial to understand how different levels of expertise might have influenced these factors.

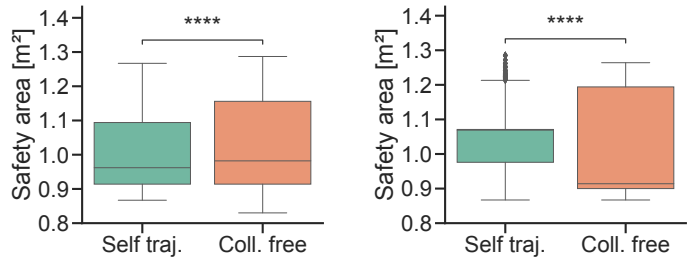
Concerning the distances between the human operator trunk and the robot base, the results are shown in Fig. 4.18. From these graphs, it is possible to see that the possibility of creating self-trajectories had a different impact on the distance from the robot base upon expertise level. The average values were M = 1.27 m, SD = 0.13, and M = 1.44 m, SD = 0.24 for the self-trajectories and collision-free in novices. Differently, the experts showed an opposite trend. The recorded values were as follows for the self-trajectories and collision-free M = 1.34 m, SD = 0.19, and M = 1.26 m, SD = 0.13. A Mann-Whitney U test was applied to check statistical significance as long the t-test preconditions did not hold, and a statistical difference was found for both groups. Therefore, it is possible to conclude that novices kept closer to the robot base than experts whenever they could program their trajectory. On the contrary, experts stayed closer to the robot base whenever the robot was performing collision-free motions.

Connected to these results is also the distance between the human operators' hands




**(a)** Experts **(b)** Novices

**Figure 4.19:** Results of the distances of human operators' hands to the robot EE. To the left are results for the experts, and to the right are results for the novices. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed.




**(a)** Experts **(b)** Novices

**Figure 4.20:** Results of safety area in the collaborations divided by group. To the left are results for the experts, and to the right are results for the novices. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed.

and the robot EE; the results are shown in Fig. 4.19. As expected, the results are similar to the one above. The recorded values for the novices were as follows: M = 0.85 m, SD = 0.23, and M = 0.75 m, SD = 0.17, respectively, for the collision-free and the self-defined trajectories. The recorded values for the experts were as follows: M = 0.62 m, SD = 0.15, M = 0.66 m, and SD = 0.19, respectively, for the collision-free and the self-defined trajectories. A Mann-Whitney U test was applied to check statistical significance as long the t-test preconditions did not hold, and a statistical difference was found for both groups. Therefore, experts stayed closer to the robot when performing collision-free motions, like in the previous findings. In contrast, novices tended to stay closer when the robot performed self-defined trajectories.

This experiment's last set of results is the recorded safety area; the results divided upon expertise group are shown in Fig. 4.20. Similarly to the results above, the different groups influenced the safety area. For the experts, the recorded values were M = 1.02 $m^2$, SD = 0.13, and M = 0.99 $m^2$, SD = 0.11, respectively, for the collision-free and the self-defined trajectories. For the novices, the results for the collisions-free

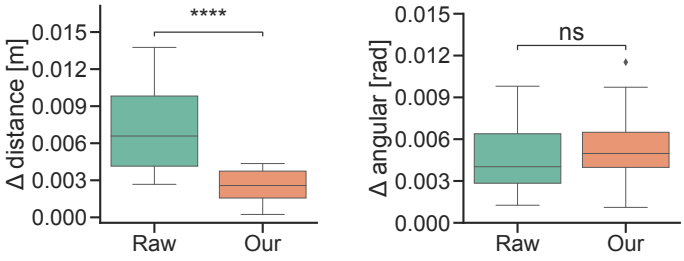
and self-defined trajectories were M = 1.02 $m^2$, SD = 0.14, and M = 1.04 $m^2$, SD = 0.09, respectively. Therefore, it is possible to understand that experts could define trajectories with smaller safety areas than novices.

#### Discussions

This study highlighted how allowing operators to define their trajectories can influence the level of collaboration between operators and robots. To understand the changes of such collaboration, this study tried to analyze and compare the differences with a state-of-the-art collision avoidance algorithm.

Firstly, similar to the findings in Ch. 2, allowing users to define their trajectories allows them to perceive a higher level of task decision autonomy when compared to collision-free ones. Unfortunately, this level of higher decision autonomy did not influence the trajectory acceptance as previously found by [126]. Thus, the level of trajectory acceptance was found to be high for the self-defined and collision-free trajectories. Thus partially confirming *H1*. Interestingly, this level of decision autonomy changed the type of collaboration. In terms of total time, users needed more time to complete the task during the self-defined trajectories. A more detailed analysis revealed that humans spent more time idle in such cases and reduced the levels of concurrent motion with the robot. This result is interesting considering that, on average, users programmed trajectories shorter than the collision-free ones as shown from the robot travel path. Therefore, in the case of self-defined trajectories, users waited more for the robot to perform actions.

However, an exciting result is when outcomes of behaviors in the two types of trajectories upon expertise are analyzed. In the results, it is possible to see that experts and novices programmed trajectories differently. In the case of experts, the trajectories were programmed to result in a higher distance between the robot EE and the human hands when compared with collision-free ones, probably due to the more considerable distance to the robot base. In the case of novices, an opposite trend can be observed. Trajectories in the self-defined case were programmed to result in a closer distance to the human hands than the collision-free ones, also probably due to the closer distance of the human to the robot base. Therefore, considering the findings of [79, 80] and Ch. 2, it is possible to infer that experts tried to optimize for joint task efficiency in case of collision-free motions as long they moved closer to the robot in this kind of collaboration. However, novices showed an opposite trend trying to optimize the joint task efficiency in the case of self-defined trajectories. Due to these different optimization levels, direct influences on the required safety area are thus observed. Experts reduced




**(a)** Translation accuracy **(b)** Angular accuracy

**Figure 4.21:** Results of the accuracy of the spatial sensor with the proposed method. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed.

the safety area in the collision-free movement, while novices reduced their safety area in the self-defined trajectories.

Aside from the in-group differences, it is also interesting to see the comparison between the different expertise groups. The analysis shows that experts ended up in situations where the safety area was, on average, smaller than novices. However, they also kept a closer distance to the robot when compared to novices. Therefore, despite both groups being in safe collaborations, experts tended to program the robot in a way that the safety braking parameters of individual joints were in a more optimal area of operation [154] compared to novices, leading to slightly smaller safety areas despite their closeness to the robot.

Considering these results, therefore, when allowing users to personalize the trajectories, their background must be considered to provide the best workspace that fits their requirements. For example, experts could benefit from a smaller workspace to reduce the travel paths of the users. However, further investigations with a larger user sample size might be needed to understand how this optimization could change with more repetitions, as it could be expected that both groups might evolve their interaction strategies.

Aside from the results obtained from the user study, to ensure that the study was correctly designed, a further analysis of the accuracy of the spatial sensor with the proposed correction algorithm was performed. The results of this study are shown in Fig. 4.21. The analysis shows that the proposed method improves the translation accuracy while keeping the rotation accuracy unchanged but still at a good level. More precisely, for the translation, the proposed method reports M = 2.5 $\times 10^{-3}$ m, SD = 1.3 $\times 10^{-3}$ whereas the raw data from the spatial sensor with a single calibration point (i.e., raw) reports M = 7.2 $\times 10^{-3}$ m, SD = 3.4 $\times 10^{-3}$. Additionally the euclidean distance infinity norms are $\|x\|_\infty = 1.40 \times 10^{-2}$ m and $\|x\|_\infty = 4.3 \times 10^{-3}$ m for the raw and

**Table 4.1:** Average deviation of simulated robot properties to actual measured values, the table reports the average change in degree and its standard deviation. The deviation of joint angles is on the line above. On the bottom line is the deviation of joint speeds.

| Joint | J3 | J2 | J1 | J4 | J5 | J6 |
|---|---|---|---|---|---|---|
| deg $\times 10^{-3}$ | 49 ± 7 | -34 ± 4 | 35 ± 3 | 119 ± 4 | -46 ± 3 | -9 ± 3 |
| deg/s $\times 10^{-3}$ | -3 ± 2 | -1 ± 1 | 0 ± 1 | 0 ± 1 | 1 ± 1 | -1 ± 1 |

the proposed method respectively. A t-test reported statistical significance. Therefore, it is possible to conclude that the proposed method can reduce the translation error of the spatial device. In the case of angular accuracy, an improvement could not be measured, and the two populations should be considered similar. However, the error is relatively small. Therefore, no major influences were recorded during the study as long the raw method reported M = 4.6 $\times 10^{-3}$ rad, SD = 2.5 $\times 10^{-3}$ and the proposed method reported M = 5.5 $\times 10^{-3}$ rad, SD = 2.3 $\times 10^{-3}$.

An additional evaluation was done on the CAS simulator by comparing the simulated joint angles with the real ones in various motions. The results are reported in 4.1. As the table shows, the average deviation of the simulated values compared to the actual measured values is very insignificant. For the angle comparison, the maximum average difference is 0.119°. This results in a maximum simulation difference of 0.79%. Therefore, similar to the evaluation of the spatial sensor calibration, the simulator behaved as expected. Therefore, no significant limitations to the study besides the small user sample size are identified.

## 4.6 Conclusions

### Summary

Through this chapter, this work wanted to discover further impacts given by intuitive programming interfaces. More specifically, this section tackled *Challenge 2* (intuitive programming interfaces that simplify safety assessments) by proposing and studying the impacts of a method for defining robotic trajectories via a spatial interaction.

To study these impacts, the chapter started with an overview of the state of the art and identified the main challenges in the field. During this literature review, it was discovered that few examples exist regarding how operator-defined trajectories could influence human-robot collaboration, especially regarding safety. Therefore, the chapter proceeded with an overview of some preliminaries allowing users to define their safe trajectories. Initially, a new EE design was proposed. Such design was necessary

as the literature did not have any example of an EE that could be adapted to different grasping tasks while satisfying requirements for evaluating safe trajectories like robot operating speeds. Therefore, with this design, safe robot operating speeds for a reconfigurable EE were identified. Second, the precision of the selected spatial sensor was measured to identify if the sensor could be used for recording trajectories. Potential inaccuracies were observed during this step when the spatial sensor was moved far away from the calibration point as errors up to $1.40 \times 10^{-2}$ m were measured. Third, the level of usability of trajectory recording via the spatial sensor was measured, which yielded a usability score of 77.5% when measured with the SUS. Next, with this data from the preliminaries, a method for defining trajectories based on an algorithm to improve the accuracy of the spatial sensor was presented alongside a study to identify the relations between robotic expertise and safety levels.

The method proposed for recording trajectories was more accurate, with spatial sensor errors decreasing to a maximum value of $4.3 \times 10^{-3}$ m. Additionally, using the proposed method exhibited high decision autonomy from the users, averaging 78.20% when measured with WDQ. Then, when comparing robotics experts and novices, it was observed that novices tended to create trajectories having a higher distance between humans and the robot EE, with an average distance of 0.75 m. In contrast, experts created trajectories with an average distance of 0.66 m. Despite these differences, novices maintained a closer distance from the robot base than experts during the execution of such trajectories, averaging a distance of 1.27 m for novices and 1.34 m for experts. This behavior directly influenced safety in the case of self-defined trajectories, as the safety area was smaller for experts, with an average value of 0.99 $m^2$ than for novices, with an average value of 1.02 $m^2$.

#### Discussions and Future Work

In this study, the effects of allowing users to define their trajectory on the safety level of the application were presented. According to the conducted studies, allowing users to define their trajectories can be done, and the method of the spatial sensor provided is a promising approach that could lead to a high level of usability. However, the level of safety can be strongly impacted by how the trajectory is defined. First, if a robot can move freely without a safety cage, safety tests on the hazardous robot areas must be done as clarified with the robot EE. Second, the robot's trajectories must be verified with additional software like Computer Aided Safety to ensure that the selected safety sensor can stop the robot in time. In this latter point, different levels of robot expertise require different types of collaboration to ensure safe interactions. The study with ten

individuals identified that robot experts define trajectories with smaller safety areas than novices while being closer to the robot. Nevertheless, the in-group comparison identifies that experts optimized the joint-task efficiency in the collision-free scenario. In contrast, novices optimized the joint-task efficiency in the self-defined trajectories scenario.

However, despite these differences, the methods presented here allow SMEs to allow users to create their trajectories and conduct safety evaluations with an easy-to-use method based on spatial interaction, thus providing an intuitive programming method that embeds safety evaluations and answers positively to the aforementioned challenge. Nevertheless, future work should consider how to design the collaborative workspace based on user background due to different trajectory requirements to ensure high levels of acceptability while guaranteeing safe collaboration. Additionally, future work could guide users on which areas are best suited for programming safe trajectories to guarantee users a high level of autonomy while not jeopardizing their safety and simplifying the safety analysis.

Chapter 5

# Scalability of User-Defined Robotic Behaviors To Industrial Scenarios

Contents



To ensure the adoption of the proposed methods for successfully addressing the needs of SMEs, however, all the components in the previous chapters must fit well together, considering the common practices and standards in the industry. Therefore, this section presents how user-defined cobot behaviors can be integrated into manufacturing SMEs processes. This section describes how a simple modeling language can be used for robot programming and safety risk assessments. The content of this chapter is partially based on the following publications:

- Pantano, M., Eiband, T., and Lee, D., "Capability-based Frameworks for Industrial Robot Skills: a Survey," 2022, 18th International Conference on Automation Science and Engineering (CASE).

- Pantano, M., Pavlovskyi, Y., Schulenburg, E., Traganos, K., Ahmadi, S., Regulin, D., Lee, D., and Saenz, J., "Novel Approach using Risk Analysis Component to Continuously Update Collaborative Robotics Applications in the Smart, Connected Factory Model," 2022, Applied Sciences, Volume 12.

To present the content of this chapter, the structure is as follows. Initially, a detailed literature review regarding a common terminology for describing robotic capabilities is presented. Next, a method of how this terminology can be used for simplifying the risk analysis together with a common Business Process Model Notation (BPMN) is outlined. Finally, the results of a user test to evaluate the acceptability of this method based on standard terminology for robot programming are outlined.

## 5.1 Definition of a Common Terminology

For integrating user-defined cobot behaviors, standardization is needed on how different user-defined robot behaviors can be interpreted. This need is driven by the fact that SMEs might need to change robot programs often due to heterogeneous product variants. Therefore, it is essential to understand how user-defined behaviors can be interpreted and executed by the control algorithms of several robot brands. In other words, considering some fundamental concepts of Industry 4.0 (I4.0), it is necessary to define generalized robot capabilities that can be employed to execute user-defined behaviors. Capabilities are defined by I4.0 as "implementation-independent specification of a manufacturing function to accomplish a virtual or physical effect" [155]. Therefore, if users are defining robot behaviors, for their execution, the robot must have capabilities that fulfill the user behavior definition. In other words, if the user shows that the robot should reach one position, the robot must be able to move to a defined position, which should be robot control-independent. However, despite the concept being pretty trivial, the literature on robotics capabilities is diverse, and several definitions and nomenclatures have been proposed [156]. Therefore, recent literature needs to be reviewed to understand how robot capabilities are defined [157].

### 5.1.1 State-of-the-art Survey

As mentioned in the introduction, the literature regarding robot capabilities is various. Therefore, to shed light on the topic, a structured literature review (SLR) [158] is conducted to assess the main topics of capabilities and robots. The main goals of this

review are to find the most common nomenclatures used in robotics for describing capabilities, which are the most requested for industrial scenarios, and how those are implemented. SLR is selected as long it is a reliable approach to summarize a specific research field that can be well documented, and it is known to reduce the impact of researchers' bias in the review [159].

#### Background, Nomenclature and Research Method

A structured method with well-defined boundaries and data collection procedures must be set to implement an SLR. Therefore, as long as this research wants to find the most common nomenclature used, reviewing a few concepts related to capability engineering and how those can influence the robotics field is crucial. As initially mentioned, capability-based engineering envisions the production of goods by matching the required capability to manufacture a specific good with machines able to satisfy the production needs. To allow this linking between product manufacturing requirements and available machines, the concept of Product Process and Resources (PPR) was thus introduced [160]. PPR is a modeling approach that defines relations between products, the necessary resources (e.g., machines), and the process value chain. Therefore, it has been widely adopted by several Product Lifecycle Management (PLM) systems but not by robotics [156] as long the research community preferred the usage of other definitions as the IEEE 1872 Cora Ontology [161, 162]. Thus, a standard nomenclature that fits PPR and robotics must be set to allow an SLR. Unfortunately, no proposal is available in the literature; hence, one based on both terms from PPR and the robotics Factories of the Future (FoF) ontology [163] is proposed. Similar to PPR, the nomenclature has a hierarchical structure representing the necessary entities for production steps. However, additions to integrate robotics capabilities are done. A representation of this hierarchical nomenclature is shown in Fig. 5.1. The structure, using a top-down representation, can be explained as follows. First *Task*, this entity represents a complex manufacturing step that is part of a manufactured good value chain. Therefore, the good increases its value once a *Task* is performed. An example of a *Task* is the assembly of a gearbox. Thus, referring to PPR, a *Task* is intended as Process. Second *Parameterized skill*, this entity represents a detailed step within a *Task* for achieving a specific goal needed along the value chain creation. The information in this entity is used by one or more *Resource* (e.g., machine) for actuating actions needed to fulfill a goal. Thus, *Parameterized skill* can be seen as implementation-dependent *Skill*. With this consideration, then, *Skill* is an implementation-independent specification of a manufacturing function, and it is an aggregation of simple functions denoted as *Primitive* with specific

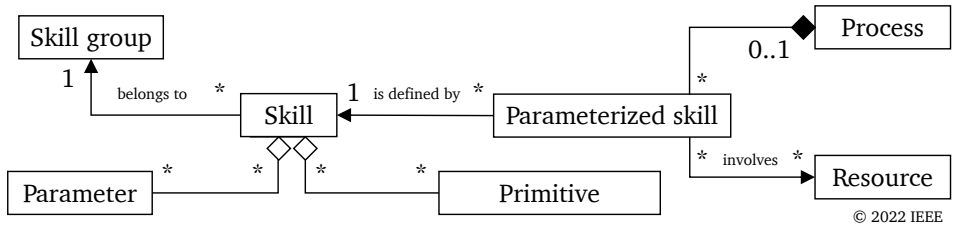



**Figure 5.1:** Schema of the hierarchical nomenclature describing capability-based frameworks in robotics. The scheme is obtained by merging nomenclature from Product Process Resource (PPR) [160] and robotics FoF ontology [163].

*Parameter*. To conclude, *Skill* are part of a *Skill group*, allowing easier search within different manufacturing functions. For clarity, an example of how to use this nomenclature is reported here. The example is also shown in Fig. 5.2. A transmission gearbox needs to be automatically assembled. Therefore, an operator creates a new *Process* and assigns different robotic *Skills* for its assembly by selecting them from several *Skill group*. Then, the operator populates Parameters via some user-defined behaviors, and *Parameterized skill* for a specific robot is automatically created considering the robot selected as *Resource*. Finally, the robot executes the *Task* by using *Primitive* and the assigned parameters. Once the research questions and the nomenclature used for the literature classification are defined, the next step in the SLR is to define the inclusion and exclusion criteria. Regarding inclusion, the research considered robotics applications, defined as using robotic skills to allow the robot to face altering environments and connect to industrial scenarios. The paper had to be listed in the Scopus database. Other databases like Web of Science and ISI Web of Knowledge were discarded as long they did not provide additional results. Regarding exclusions, any research discussing the human skills necessary to program a robot was discarded as long as this review wants to highlight how capability-based engineering is integrated into robotics. Finally, the classification of the research results was defined. As initially mentioned by the research questions, this review aims to assess the most common nomenclatures for robot capabilities, which are the most studied, and how they are implemented. To answer these questions, the terminology previously defined was used to organize the nomenclature used by the literature identified with the inclusion and exclusion criteria and the implementation framework.

### 5.1.2 Results and Discussions

Having defined the research method, the Scopus database was queried on 6 October 2021, and the database obtained 210 results for a time frame between 2014 and the query date. This time frame was selected because, with a larger time frame, no addi-

tional relevant literature was identified. Then, the papers were analyzed by reading the title and abstract, and according to the exclusion criteria, 149 works were discarded. The remaining 61 papers were thoroughly analyzed and classified according to the research method. An additional 27 papers were included in the analysis, as they contained references to the preliminary work of some of the authors whose papers were already included in the set. Through the classification, it was identified that the terminologies related to regarding *Skill*, *Process*, and *Primitive* as shown from Fig. 5.3 are the most used. More precisely, the literature showed that 79 out of 88 employ concepts similar to *Skill*, 65 out of 88 use concepts similar to *Process*, and 49 out of 88 use concepts similar to *Primitive*. However, the naming given to these three classes varies across researchers. In particular, researchers often call *Skill* as Skill (39 out of 79 works), *Process* as a task (35 out of 65) but rarely *Primitive* as primitive (12 out of 49 works). Therefore, to ensure a common understanding of PPR in the robotics domain, the word task should describe a *Process*, and the usage of the word primitive should be well explained. Regarding all the other classes, they were found less relevant, with *Skill group* used only by 14 out of 88, *Parameterized skill* 17 out of 88, and Parameter 39 out of 88 works. Additionally, the classification yielded that the most common *Process* type is assembly (4 out of 49), and the most common *Skill* types are pick (10 out of 79) and place (7 out of 79). The most frequent *Primitive* type is open gripper (10 out of 49) as shown in Fig. 5.4. Regarding implementation, the reviewed literature reported various frameworks. However, of the works that reported on implementation (39 out of 88), the most common one was the Robot Operating System (ROS) [165] (9 out of 39).

Therefore, considering that the research specifically targeted manufacturing use

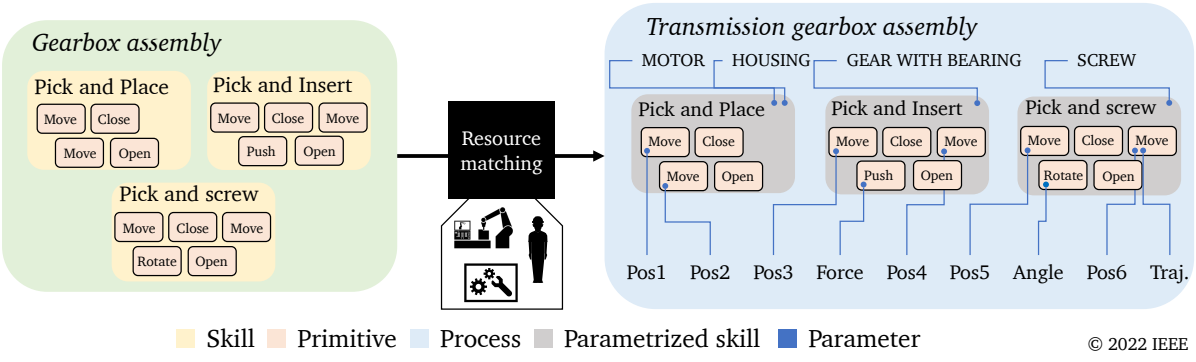



**Figure 5.2:** Usage example of the hierarchical framework for *Process* transmission gearbox assembly. The *Process* is an ensemble of several *Parameterized skill* containing *Primitive*. Parameterized skill are generated by general *Skill*, which have specific parameters for determined resources (i.e., robot). Reprinted from Capability-based Frameworks for Industrial Robot Skills: a Survey, International Conference on Automation Science and Engineering, Pantano M. et al. [164], Copyright (2022), with permission from IEEE.

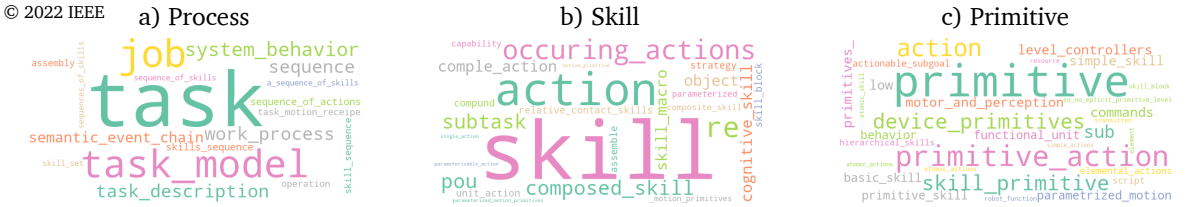



**Figure 5.3:** Wordcloud representing the frequency of appearance of words used to describe *Process* (a), *Skill* (b), and *Primitive* (c) in the reviewed literature. The bigger the word, the more frequent. Reprinted from Capability-based Frameworks for Industrial Robot Skills: a Survey, International Conference on Automation Science and Engineering, Pantano M. et al. [164], Copyright (2022), with permission from IEEE.



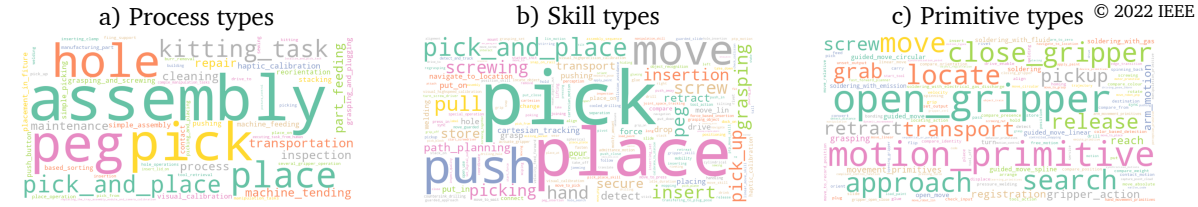



**Figure 5.4:** Worldcloud is used to represent the frequency of *Process* (a), *Skill* (b), and *Primitive* (c) types in the reviewed literature. The bigger the word, the more frequent.Reprinted from Capability-based Frameworks for Industrial Robot Skills: a Survey, International Conference on Automation Science and Engineering, Pantano M. et al. [164], Copyright (2022), with permission from IEEE.

cases, it is possible to conclude that for industrial scenarios, pick and place operations skills should be integrated for targeting assembly processes and that ROS is an accepted framework. Henceforth, when proposing a programming tool for industrial SMEs, it is essential to support pick-and-place applications that use the ROS framework.

## 5.2 Safety for Industrial Scenarios with a Common Terminology

As previous chapters show, safety can be integrated with cobots and user-defined behaviors. However, following the safety procedure outlined by ISO 12100:2010 [16], the risk assessment process is iterative, and documentation must be provided before any cobot application rollout. This applies also to any change performed after the initial commissioning [138]. Having identified that the most investigated capabilities for industrial scenarios are pick and place, there is now a need to investigate how pick and place operations can be described so the safety procedure according to ISO 12100 can be efficiently integrated, avoiding the usage of paper and pencil approaches, which can be hard to edit and review.

### 5.2.1 Motivation and Related Works

The risk assessment of any cobot operation comprises two parts: the risk analysis, where possible hazards are identified, and the risk evaluation, where the hazard severity and probability of occurrence are evaluated. Thus, the risk assessment process involves identifying potential hazards, proposing mitigation measures, and then repeating the assessment to ensure that the measures effectively reduce risk and is a continuous, cyclical process that can be seen as dull and time-consuming [166]. Novel research has proposed automated methods to reduce the number of steps, like in [167]. Other works focus on empowering non-safety experts to conduct a risk assessment, like in [168]. However, to the author's knowledge, the presented work has yet to give specific solutions for pick and place while allowing frequent changes given by user-defined behaviors. Therefore, this section describes a novel software tool employing a standard modeling approach for allowing the risk assessment review for frequent changes in cobot behavior in industrial pick and place.

### 5.2.2 Risk Analysis for Frequent Changes Based on BPMN

The software tool needs a shared description across process changes to allow documentation across different risk assessment iterations. The approach presented here is grounded in describing the human-cobot operations to satisfy this requirement. Such description follows the common terminology identified in Sec. 5.1 as a considerable share of the research community adopts it. However, it is integrated with a standard-

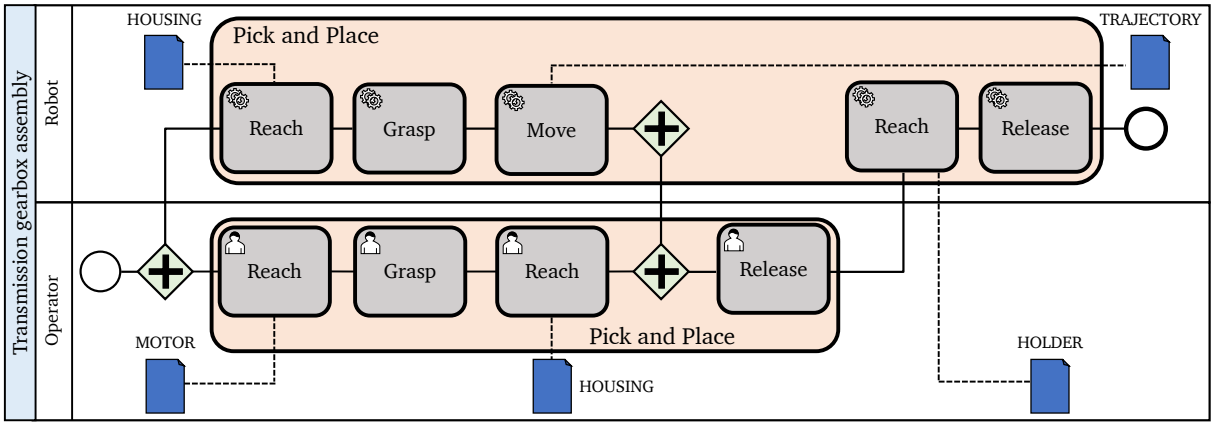



**Figure 5.5:** Usage example of the BPMN-based modeling for a transmission gearbox assembly *Process* modeled as a pool (in blue). Each *Resource* has a swim lane with several *Parameterized skill* (in orange) with several *Primitive* (in gray) connected to data objects *Parameter* (in blue). The flow of the *Process* is described by the connecting lanes and the AND logical operator (in green) for parallel flows.

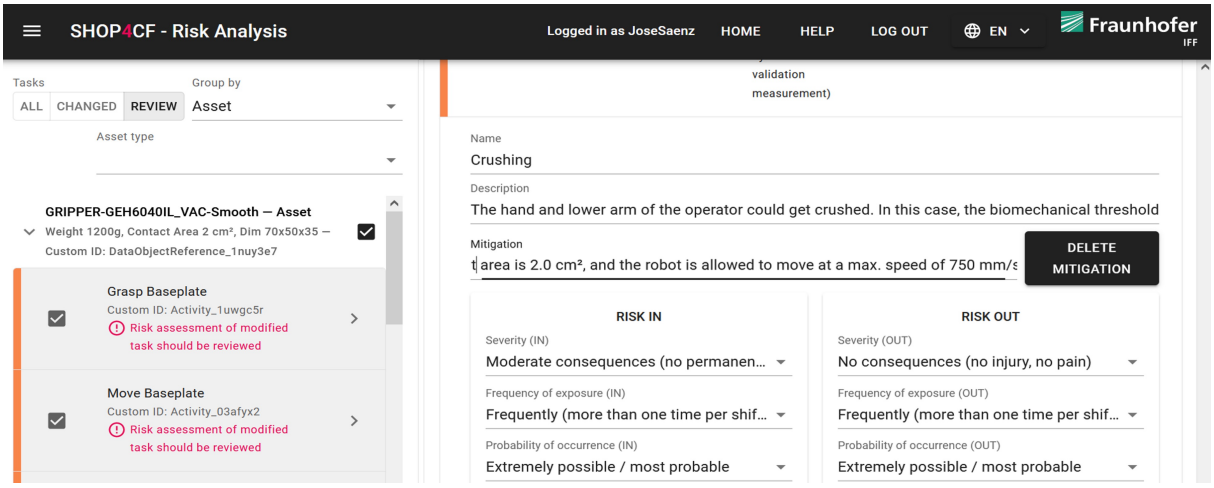


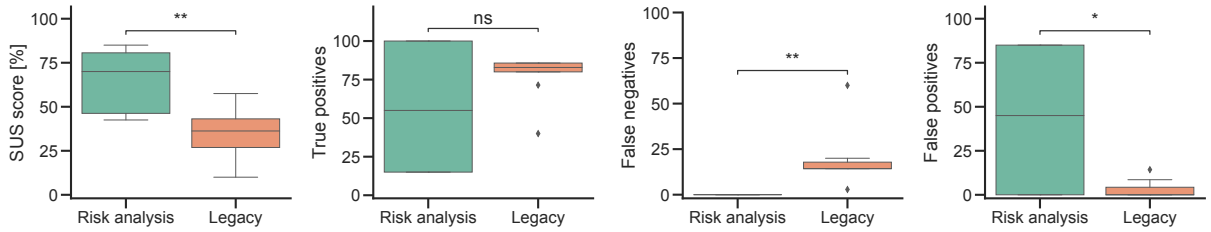

**Figure 5.6:** Screenshot of the review tab of the risk analysis tool. To the left, all the *Primitive* are shown, and if reviews are necessary to change, a pink text is displayed. To the right, the tab to assess the risk level and propose mitigations for each *Primitive* is shown. In the tab, hazards are selected and rated based on severity and probability of occurrence.

ized machine-readable modeling language to allow interaction with a software tool. The Business Process Model and Notation (BPMN) is thus selected. The selection is driven by the fact that BPMN is a modeling language that can also be used for the execution of processes [169], it can be easily used to bridge between business and manufacturing [170] and has a standardized Extensible Markup Language (XML) interpretation. Furthermore, research presented in [171–173] showed how BPMN could be adapted to human-cobot collaborations. However, the modeling notation, in this case, requires several communication messages to synchronize humans and robots, which adds unneeded complexity to the modeling. This is because different pools are associated with each actor, requiring precise timing of actions. In order to simplify these complexities, the proposed modeling approach uses single pools to represent a *Process* with swim lanes associated with a *Resource*. Then, logical gates with merge points divide the work among different *Resource*. Finally, each action of a *Resource* is represented by an activity that symbolizes a *Primitive* with connected data objects as *Parameter*. An example of a human-cobot *Process* using this interaction is shown in Fig. 5.5. With this modeling method with XML representation, the risk analysis tool can interpret the Process and its changes. More specifically, the risk analysis tool needs to be used in designing and adapting cobot applications, and it is meant to guide a user to review relevant changes. To do so, the risk analysis software component parses the BPMN XML description with its parameters first. Then, it guides the safety reviewer to check all the process steps (i.e., *Primitive*), and the risk analysis tool asks to assign hazards and hazard levels considering the probability of occurrence and severity. To simplify the steps, hazards can be selected from a drop-down list containing HRC-related

risks from Annex A of ISO 10218-2. Through this level of granularity, which considers different robot motions and their interaction with the environment, it is thus possible for the safety expert to consider hazard levels as long each *Primitive* has information about robot speed, movement type and involved *Resource*. Once the review process is finished, the safety expert can approve or not the Process. If the changes to the *Process* appear along the way, the risk analysis uses the unique identification numbers of the BPMN XML description to highlight modifications requiring a review. An example of these steps is shown in Fig. 5.6. Thus, with this tool, it is easier to avoid omission errors as long the Process is well documented and the tool prompts the reviewer to check all the changes.

### 5.2.3 Results and Discussions

Having the concept of the risk analysis tool, its efficacy has to be evaluated when compared with legacy tools based on an Excel spreadsheet designed in Visual Basic for Applications (VBA) as long it is a standard tool used in Siemens factories. Therefore, a user study is conceived. The study aims to measure if the risk analysis tool can effectively reduce omission errors in a scenario where a change to an existing process must be made. Therefore, a use case concerning manufacturing a medical gearbox, often subjected to changes due to its small batch production, is selected. In this situation, the users are requested to change the specification of a component and the EE used for the assembly. With these changes, the collision surfaces increased; therefore, the hazard severity had to be reduced along with the mitigation measures. A within-subject study design was performed to conduct the study and compare the tools. In the experiment, eight users participated and interacted with the two systems to achieve the desired modification with a balanced, randomized order. After each interaction, the users rated tool usability with SUS. Additionally, the performance in completing the tasks was recorded. Namely, true positives, false positives, and false positives were monitored. The experiment results with statistical significance are shown in Fig. 5.7. By analyzing the results, it is possible to perceive that the proposed software tool can diminish errors (i.e., false positives) and omissions (i.e., false negatives) while maintaining the same accuracy as the legacy tool (i.e., true positives) with higher usability. More specifically, for the false negative rates, the measured error rates were M = 0.0%, SD = 0.0 for the proposed method, and M = 19.0%, SD = 17.0 for the legacy tool. For the false positive rates, the measured error rates were M = 43.0% SD = 44.0 for the proposed method and M = 3.0% SD = 5.0 for the legacy tool. For the true positive rates, the measured rates were M = 56.87%, SD = 44.79 for the proposed method, and




**(a)** Usability **(b)** True positive rate **(c)** False negative rate **(d)** False positive rate

**Figure 5.7:** Outcomes of the user test. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed. The evaluation shows that the risk analysis tool can reduce the number of false negatives (i.e., omissions) and false positives (i.e., errors). Reprinted from Novel Approach using Risk Analysis Component to Continuously Update Collaborative Robotics Applications in the Smart, Connected Factory Model, Pantano M. et al. [174], Copyright (2022).

M = 77.14%, SD = 15.72 for the legacy tool. For usability using SUS, the proposed tool scored M = 65.0%, SD = 18.32, and the legacy tool scored M = 35.0%, SD = 16.03. A Welch's t-test was conducted after ensuring preconditions to the test. The test yielded statistical significance for usability, false positive rate, and false negative rate but not for the true positive rate.

Considering this data, it is possible to conclude that using a model-based BPMN flow description with the defined common terminology to guide users in reviewing and performing the safety analysis is beneficial as it can reduce the number of omissions and be accepted by a broad user pool. However, further work is required to improve the false positive and true positive rates. Once these have been solved, the barrier of entry for performing risk analysis with frequent changes may be reduced as the tool ensures that all the data is inserted into the tool while reviewing a safety assessment.

## 5.3 User Acceptance of a Model-based Programming

With a well-defined concept for modeling human cobot workflows that can be used for risk analysis, the last step is to allow the integration of user-defined cobot behaviors and monitor the level of acceptance and usability.

### 5.3.1 Motivation and Related Works

As pointed out in the introduction, cobot programming is one of the barriers that influence the acceptance of the cobot design. More precisely, applications in SMEs require cobots that can sense the environment and adapt to its changes [35]. A comprehensive review of cobot programming for industrial tasks is already presented in [30]. How-

ever, in this section, it is essential to consider robot programming technologies that can be applied to different robot brands with Generic Robot Programming (GRP). In GRP, robot programming is done in a programming language or model that is robot-independent; thus, the same program can be executed by different robot brands [175]. However, to allow integration of GRP, a robot must be enabled with skills that can be parameterized, as also identified in Sec. 5.1. Therefore, "low-level drivers," which specify how a robot should interpret each skill, must be integrated. Common examples of GRP usage are in commercial and open source tools like RAZER [176], Collaborative Programming in programming by demonstration [136], Assembly [175] or ArtiMinds[1]. The user is presented with an intuitive interface to program robot motions decoupled from the robot programming language in these tools. Thus, the user selects robot skills and parameterizes them by LfD or simulator. Then, the generalized program is translated into robot execution through appropriate drivers. However, to the author's knowledge, none of the tools allow the user to integrate additional information in the programming, like sensors' data or human-robot collaboration sequences. Thus, an extension of GRP, which also considers possible additional steps, has been integrated with ADAPT [171–173]. In this work, the authors argue that workflow-based programming based on BPMN can create more complex flows that integrate external information. Additionally, robot code can be generated using the standardized BPMN format. However, they need to show if this approach is as usable as the GRP ones. Therefore, although the BPMN approach helps to model complex interactions, usability studies still need to be performed. Therefore, there is a need to investigate if GRP based on BPMN can be as usable as other GRP programming techniques.

### 5.3.2 Programming through a Business Notation

As highlighted in Sec. 5.2, the modeling approach for describing human-robot collaboration workflows is based on a BPMN description. Such workflow contains information regarding *Process*, *Resource*, *Parametrized skill*, *Primitive*, and *Parameter* using the BPMN constructs of the pool, swim lane, group artifacts, actions, and data objects. Therefore, three crucial steps exist to allow the programming of robot behaviors for GRP. First is the workflow description. Second, the population of parameter values is connected to each action. Third, the execution of the robotic hardware. This section describes the adopted approach on the three levels.

[1] https://www.artiminds.com (accessed on 12/12/2022)

#### Workflow Description

As highlighted before, a unique BPMN pool contains the workflow describing one process. Therefore, each action is allocated in separate swim lanes depending on the actor that should execute it. To connect between different actions, a non-robotic expert needs to select and connect directional arrows between them to outline the actions' order. However, following the BPMN standard, actions added by a user can be countless and cannot be easily related to robot operations. Additionally, such actions might not have the correct level of granularity, considering that the modeling approach requires each action to be a *Primitive*. To overcome these limitations and hurdles, an action library describing a limited set of *Primitive* is introduced. By having this library, users are then prompted to select a limited set of actions when describing the workflow. However, a similar set of actions as described by [173] has been selected to ensure that users can express any modeling need. Therefore, actions available in the library are *Primitive* extracted from the Methods Time and Measurement (MTM) [177] and the Robot Time and Motion (RTM) [178] systems. This selection was made as long as MTM and RTM are well adopted in the industrial domain to calculate the performance of robot and human movements. More precisely, considering that the review performed in Sec. 5.1 highlighted that the most investigated operations are pick and place, the set of *Primitive* available in the library are Move, Reach, Grasp, and Release. Where Move stands for motions in the free space not close to objects, Reach stands for robot motions close to physical objects. Moreover, Grasp and Release are for actuating the EE. Through this library, users have all the necessary building blocks for creating pick-and-place scenarios. However, decision points had to be integrated to describe complex flows. In this regard, the logical gates of AND and OR from the BPMN standard are available to the user. Using these, the user can model parallel executions or decision points with if-else statements.

#### Workflow Parametrization

However, to allow the execution of a workflow, the defined *Primitive* with logical gates need parameters either describing positions or trajectories that the robot has to reach or follow. Data artifacts with data object references are used to embed this information in the workflow. Therefore, parameters expressed as data artifacts are connected via references to the *Primitive*. More specifically, the Reach is connected to the object position location, whereas Move is connected to a trajectory to be executed. In this way, the user must specify the target object and trajectory for the execution. A 6D object pose detector and grasp selection are used through a camera system as described in the

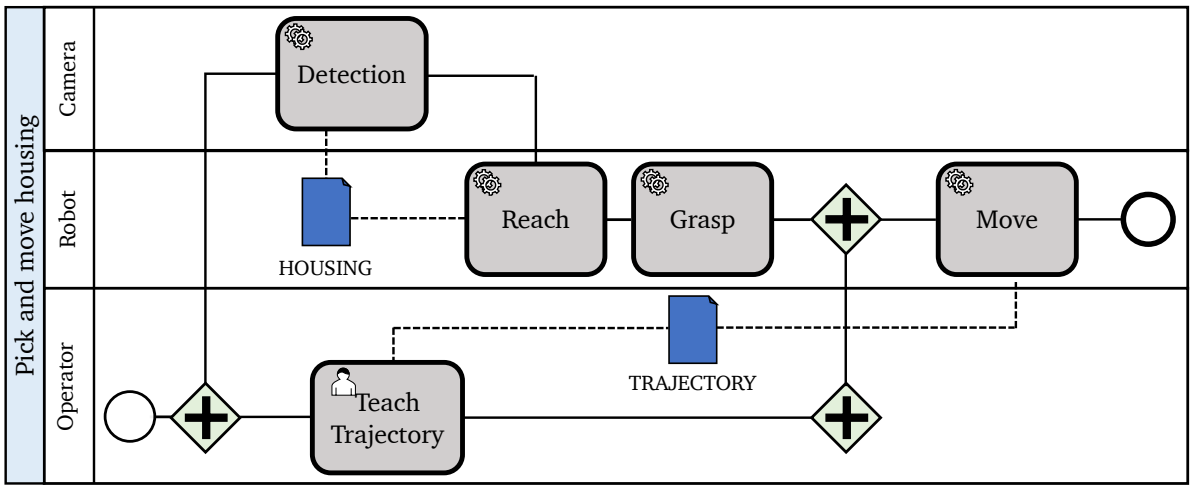



**Figure 5.8:** Example of workflow parametrization using a vision sensor and trajectory teaching. When the workflow starts, the camera detects the housing and saves it in a parameter while the operator teaches a trajectory. Once these steps are done, the robot reaches the housing, grasps it, and then moves it, following the trajectory saved by the operator.

methods of Sec. 3 to obtain position information. Therefore, to consider different parts locations, an additional *Primitive* Detection is added for a camera resource. The path is recorded and saved to obtain trajectory information using the methods explained in Sec. 4. Therefore, to consider teaching different trajectories, an additional *Primitive* Teach trajectory is added to the operator resource. An example of this modeling approach is shown in Fig. 5.8.

### Workflow Execution

Once the workflow has been parameterized, it is time to execute it. As long the workflow is based on BPMN, the BPMN execution semantic is thus exploited. However, considering that the *Process* might have multiple actors, it is mandatory to connect the execution semantics to a common communication framework that can exchange information between robotic systems and environment (e.g., camera, HMI). The architecture proposed in the Smart Human-Oriented Platform for Connected Factories (SHOP4CF) has been adopted for this scope. The architecture in SHOP4CF defines that software components can be used to address manufacturing tasks, which can be used on global or local tasks [179]. Global means that components deal with information between different manufacturing cells, and local means that components deal with information for a single manufacturing cell. A schematic representation of this architecture is shown in Fig. 5.9. In this design, the local layer is the most important because the presented workflow execution deals with processes within a single manu-

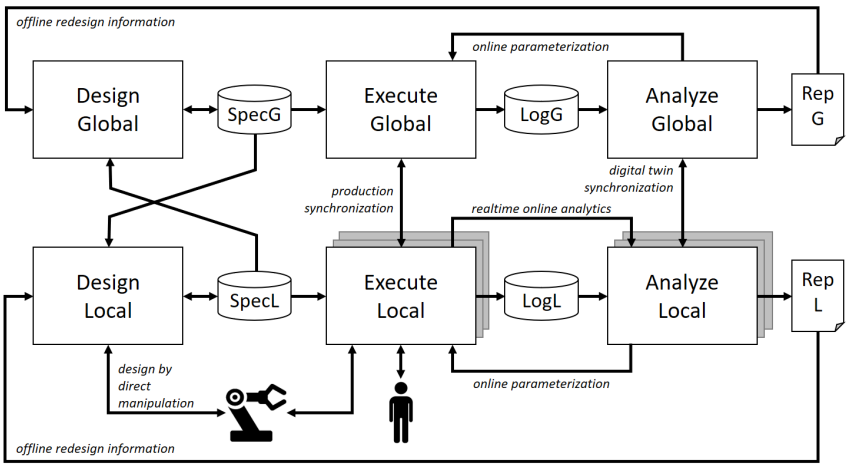



**Figure 5.9:** Architecture for software components developed in SHOP4CF. The system design comprises two primary layers: local and global. The local layer encompasses components dedicated to handling data from individual work cells, while the global layer includes components responsible for managing data across multiple work cells. Reprinted from Deliverable 3.2 SHOP4CF Zimniewicz M. et al. [179], Copyright (2020).

facturing cell. In this layer, the SHOP4CF architecture defines manufacturing processes as being subdivided into several sub-steps known as tasks. Each Task is assigned to an agent in charge of its execution. However, considering that there might be different agents with different software components on a shopfloor, the SHOP4CF architecture defines a standard data model for communicating Task information across agents and components [180]. The data model is displayed in Fig. 5.10. An agnostic middleware, Fiware, is selected to implement this data model. Fiware is a standard-based open-source Internet of Things (IoT) platform [181]. The platform uses a common communication channel to exchange information using standardized data models. The communication pattern follows the publisher/subscriber paradigm. Therefore, data entities, expressed through the NGSI-LD format, are published on a Context Broker (CB)

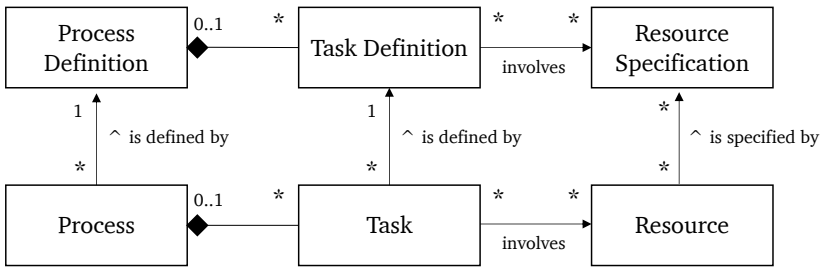



**Figure 5.10:** Data model used in SHOP4CF. The data model defines entities for the execution on the bottom layer and the specification on the top layer. Reprinted from Deliverable 3.3 SHOP4CF Zimniewicz M. et al. [179], Copyright (2021).

through a simple Representational State Transfer Application Programming Interface (REST API), and each agent with CB access can read and write the information. Using the Fiware platform, data entities reflecting the SHOP4CF data model can be published and subscribed. Among the different entities defined in SHOP4CF, the most relevant to the workflow execution is the Task. As previously defined, a task is a substep in a manufacturing process. Therefore, considering the results of Sec. 5.1, it can be identified as *Primitive*. Interacting actors to communicate via Tasks makes it possible to integrate a workflow execution. Task in the SHOP4CF data model are NGSI-LD entities containing information regarding the manufacturing substep, and they can be grouped into four main classes. First, a unique id identifying the Task in the CB and a reference to the activity id of a BPMN workflow. Second is a set of involved agents responsible for executing the Task. Third, work and output parameters are given before or after the task execution—finally, a status parameter describing the task progress. Therefore, once Tasks are published on a CB, they can be queried by different actors, and orchestration among several agents can be achieved. More specifically, the status parameter is designed to contain different values: pending, assigned, inProgress, completed, paused, and failed. Therefore, by reading this parameter, it is possible to know the Task progress and allow the workflow execution. However, for its implementation, Task orchestration software has been added. In this case, the Camunda BPMN execution engine, known as the Camunda platform, is selected. The Camunda platform allows the execution of a BPMN flow and works on the principle of a moving token. More precisely, whenever the token reaches an activity, a signal to start its execution is sent, a Task is published on the CB with a status equal to pending, and the BPMN execution is halted. Afterward, agents read the Task content from the CB and identify if it involves them. If yes, they start the Task execution and change the status to inProgress. Next, once the Task is completed or failed, the status is updated to its respective state. This way, the Camunda platform can resume the BPMN execution once the Task status changes, and the token can move to a new Task. For clarity, this process is shown through the pseudo-code in Algorithm 1. Using this paradigm based on a Task published in a CB and monitored by the Camunda platform, the execution of a BPMN workflow can be simply integrated.

#### Execution on Robotic Hardware

Once a BPMN task workflow has been created, the last step is to allow the execution of *Primitive* on robotic hardware. As highlighted previously, the goal is that different robotic platforms can be used for the execution. Therefore, hardware independence

**Algorithm 1** Execution of a workflow

```
1: Initialize a token T to represent the progress in the BPMN flow
2: while T is not at the end of the BPMN flow do
3:     if T reaches an activity then
4:         Send a signal to start the execution of the activity
5:         Publish a Task τ on the CB with status set to 'pending'
6:         Halt the BPMN execution
7:     Agents read Tasks from the CB
8:     if τ involves the agent and status is set to 'pending' then
9:         Start executing τ
10:        Change the τ status to 'inProgress'
11:    Wait for τ to be completed or failed
12:    Change the τ status to either 'completed' or 'failed'
13:    Resume the BPMN execution
```

is necessary. To solve this issue, a software-based abstraction method is selected. The main scope of this software generalization is to allow the parametrization of robotic actions with standard interfaces, which different robot systems can adapt. The ROS is selected for this scope as long as it is a widely accepted robotic middleware, as also identified in Sec. 5.1. ROS allows the control of different robot brands using standardized interfaces that communicate via a publisher and subscriber paradigm. However, considering the workflow parametrization and description, it is important to denote that the information received by the robotic software will be either the positions of objects with grasping points or trajectories. Moreover, this information will be delivered once the robot must execute the *Primitive*. Thus, an enabled ROS robot must be able to accept goals containing either positions or trajectories and execute them at defined timestamps, reporting their status back to the workflow execution. For this purpose, the Moveit! [182] framework can be used. MoveIt! is an ensemble of libraries and tools for planning and executing robot-independent trajectories. However, an additional abstraction layer based on ROS actions is introduced to simplify the integration of MoveIt! for orchestration with other processes. ROS actions are templates to design a goal-oriented robot behavior and track its progress. In other words, a ROS action enables a robot to have capabilities related to movements abstracted by a software layer. The working principle of ROS actions is relatively simple as long they adopt a server/-client approach. Therefore, a server node is always running in the background, waiting for a signal (i.e., goal) to execute the movement while the client node sends the goal and monitors its execution. This process can be visualized with the following pseudocode in Algorithm 2. ROS action goals, however, can be customized to the application's

needs. Therefore, a custom goal has been created. The custom goal is structured with eleven parameters to satisfy the requirements previously identified. These parameters include two controlling variables specifying if the robot must drive to a goal either with a Point To Point (PTP) trajectory in a straight line or following a predefined trajectory with a linear (LIN) motion. A set of coordinates and orientations of an end position to be reached by the robot EE in the case of the PTP motion control. A set of points defining the trajectory to be followed together with the TCP to which trajectories and end points are defined in the case of LIN control. In this way, the client and server can easily communicate to actuate motions either to reach an endpoint via PTP or follow a predefined trajectory path. Therefore, by using this pattern, robotic actions can be triggered and monitored.

**Algorithm 2** ROS Action paradigm for robot motion

1: Initialize ROS Action Server $\mathcal{S}$ and ROS Action Client $\mathcal{C}$
2: **if** Task $\tau$ for robot is in 'pending' **then**
3: Define custom goal parameters $\mathcal{G}$ for robot motion
4: Send goal $\mathcal{G}$ to ROS Action Server $\mathcal{S}$
5: Change $\tau$ status to 'inProgress'
6: **while** $\mathcal{S}$ is not finished executing $\mathcal{G}$ **do**
7: Monitor execution status using ROS Action Client $\mathcal{C}$
8: Change $\tau$ status to 'completed' or 'failed' depending on the outcome of $\mathcal{S}$

#### Sensors Integration

Similar to the robotic *Primitive*, sensors *Primitive* need to be considered. Analogously to the paradigm proposed in the previous section, actions can be exploited to allow monitoring and execution of sensor capabilities. ROS is a middleware mostly known for integrating libraries for robots; however, it also provides an extensive collection of drivers for sensor devices. Therefore, it can be easily used for sensors' actions. However, different from the robot ones, these actions must be able to both receive goals and send information about the desired sensor value; therefore, feedback has to be integrated. For the camera detection action, this is incorporated as follows. For the goal, the action is structured to require the target object's name, the coordinate frame to which position and orientation are expressed, and finally, the id of the BPMN data object to which this data refers. As feedback, this action returns the path of a file containing the position and orientation of a reliable object's grasping pose alongside the object's position and orientation. For the trajectory recording action, integration was slightly different. For the goal, the action accepts an id of a BPMN data object

referring to the trajectory, a trigger signal to start and stop the recording, and the coordinate system to which the trajectory is saved. As feedback, it returns the path of a file containing the trajectory points with reference to a defined coordinate system. With this approach, sensors can be triggered, and the outcomes of their detection are saved in files that the BPMN process execution can use to parameterize robot actions, as explained in the previous sections.

#### Software Implementation and Connection to Workflow Execution

The last step for the workflow execution is to generate programming instructions to interpret data published in the Fiware CB and coordinate the execution of ROS actions, minimizing the need for programming knowledge. To achieve this goal, three components are needed. First is the integration of ROS actions in ready-to-use packages. Second, a software interpreter will read the data from the Fiware CB and redirect the different requests to the ROS action packages using the paradigm of ROS action clients. Third, deployment of the correct software for each workflow.

To fulfill the first goal, the concept of containerization is exploited [183]. Containerization, or dockerization, consists of creating standalone software containers that fulfill determined actions in a distributed application. Containers are considered standalone as long they include all necessary libraries and dependencies for running specific software, thus limiting the need for an end user to have a software background to use them. Therefore, goals can be sent once application containers are created for each ROS action, avoiding the need to implement software functions, allowing hardware devices to inherit capabilities of performing functions via the installation of application containers, and reducing the need for specialized hardware by the user.

For the second goal, automatic software generation is used. More specifically, considering that each BPMN workflow is modeled following a defined set of rules, as previously explained, it is possible to generate calls to ROS actions automatically. Thus removing the need to reprogram the software interpreter whenever changes to the BPMN workflow are applied. A software parser based on the XML schema of BPMN is used to integrate this capability. The parser is responsible for reading the XML schema and generating code to call the ROS actions with the suitable parameters. This is possible by exploiting the standard structure of the XML and assigning defined ROS actions by classifying the content of a BPMN activity to one of the available *Primitive* via a Natural Language Processing (NLP) pipeline. This pipeline works by reading the name given to each activity in the BPMN and assigning a similarity score to the available capabilities in the ROS actions by using the text classifier of GPT-3 [184] and finally,

creating a file with calls to the ROS actions referring them to the correct BPMN activity ids. Thus, the interpreter can be automatically generated whenever a new BPMN workflow is created. Finally, an automatic deployment of application containers is employed to achieve the last goal. The scope of this last component is to ensure the delivery of software packages to the target hardware. To achieve this goal, methods highlighted before are again applied. Therefore, software deployment is based on the application containers generated for the ROS actions but simplified using the technology of Docker swarm. Docker swarm is a tool for orchestrating containers, and it is provided by Docker[2]. Through Docker Swarm, it is possible to use already-built application containers to deploy applications on target hardware. In this case, Docker swarm is used to deploy a recipe compromising a list of application containers necessary for workflow execution. The list is created by analyzing the BPMN XML schema to identify the necessary actions. Then, the corresponding application container is obtained by the same classification for selecting the correct ROS action. Thus, all the necessary software components are already available when the workflow needs to be executed. Aside from the software pipeline, to guarantee smooth usage in industrial settings, the Siemens Industrial Edge platform is employed in the robotic cell as long it allows the deployment of docker containers to different target devices, which is essential to ensure hardware compatibility. The Siemens Industrial Edge platform was used on a hybrid network and hardware topology, integrating devices for Information Technology (IT) and Operational Technology (OT). This way, the novel workflow based on the IT technologies described above can interface with signals and information from the OT level (e.g., digital signals, safety signals). For clarity, this new topology is shown in Fig.5.11. This integration guarantees that existing hardware for interfacing with signals can be leveraged without developing additional software.

### 5.3.3 Experimental Results

The previous sections presented a novel method for workflow programming, which will be denominated Multi-modal programming for clarity. The next step is to assess its usability against programming methods, which can include sensor data and allow users to customize the robot to drive through defined points given by a sensor or a defined trajectory. The literature review identified that a promising GRP method similar to the one proposed here is Assembly [175]. Therefore, this section presents the results of a user study comparing Multi-modal programming and Assembly.

[2]https://www.docker.com (accessed on 22/12/2022)

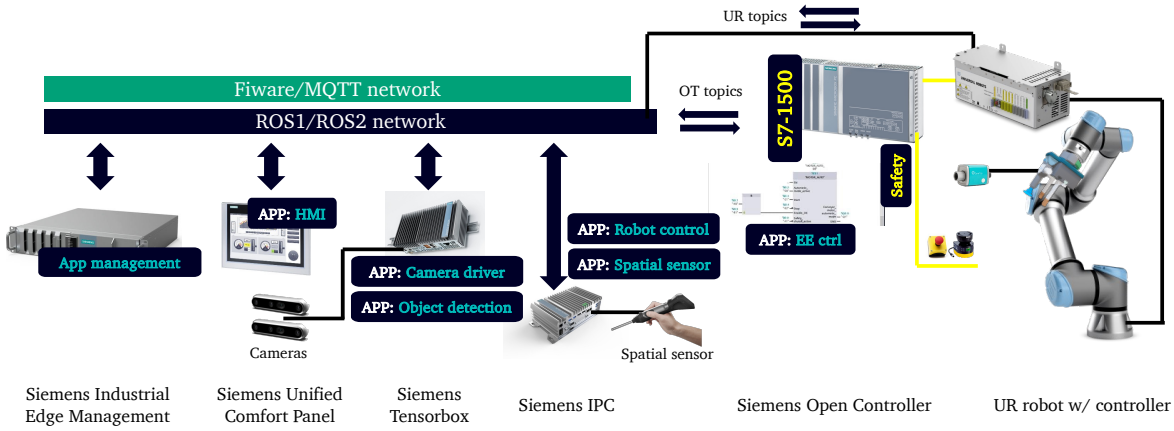



**Figure 5.11:** Network and hardware topology used for the software implementation. The topology fuses hardware from both the IT and OT domains. This guarantees integrating legacy components (e.g., digital inputs) with state-of-the-art IT software. To enable this architecture, the Siemens Industrial Edge Platform. The platform allows the management of apps on different devices.

### Use Case and Experiment Design

A use case from the assembly of a medical gearbox is considered for measuring the difference in usability. More specifically, the slot insertion of a stepper motor into a specific holder is required whenever gearboxes with different torque characteristics need to be assembled. For the sake of clarity, Fig. 5.12 shows the experimental setup for this task. To accomplish this task, the user was in charge of programming three trajectories. This led the robot to pick each motor and then drive towards one of the holders, avoiding an obstacle in the middle with a specific order, i.e., each motor was assigned to a specific holder.

A user study with subsequent interactions was envisioned to compare the two programming methods. For clarity, Fig. 5.13 shows the experiment design. In this experiment, therefore, users were initially introduced to the systems by watching an explanatory video and conducting an exemplary interaction with each method until time $t_1$. Afterward, two subsequent programming interactions unfolded, and at the end of each interaction, usability and workload were measured with SUS and NASA TLX at times $t_2$ and $t_3$. In this way, data for comparing the two methods was available. It is important to denote that in this experiment, the user was mainly in charge of defining the specific trajectory and the order of execution. This was necessary as long as this evaluation concentrated on the workflow programming. Therefore, other parameters like motors' positions and grasp locations were already specified with the method explained in Sec. 3 for the Multi-modal programming and were manually predefined for the Assembly method. Moreover, before replying to the questionnaires after each interaction, the user was shown the outcome of the programming by execution of the

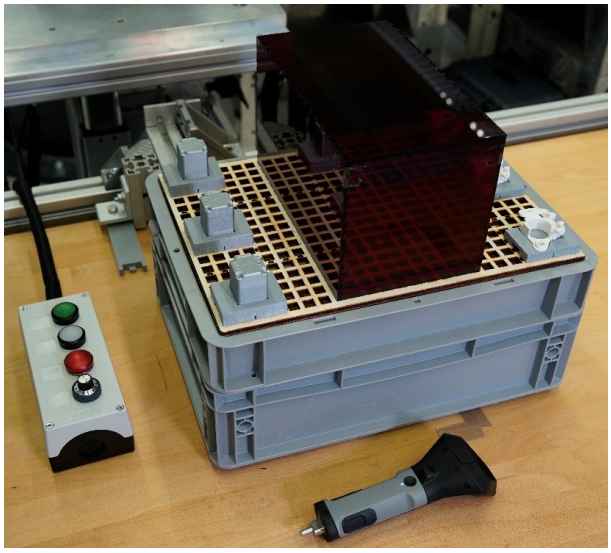


**Figure 5.12:** Use case for the evaluation of the Multi-modal programming. The user was responsible for programming trajectories required for the robot to retrieve motors and position them into their respective holders. During trajectory programming, the users had to navigate around an imaginary obstacle.

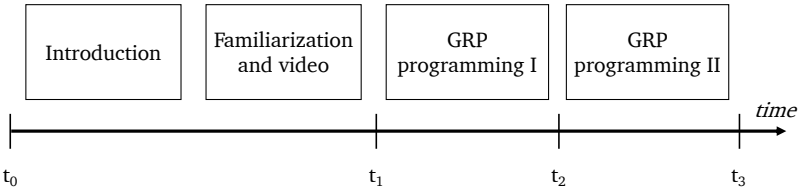
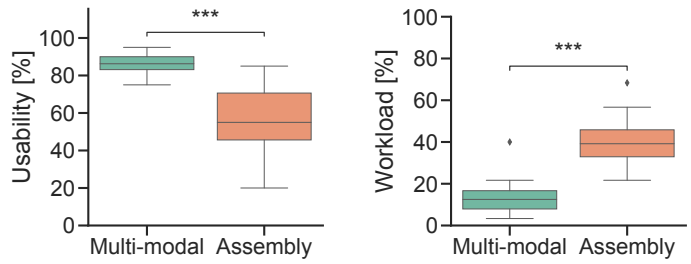



**Figure 5.13:** Experiment design for the comparison of Assembly with Multi-modal programming. Initially, the users were given an introduction. They had time to familiarize themselves with the two systems by watching a video and trying the approach on an exemplary task between $t_0$ and $t_1$. Afterward, the programming interactions unfolded, and at the end of each usability with SUS and workload with NASA TLX were measured at $t_2$ and $t_3$.

programmed workflow on real robotic hardware. This ensured that the usability and workload evaluations considered the outcome of the programming effort.

### Results and Discussions

In the study, 12 individuals participated, age M = 26.07 yrs, SD = 2.09, height M = 178.43 cm, SD = 10.66, and the group was equally distributed between robot experts and non-experts. All of them performed the test correctly, and no data was discarded. The results for usability and workload for the whole user group are shown in Fig. 5.14. As it is possible to see from the figure, the Multi-modal programming approach proves to be more usable and leads to a smaller workload. More precisely, the reported SUS values were M=85.42%, SD=6.56 and M=56.04%, SD=19.17 for Multi-modal programming and Assembly, respectively. Regarding the NASA-TLX values, outcomes were

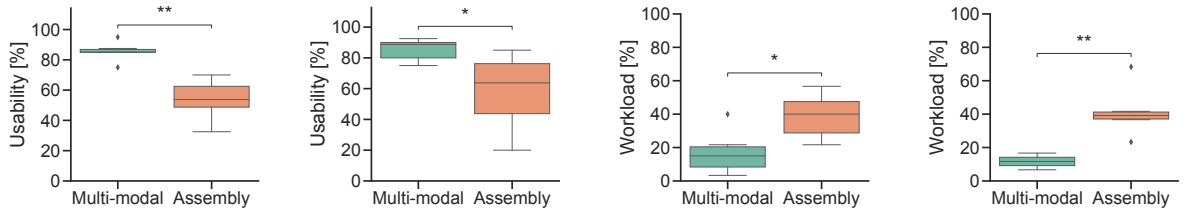



**(a)** Usability **(b)** Workload

**Figure 5.14:** Results of the user evaluation for the proposed programming (Multi-modal programming) and a state-of-the-art tool (Assembly). To the left, usability is measured in percentage according to the SUS scale, and to the right, workload is measured in percentage according to the NASA-TLX scale. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed.

M=14.31%, SD=9.59 and M=40.14%, SD=13.51 for Multi-modal programming and Assembly, respectively. A Mann-Whitney U test was applied to compare the results as long normality precondition did not apply, and the test reported statistical significance. Aside from the general evaluation, it is interesting to see how experts and novices rated the Multi-modal programming; the results for the user group divided upon expertise are shown in Fig. 5.15. Like the whole user group, Multi-modal programming is more usable when discerning between robotic expertise and the Assembly. More specifically, the reported SUS values were M=85.42%, SD=7.31 and M=58.33%, SD=24.88 for Multi-modal programming and Assembly, respectively, for non-experts. For experts, the reported SUS values were M=85.42%, SD=6.41 and M=53.75%, SD=13.30 for Multi-modal programming and Assembly, respectively. The reported TLX values were M=11.67%, SD=3.80 and M=41.39%, SD=14.74 for Multi-modal programming and Assembly, respectively, for non-experts. For experts, the reported TLX values were M=16.94%, SD=13.09 and M=38.89%, SD=13.44 for Multi-modal programming and Assembly, respectively. A Mann-Whitney U test was applied to compare the results as long normality precondition did not apply; the test reported statistical significance for all the results.

Considering the results of this evaluation, it is possible to conclude that the proposed Multi-modal programming proves to be more usable compared with the Assembly one both for non-experts and novices. Therefore, it is a good tool for robot programming in industrial pick-and-place tasks. Therefore, the barrier to entry into robot programming with frequently changing pick-and-place applications and various expertise levels can be addressed when using this tool. However, future work should investigate how the tool behaves in different scenarios and robot brands to ensure the interoperability of the tool in diverse situations.




**(a)** Experts **(b)** Novices **(c)** Experts **(d)** Novices

**Figure 5.15:** Results of the user evaluation for the proposed programming (Multi-modal programming) and a state-of-the-art tool (Assembly). To the left, usability is measured in percentage according to the SUS scale, and to the right, workload is measured in percentage according to the NASA-TLX scale. The statistical significance of the results is displayed with asterisks. If no statistical significance is found, *ns* is displayed.

## 5.4 Conclusions

### Summary

Through this chapter, this work wanted to discover how the proposed intuitive programming interfaces could be scaled to industrial applications. More specifically, this section tackled *Challenge 3.1* (allow SMEs to change robot programs frequently to meet production demands) and *Challenge 1* (intuitive programming interfaces that incorporate various programming techniques) by proposing and studying the impacts of a standard programming framework to integrate all the programming aspects for a collaborative application.

The chapter analyzes the most relevant robotics applications for industrial environments to study such influences. In this part, it was identified that the most common industrial applications require methods for pick and place and that a skill-based representation of robot programming can be beneficial. Henceforth, a workflow representation of the robot behavior was presented to simplify the risk analysis and aid companies in the complex safety evaluation process in collaborative robotic environments with pick-and-place applications based on skills. Such safety analysis leads to lower omission errors than a conventional safety review application, as the measured error rate was null. Last, a robot programming method based on the previous findings and the safety workflow representation was proposed. The programming method, Multi-modal programming, allows for the programming of different robot brands due to the usage of a general architecture proposed in the SHOP4CF project and the ROS. Moreover, the programming tool was tested against a state-of-the-art one in an assembly task. The test reported that the Multi-modal programming method was more usable and led to less workload as long the usability measured with SUS was 85.42% and the workload measured with TLX was 14.31%.

### Discussions and Future Work

As mentioned above, this chapter focused on the applicability of the proposed human-robot interactions in industrial applications. To address this challenge, this chapter proposes a workflow-based programming method for robots that addresses both robot programming simplification and risk assessment for typical industrial applications. Regarding the application, the proposed tool tries to address the pick and place applications, which resulted in being the most investigated in the industry. Regarding risk evaluation, the workflow-based tool simplifies the risk assessment process as long it allows the tracking and classifying of hazards consistently and reliably, reducing errors during the process. Third, regarding integration and scalability in different scenarios, the workflow-based programming, using the SHOP4CF architecture and ROS, can quickly adapt to different robot brands and sensors. Last, regarding usability, the workflow-based tool, with its ability to receive data from multiple sensors like a spatial sensor for trajectory programming, proved much more straightforward considering state-of-the-art tools and considering these points and the initial challenge of this work to address the needs of SMEs to be able to program robots frequently to meet production demands. It is possible to conclude that the workflow-based notation can be integrated into industrial scenarios as long it allows the integration of pick-and-place applications while considering safety, reducing the effort needed for its usage, and allowing for integrating multiple sensors and robots, thus solving the aforementioned challenges. However, future work should focus on testing such tools on a broader user pool to effectively estimate the improvements in usability in long-term studies. Additionally, future work should address how applications different from pick-and-place could be addressed with the workflow-based tool; however, being the tool built around a standardized architecture, this process should be relatively straightforward.

Chapter 6

# Conclusion and future research directions

Contents



Collaborative robots can be an excellent tool for small and medium enterprises to address challenges given by volatile market demands and competitive manufacturing prices. However, their introduction poses some challenges, and their adoption has shown a slowdown, as shown by the change of cobots' year-on-year growth, which reduced to 6% in 2020 [4]. Therefore, the market is now in a sobering phase [143]. Notwithstanding, this phase is typical after an initial hype. Therefore, there is a need for a new technology assessment to determine cobot possibilities and prerequisites [25]. Several technologies and standards have been introduced for cobots. However, there is a need to evaluate how those perform regarding end-user usability and applications in industrial scenarios to ensure acceptance of the technology in manufacturing. This work focused on this latter point and addressed three main barriers to cobots' adoption: interfaces, safety, and applications by proposing a set of ease-to-use programming methodologies that are integrated into a single workflow, which simplifies integration and safety reviews. To present this work's primary outcomes, a summary of each chapter is given, and then discussions and future research directions are presented.

## 6.1 Summary

To present the research content, this work has been structured into four main chapters to address some of the main challenges identified when adopting collaborative

robots, like the necessity for streamlined, user-friendly programming approaches that seamlessly incorporate various cobot programming techniques (*Challenge 1*). Next, the cobot programming techniques should integrate simplified safety assessments (*Challenge 2*). Finally, SMEs must deploy cobots that adapt to changing work environments and production demands (*Challenge 3.1*). However, with ease-of-use interfaces, SMEs' primary obstacle when using robot programming methods is the lack of access to robotic experts (*Challenge 3.2*).

Chapter 2 investigates the influences of user robotic expertise and the level of robotic task personalization through user-defined co-assembly locations for tackling challenge *Challenge 3.2*. Therefore, the need to allow end users to personalize the task was discovered after an initial state-of-the-art review. To address this need, the work proposed a method for users to personalize the task by allowing them to locate materials for the human-robot collaboration at user-defined positions based on computer vision and parameterizable robotic inverse kinematics. Next, considering different robotics backgrounds, two studies were conducted to investigate the impact of such personalization on the user's physical ergonomics and task performance. The studies underlined that allowing users to define co-assembly locations benefits the level of autonomy as the measured average autonomy score improved by 25% independent of the robotics expertise. Moreover, with the approach, users could maintain relatively safe physical ergonomics with an average RULA of 4 while not drastically changing the task performances. Last, it was observed that experts tended to stay 1.00 m closer to the robot compared to the novices.

Chapter 3 focused on investigating influences given by user-defined robotic grasp points on acceptance and robustness to tackle challenge *Challenge 3.2*. Therefore, after an initial state-of-the-art review, which underlined the need for users to be able to bias automatic grasp selection algorithms towards a specific region. The chapter presented three preliminary studies to investigate the precision of users when defining points via a spatial device. Next, having discovered the performances of the spatial device, a method for allowing users to influence the grasp location via a novel user grasp metric was proposed. Finally, the chapter presented studies to compare the proposed method with existing market solutions. The investigations showcased that the approach can improve usability with an average usability score of ca 82% while maintaining a low average workload score of ca 19% and a high grasp success rate of ca 87%, thus allowing non-experts to define grasp locations without extensive prior knowledge.

Chapter 4 focused on the influences on safety and acceptance when users can define their trajectories considering the user's robotic background to tackle *Challenge 2*. Therefore, after an initial state-of-the-art review, which underlined the need for users

to create their trajectories and to study how this influences safety. The chapter presented some preliminary studies to underline how safe applications can be integrated when robots navigate in the free space without cages. Then, a method to allow users to define their trajectories using a spatial device correctly calibrated was presented. Finally, the chapter concluded with a study of how safety is impacted when robotic experts and novices program their trajectories through the spatial device. The studies underlined that such a programming method achieves a high level of autonomy of ca 78% independent from robotics expertise. However, the investigations also identified that expertise plays a role in defining robotics trajectories as novices tended to create trajectories requiring 5% more safety area than experts while maintaining ca 0.10 m larger distance to the robot end-effector.

Chapter 5 presented a workflow-based programming for integrating the different programming methods in a unique tool, which allows for integrated safety revision, especially for pick-an-place applications to tackle *Challenge 3.1* and *Challenge 1*. Therefore, the chapter initially investigated a standard nomenclature known to industrial audiences and then proposed a taxonomy for allowing the programming of robots based on skills. Then, the tool integration on a robotic cell was outlined, and user tests were presented to measure the tool's usability level. The studies identified that with this tool, users do not omit any step in the risk assessment process while achieving a high level of usability as it improved by 51% when compared with a state-of-the-art tool independent from the robotics expertise.

## 6.2 Discussions

This work addressed the main adoption barriers of collaborative robots, proposing and studying novel programming methods based on spatial interactions. The main results of this research are presented following the chapter structure for clarity.

#### Intuitive Programming Interfaces and User Impacts

In the scope of intuitive programming interfaces and their user impact, this work proposes and investigates the impacts of two spatial-based interactions.

The first is a novel spatial programming method based on the user's definition of co-assembly locations. This method was proposed as the research shows that users who can personalize their tasks can better collaborate with a robot. However, considering that adjusting co-assembly locations might influence the task and the operator's well-being, this chapter investigated how different levels of robotics expertise can in-

fluence these factors. The primary outcomes of this chapter were as follows. Firstly, users who can place parts and thus program co-assembly locations perceive a higher task decision autonomy. This higher level of autonomy can benefit collaboration, as also identified by other works in different domains [65, 66]. However, robotic experts and robotic novices use this level of autonomy differently. The studies underlined that when operators can decide on the co-assembly locations, novices tend to choose locations that benefit their physical ergonomics while keeping a larger distance from the robot. Experts chose locations that worsened their physical ergonomics while keeping a closer distance from the robot. Therefore, considering the literature on joint task efficiency [79, 80], this work underlines that experts might treat the robot as a teammate as long they try to minimize the overall travel path, both theirs and the robot. On the other hand, novices might feel differently, and they minimize the robot's travel path while increasing their travel path to minimize the impact on the physical ergonomics, thus optimizing for individual efficiency. Therefore, this work highlights that when operators can personalize the task by modifying co-assembly locations, the operator's background should be considered to ensure a task design that best satisfies the operator's needs.

The second is a novel spatial programming method that enables the user's definition of grasp locations. Similar to the one above, this method was proposed to allow users to personalize the grasp location and improve human-robot collaboration. The primary outcomes of this chapter were as follows. First, if an operator is empowered to mark a location via a spatial device and the information is used to label camera images, inaccuracies of the spatial sensor and the operator must be accounted for differently from what was found by [102]. The chapter demonstrated that convolutional neural networks could solve the issue. Secondly, when users can label points with the spatial device, the usability and workload are improved compared to state-of-the-art techniques based on point clicks. Therefore, this method highlights that non-experts can be empowered to define grasping locations via a natural interaction with a spatial device, and such empowerment leads to improved usability and quality of the defined locations. Therefore, this work highlights that the proposed spatial interaction can be used for robot programming by non-experts. However, they must be aided by algorithms to fine-tune the initial labeling locations, as pointed out by [185].

Taking into account these results, the two chapters demonstrated that the proposed spatial interaction methods can be easily used by non-experts, thus allowing SMEs to better integrate robots without deep robotic expertise and successfully answering the requirements set by *Challenge 3.2*.

#### Intuitive Programming Interfaces and Safety Implications

In the scope of intuitive programming interfaces and safety implications, this work proposed monitoring and estimating how the safety level is impacted when users are empowered to define their trajectory via a spatial sensor. However, considering that the spatial sensor can have inaccuracies and that the level of expertise could impact the type of collaboration, this chapter investigated how expertise influences safety and how to improve the accuracy of the spatial sensor. The primary outcomes of this chapter were as follows. Firstly, the usage of a spatial sensor based on photo-reflective sensors requires a unique calibration technique as long inaccuracies appear the far away the sensor is placed from the initial calibration region; therefore, the proposed calibration can reduce errors that appear in these sensors [186]. Secondly, experts and novices have different ways to program robot trajectories when empowered via the spatial device. More specifically, robotic experts tend to program trajectories that require a smaller safety area than those programmed by novices. Moreover, it was found that novices tended to stay closer to the robot when using the spatial sensor compared to the experts. In contrast, experts tended to stay closer to the robot when executing collision-free trajectories. Hence, considering [79, 80], it is possible to deduce that joint task efficiency is interpreted differently by experts and novices. Experts improve the joint task efficiency when the robot is operating in the autonomous collision-free mode, whereas novices improve the joint task efficiency when they define their trajectory. Therefore, this work highlights that collision avoidance motions might guarantee acceptance only in certain conditions, depending on the user background, thus extending the results of [126]. Therefore, the user's background must also be considered when designing safe robot applications, as operators might behave differently upon their previous exposure to robot collaboration.

Considering these results, it is possible to summarize that the spatial-based programming method can satisfy the needs of SMEs in terms of a simple programming method that can simplify the safety evaluation, also allowing for different user backgrounds, thus satisfying the requirements outlined in *Challenge 2*.

#### Scalability to Industrial Applications

In the scope of scalability to industrial applications, this work proposed a programming method based on a workflow description, which can simplify the programming and review the safety hazards. However, since non-experts can conduct the programming of the robot, the research concentrated on proposing an ease-to-use method for the definition of robotic behaviors, especially targeting pick-and-place applications as long

they are the most required in industrial applications at the time of this research. The primary outcomes of this chapter are as follows. First, workflow-based programming with standardized robotic skills simplifies the review of the risk evaluation, mitigating the number of errors made, and it can be an ease-to-use tool for SMEs to integrate and simplify the safety evaluation as highlighted by [10]. Second, the workflow-based programming method was tested with users, and it was discovered that it improves usability and workload compared to state-of-the-art methods like [175]. Third, the workflow programming method allows the integration of all the spatial programming methods proposed in the previous chapters. Last, the workflow programming method can be integrated, levering existing technology stacks. Therefore, this method answers the primary needs of SMEs to have methods to easily program, deploy, and maintain collaborative robotic applications [10].

Considering this, it is then possible to conclude that the proposed workflow-based programming allows SMEs to integrate different programming methods for satisfying different production demands, thus successfully answering the needs outlined in *Challenge 3.1* and *Challenge 1*.

## 6.3 Future work

This research proposed methods to program robots using spatial interactions and a workflow-based programming method to address four main challenges SMEs face when integrating collaborative applications. During the investigations and the proposal of methods for tackling the challenges, careful attention has been placed on the user's background to ensure that users with limited robotics expertise can use the proposed methods to ensure inclusivity across the diverse employee base. Therefore, this work highlighted that considering the user background is essential for ensuring smooth human-robot collaborations when spatial interactions are used. The proposed methods can help towards this goal. However, to the best of the author's knowledge, this is one of the only works that puts enough effort into investigating the effects of robotic expertise in human-robot collaboration to ensure inclusivity across different ranges of users. Unfortunately, this is just the first ground-stone towards the final goal of allowing anyone, regardless of background, to use robots to help in their tasks. Therefore, there are still several gaps in our understanding regarding how expertise might influence the full spectrum of human-robot collaboration. Therefore, future works should concentrate more on these aspects. Most importantly, when considering different levels of robotics expertise, future work should investigate how robotics knowledge evolves

from novices to experts, especially grounding the research with real-use cases with meaningful robot interactions. To achieve this goal, robotics should work more closely with fields from psychology or neuroscience to deeply understand human behaviors and influences given by previous exposure to robotics technology. For example, compelling research areas are trust levels in human-robot teams and impacts on collaboration, or more specifically, impacts on relative distances between users and robots, especially when users are empowered to personalize their tasks. By studying these relations more, it will be possible to answer some questions about how robotics knowledge evolves and changes human-robot interaction. Understanding these relations will allow engineers and researchers to craft human-robot interactions tailored around the user, ensuring everyone can interact with robots safely and inclusively.

Aside from this, it is essential to underline that the research was unfortunately limited to performing evaluations with user groups ranging between ten and twenty individuals; therefore, aside from the more significant research questions regarding user background highlighted above, future work should focus on testing the proposed concepts with larger user groups. Additionally, of further interest for deeper investigation are spatial interactions that consider pick-and-place applications and have more complex interactions with objects like grinding or insertions as they allow users to better express their needs in different scenarios. This will ensure that a broader range of capabilities can be integrated with collaborative robots, ensuring companies can satisfy all the manufacturing demands considering employees' inclusivity.

Appendix A

# Presented and Published papers

This work incorporates previously published content from various sources. Specifically, there are ten relevant publications, three journal papers, and seven conference papers, listed chronologically:

P1 Pantano, M., Regulin, D., Lutz, B., and Lee, D., "A human-cyber-physical system approach to lean automation using an industrie 4.0 reference architecture", 2020, Procedia Manufacturing, Volume 51.

P2 Pantano, M., Blumberg, A., Regulin, D., Hauser, T., Saenz, J., and Lee, D., "Design of a Collaborative Modular End Effector Considering Human Values and Safety Requirements for Industrial Use Cases," 2022, Human-Friendly Robotics 2021, Springer Proceedings in Advanced Robotics, Volume 23.

P3 Pantano, M., Pavlovskyi, Y., Schulenburg, E., Traganos, K., Ahmadi, S., Regulin, D., Lee, D., and Saenz, J., "Novel Approach using Risk Analysis Component to Continuously Update Collaborative Robotics Applications in the Smart, Connected Factory Model," 2022, Applied Sciences, Volume 12.

P4 Pantano, M., Eiband, T., and Lee, D., "Capability-based Frameworks for Industrial Robot Skills: a Survey," 2022, 18th International Conference on Automation Science and Engineering (CASE).

P5 Pantano, M., Yang, Q., Blumberg, A., Reisch, R., Hauser, T., Lutz, B., Regulin, D., Kamps, T., Traganos, K., and Lee, D., "Influence of task decision autonomy on physical ergonomics and robot performances in an industrial human–robot collaboration scenario," 2022, Frontiers in Robotics and AI, Volume 9.

P6 Reisch, R. T., Pantano, M., Janisch, L., Knoll, A., Lee, D., "Spatial Annotation of Time Series for Data Driven Quality Assurance in Additive Manufacturing," 2022, 17th CIRP Conference on Intelligent Computation in Manufacturing Engineering (ICME).

P7 Caporali, A., Pantano, M., Janisch, L., Regulin, D., Palli, G., and Lee, D., "A Weakly Supervised Semi-automatic Image Labeling Approach for Deformable Linear Objects," 2023, IEEE Robotics and Automation Letters (RA-L), Volume 8.

P8 Pantano, M., Schmidt, M., Bolano G., Schulenburg, E., Regulin, D., and Saenz, J., "Analyzing the Influence of Self-defined Trajectories on Safety and Task Ownership: An Empirical Study," 2023, industrial paper at the 19th IEEE Conference of Automation Science and Engineering (CASE).

P9 Pantano, M., Klass V., Yang Q., Sathuluri A., Regulin D., Janisch L., Zimmermann M., and Lee, D., "Simplifying Robot Grasping in Manufacturing with a Teaching Approach based on a Novel User Grasp Metric," 2023, 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM).

P10 Pantano, M., Curioni A., Regulin D., Kamps, T., and Lee, D., "Effects of Robotic Expertise and Task Knowledge on Physical Ergonomics and Joint Efficiency in a Human-Robot Collaboration Task," 2023, IEEE-RAS International Conference on Humanoid Robots.

Furthermore, the author contributed as a co-author to the following publications: four journal papers and six conference papers, presented in chronological order:

P11 **Pantano, M.**, Kamps, T., Pizzocaro, S., Pantano, G., Corno, M., and Savaresi, S., "Methodology for Plant Specific Cultivation through a Plant Identification pipeline," 2020, November, 2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor).

P12 Reisch, R., Hauser, T., Lutz, B., **Pantano, M.**, Kamps, T., and Knoll, A., "Distance-based multivariate anomaly detection in wire arc additive manufacturing," 2020, December, 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 659-664). IEEE.

P13 Hauser, T., Reisch, R.T., Breese, P.P., Lutz, B.S., **Pantano, M.**, Nalam, Y., Bela, K., Kamps, T., Volpp, J., and Kaplan, A.F., "Porosity in wire arc additive manufacturing of aluminium alloys," 2021, Additive Manufacturing, 41.

P14 Lutz, B., Kisskalt, D., Mayr, A., Regulin, D., **Pantano, M.**, and Franke, J., "In-situ identification of material batches using machine learning for machining operations," 2021, Journal of Intelligent Manufacturing, 32, pp. 1485-1495.

P15 Pizzocaro, S., Corno, M., **Pantano, M.**, and Savaresi, S., "Magnetometer Aided GPS-free Localization of an Autonomous Vineyard Drone," 2021, June, 2021 European Control Conference (ECC) (pp. 1132-1137). IEEE.

P16 Bolzonella, C., Bugin, G., Gallo, P., Gardiman, M., Meneghetti, F., Pallottino, F., Pantano, G., **Pantano, M.**, Rakun, J., Lepej, P., and Vicino, D., "Rovitis 4.0: An Autonomous Robot for Spraying in Vineyards," 2022, Safety, Health and Welfare in Agriculture and Agro-food Systems: Ragusa SHWA 2021, 252, p. 176.

P17 Rautiainen, S., **Pantano, M.**, Traganos, K., Ahmadi, S., Saenz, J., Mohammed, W.M., and Martinez Lastra, J.L., "Multimodal interface for human–robot collaboration," 2022, Machines, 10(10), p. 957.

P18 Rakun, J., **Pantano, M.**, Lepej, P., and Lakota, M., "Sensor fusion-based approach for the field robot localization on Rovitis 4.0 vineyard robot," 2022, International Journal of Agricultural and Biological Engineering, 15(6), pp. 91-95.

P19 Moya, A., Bastida, L., Aguirrezabal, P., **Pantano, M.**, and Abril-Jiménez, P., "Augmented Reality for Supporting Workers in Human–Robot Collaboration," 2023, Multimodal Technologies and Interaction, 7(4), p. 40.

P20 Calciolari L., **Pantano, M.**, Pantano G., and Concheri G.,"Preliminary Design and Analysis of a Modular Autonomous Mobile Robot for Vineyard Operations," 2023, VII International Conference on Safety, Health and Welfare in Agriculture and Agro-food Systems Ragusa SHWA.

# Bibliography


[1] Guenat, S. et al. “Meeting Sustainable Development Goals via Robotics and Autonomous Systems”. In: *Nature Communications* 13.1 (Dec. 2022), p. 3559. ISSN: 2041-1723. DOI: 10.1038/s41467-022-31150-5.

[2] Kiyokawa, T., Takamatsu, J., and Koyanaka, S. “Challenges for Future Robotic Sorters of Mixed Industrial Waste: A Survey”. In: *IEEE Transactions on Automation Science and Engineering* (2022), pp. 1–18. ISSN: 1545-5955, 1558-3783. DOI: 10.1109/TASE.2022.3221969.

[3] Sparrow, R. “Kicking a Robot Dog”. In: *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Christchurch, New Zealand: IEEE, Mar. 2016, pp. 229–229. ISBN: 978-1-4673-8370-7. DOI: 10.1109/HRI.2016.7451756.

[4] International federation of robotics, ed. *World Robotics 2021: Industrial Robots*. Franckfurt: VDMA Services GmbH, 2021. ISBN: 978-3-8163-0746-4.

[5] *Commission Recommendation of 6 May 2003 Concerning the Definition of Micro, Small and Medium-Sized Enterprises*. 2003. (Visited on 10/13/2022).

[6] Perzylo, A. et al. “SMErobotics: Smart Robots for Flexible Manufacturing”. In: *IEEE Robotics & Automation Magazine* 26.1 (Mar. 2019), pp. 78–90. ISSN: 1070-9932, 1558-223X. DOI: 10.1109/MRA.2018.2879747.

[7] European Commission et al. *Annual Report on European SMEs 2020/2021 : Digitalisation of SMEs : Background Document*. Ed. by Hope, K. Publications Office, 2021. DOI: 10.2826/120209.

[8] European Commission. Directorate General for Communication. and Kantar. *SMEs, Start-Ups, Scale-Ups and Entrepreneurship: Summary*. LU: Publications Office, 2020. (Visited on 09/12/2023).

[9] Ballestar, M. T., Díaz-Chao, Á., Sainz, J., and Torrent-Sellens, J. “Knowledge, Robots and Productivity in SMEs: Explaining the Second Digital Wave”. In: *Journal of Business Research* 108 (2020), pp. 119–131. ISSN: 01482963. DOI: 10.1016/j.jbusres.2019.11.017.

[10] Villani, V., Pini, F., Leali, F., and Secchi, C. “Survey on Human–Robot Collaboration in Industrial Settings: Safety, Intuitive Interfaces and Applications”. In: *Mechatronics* 55 (2018), pp. 248–266. ISSN: 09574158. DOI: 10.1016/j.mechatronics.2018.02.009.

[11] Wannasuphoprasit, W., Gillespie, R. B., Colgate, J. E., and Peshkin, M. A. “Cobot Control”. In: *Proceedings of International Conference on Robotics and Automation*. Albuquerque, NM, USA: IEEE, 1997, pp. 3571–3576. ISBN: 0-7803-3612-7. DOI: 10.1109/ROBOT.1997.606888.

[12] Akella, P. et al. “Cobots for the Automobile Assembly Line”. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation*. Vol. 1. Detroit, MI, USA: IEEE, 1999, pp. 728–733. ISBN: 978-0-7803-5180-6. DOI: 10.1109/ROBOT.1999.770061.

[13] Hirzinger, G., Brunner, B., Dietrich, J., and Heindl, J. “Sensor-Based Space Robotics-ROTEX and Its Telerobotic Features”. In: *IEEE Transactions on Robotics and Automation* 9.5 (Oct./1993), pp. 649–663. ISSN: 1042296X. DOI: 10.1109/70.258056.

[14] Rainer, B. et al. “The KUKA-DLR Lightweight Robot Arm - a New Reference Platform for Robotics Research and Manufacturing”. In: *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. 2010, pp. 1–8.

[15] Kristian, K., Hallundbaek, O., and Kasper, S. “Programmable Robot and User Interface”. Pat. WO2007099511A2. 2007.

[16] International Organization for Standardization. *ISO 12100, Safety of Machinery — General Principles for Design — Risk Assessment and Risk Reduction*. Tech. rep. ISO 12100. Geneva: ISO, 2010.

[17] International Organization for Standardization. *ISO 10218-1, Robots and Robotic Devices — Safety Requirements for Industrial Robots — Part 1: Robots*. Tech. rep. ISO 10218-1. Geneva: ISO, 2011.

[18] International Organization for Standardization. *ISO 10218-2, Robots and Robotic Devices — Safety Requirements for Industrial Robots — Part 2: Robot Systems and Integration*. Tech. rep. ISO 10218-2. Geneva: ISO, 2011.

[19] International Electrotechnical Commission. *IEC 61800-5-2, Adjustable Speed Electrical Power Drive Systems - Part 5-2: Safety Requirements - Functional*. Tech. rep. IEC 61800-5-2. Geneva: IEC, 2016.

[20] Marvel, J. A. and Norcross, R. "Implementing Speed and Separation Monitoring in Collaborative Robot Workcells". In: *Robotics and Computer-Integrated Manufacturing* 44 (Apr. 2017), pp. 144–155. ISSN: 07365845. DOI: 10.1016/j.rcim.2016.08.001.

[21] Haddadin, S. et al. "On Making Robots Understand Safety: Embedding Injury Knowledge into Control". In: *The International Journal of Robotics Research* 31.13 (2012), pp. 1578–1602. DOI: 10.1177/0278364912462256.

[22] Haddadin, S., Albu-Schaeffer, A., and Hirzinger, G. "Safety Evaluation of Physical Human-Robot". In: *Robotics: Science and system* 3 (2008), pp. 217–224.

[23] International Organization for Standardization. *ISO/TS 15066, Robots and Robotic Devices — Collaborative Robots*. Tech. rep. Geneva: ISO, 2016.

[24] Aaltonen, I. and Salmi, T. "Experiences and Expectations of Collaborative Robots in Industry and Academia: Barriers and Development Needs". In: *Procedia Manufacturing* 38 (2019), pp. 1151–1158. ISSN: 23519789. DOI: 10.1016/j.promfg.2020.01.204.

[25] Kluy, L. et al. "Mensch-Roboter-Kollaboration in KMU – Potenziale identifizieren, analysieren und realisieren". In: *Digitalisierung der Arbeitswelt im Mittelstand 1*. Ed. by Nitsch, V. et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2022, pp. 55–97. ISBN: 978-3-662-64802-5 978-3-662-64803-2. DOI: 10.1007/978-3-662-64803-2_3.

[26] Oh, S. J. "Emergence of a New Sector via a Business Ecosystem: A Case Study of Universal Robots and the Collaborative Robotics Sector". In: *Technology Analysis & Strategic Management* (Oct. 2021), pp. 1–14. ISSN: 0953-7325, 1465-3990. DOI: 10.1080/09537325.2021.1986212.

[27] Morley, E. C. and Syan, C. S. "Teach Pendants: How Are They for You?" In: *Industrial Robot: An International Journal* 22.4 (Aug. 1995), pp. 18–22. ISSN: 0143-991X. DOI: 10.1108/01439919510098416.

[28] Reisch, R., Hauser, T., Kamps, T., and Knoll, A. "Robot Based Wire Arc Additive Manufacturing System with Context-Sensitive Multivariate Monitoring Framework". In: *Procedia Manufacturing* 51 (2020), pp. 732–739. ISSN: 23519789. DOI: 10.1016/j.promfg.2020.10.103.

[29] Lopez Infante, M. and Kyrki, V. “Usability of Force-Based Controllers in Physical Human-Robot Interaction”. In: *Proceedings of the 6th International Conference on Human-robot Interaction - HRI ’11*. Lausanne, Switzerland: ACM Press, 2011, p. 355. ISBN: 978-1-4503-0561-7. DOI: 10.1145/1957656.1957790.

[30] El Zaatari, S., Marei, M., Li, W., and Usman, Z. “Cobot Programming for Collaborative Industrial Tasks: An Overview”. In: *Robotics and Autonomous Systems* 116 (June 2019), pp. 162–180. ISSN: 09218890. DOI: 10.1016/j.robot.2019.03.003.

[31] Ravichandar, H., Polydoros, A. S., Chernova, S., and Billard, A. “Recent Advances in Robot Learning from Demonstration”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 3.1 (May 2020), pp. 297–330. ISSN: 2573-5144, 2573-5144. DOI: 10.1146/annurev-control-100819-063206.

[32] Fernández-Macías, E., Klenert, D., and Antón, J.-I. “Not so Disruptive yet? Characteristics, Distribution and Determinants of Robots in Europe”. In: *Structural Change and Economic Dynamics* 58 (2021), pp. 76–89. ISSN: 0954349X. DOI: 10.1016/j.strueco.2021.03.010.

[33] Shi, J., Jimmerson, G., Pearson, T., and Menassa, R. “Levels of Human and Robot Collaboration for Automotive Manufacturing”. In: *Proceedings of the Workshop on Performance Metrics for Intelligent Systems - PerMIS ’12*. College Park, Maryland: ACM Press, 2012, p. 95. ISBN: 978-1-4503-1126-7. DOI: 10.1145/2393091.2393111.

[34] Zsifkovits, H., Modrák, V., and Matt, D. T. *Industry 4.0 for SMEs*. Springer Nature, 2020. ISBN: 978-3-030-25425-4.

[35] Schnell, M. and Holm, M. “Challenges for Manufacturing SMEs in the Introduction of Collaborative Robots”. In: *Advances in Transdisciplinary Engineering*. Ed. by Ng, A. H., Syberfeldt, A., Högberg, D., and Holm, M. IOS Press, Apr. 2022. ISBN: 978-1-64368-268-6 978-1-64368-269-3. DOI: 10.3233/ATDE220137.

[36] Gualtieri, L., Palomba, I., Wehrle, E. J., and Vidoni, R. “The Opportunities and Challenges of SME Manufacturing Automation: Safety and Ergonomics in Human–Robot Collaboration”. In: *Industry 4.0 for SMEs*. Ed. by Matt, D. T., Modrák, V., and Zsifkovits, H. Cham: Springer International Publishing, 2020, pp. 105–144. ISBN: 978-3-030-25424-7 978-3-030-25425-4. DOI: 10.1007/978-3-030-25425-4_4.

[37] Kaasinen, E. et al. “A Worker-Centric Design and Evaluation Framework for Operator 4.0 Solutions That Support Work Well-Being”. In: *Human Work Interaction Design. Designing Engaging Automation*. Ed. by Barricelli, B. R. et al. Vol. 544. IFIP Advances in Information and Communication Technology. Cham: Springer International Publishing, 2019, pp. 263–282. ISBN: 978-3-030-05296-6. DOI: 10.1007/978-3-030-05297-3_18.

[38] Souza, J. P. C., Rocha, L. F., Oliveira, P. M., Moreira, A. P., and Boaventura-Cunha, J. “Robotic Grasping: From Wrench Space Heuristics to Deep Learning Policies”. In: *Robotics and Computer-Integrated Manufacturing* 71 (2021), p. 102176. ISSN: 07365845. DOI: 10.1016/j.rcim.2021.102176.

[39] Rook, L. “Mental Models: A Robust Definition”. In: *The Learning Organization* 20.1 (2013), pp. 38–47. ISSN: 0969-6474. DOI: 10.1108/09696471311288519.

[40] Sofge, D. *Trust and Autonomous Systems*. Vol. 2013,7. Technical Report / Association for the Advancement of Artificial Intelligence SS. Palo Alto, Calif.: AAAI Press, 2013. ISBN: 978-1-57735-604-2.

[41] Hoff, K. A. and Bashir, M. “Trust in Automation: Integrating Empirical Evidence on Factors That Influence Trust”. In: *Human factors* 57.3 (2015), pp. 407–434. DOI: 10.1177/0018720814547570.

[42] Teo, G. et al. “Enhancing the Effectiveness of Human-Robot Teaming with a Closed-Loop System”. In: *Applied ergonomics* 67 (2018), pp. 91–103. DOI: 10.1016/j.apergo.2017.07.007.

[43] Demir, K. A., Döven, G., and Sezen, B. “Industry 5.0 and Human-Robot Coworking”. In: *Procedia Computer Science* 158 (2019), pp. 688–695. DOI: 10.1016/j.procs.2019.09.104.

[44] Kolbeinsson, A., Lagerstedt, E., and Lindblom, J. “Foundation for a Classification of Collaboration Levels for Human-Robot Cooperation in Manufacturing”. In: *Production & Manufacturing Research* 7.1 (2019), pp. 448–471. DOI: 10.1080/21693277.2019.1645628.

[45] Shahrdar, S., Menezes, L., and Nojoumian, M. “A Survey on Trust in Autonomous Systems”. In: *Intelligent Computing*. Ed. by Arai, K., Kapoor, S., and Bhatia, R. Vol. 857. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2019, pp. 368–386. ISBN: 978-3-030-01176-5. DOI: 10.1007/978-3-030-01177-2_27.

[46] Tausch, A. and Kluge, A. “The Best Task Allocation Process Is to Decide on One’s Own: Effects of the Allocation Agent in Human–Robot Interaction on Perceived Work Characteristics and Satisfaction”. In: *Cognition, Technology & Work* (2020). ISSN: 1435-5558. DOI: 10.1007/s10111-020-00656-7.

[47] Weiss, A., Wortmeier, A.-K., and Kubicek, B. “Cobots in Industry 4.0: A Roadmap for Future Practice Studies on Human–Robot Collaboration”. In: *IEEE Transactions on Human-Machine Systems* 51.4 (2021), pp. 335–345. ISSN: 2168-2291. DOI: 10.1109/THMS.2021.3092684.

[48] Bauer, W., Bender, M., Braun, M., Rally, P., and Scholtz, O. “Lightweight Robots in Manual Assembly – Best to Start Simply”. In: (2016).

[49] Gualtieri, L., Rauch, E., and Vidoni, R. “Emerging Research Fields in Safety and Ergonomics in Industrial Collaborative Robotics: A Systematic Literature Review”. In: *Robotics and Computer-Integrated Manufacturing* 67 (2021), p. 101998. ISSN: 07365845. DOI: 10.1016/j.rcim.2020.101998.

[50] Berlin, C. and Adams, C. *Production Ergonomics: Designing Work Systems to Support Optimal Human Performance*. London: Ubiquity Press, 2017. ISBN: 978-1-911529-12-5.

[51] Sanchez-Lite, A., Garcia, M., Domingo, R., and Angel Sebastian, M. “Novel Ergonomic Postural Assessment Method (NERPA) Using Product-Process Computer Aided Engineering for Ergonomic Workplace Design”. In: *PloS one* 8.8 (2013), e72703. DOI: 10.1371/journal.pone.0072703.

[52] Baizid, K. et al. “IRoSim: Industrial Robotics Simulation Design Planning and Optimization Platform Based on CAD and Knowledgeware Technologies”. In: *Robotics and Computer-Integrated Manufacturing* 42 (2016), pp. 121–134. ISSN: 07365845. DOI: 10.1016/j.rcim.2016.06.003.

[53] Mgbemena, C. E., Tiwari, A., Xu, Y., Prabhu, V., and Hutabarat, W. “Ergonomic Evaluation on the Manufacturing Shop Floor: A Review of Hardware and Software Technologies”. In: *CIRP Journal of Manufacturing Science and Technology* 30 (2020), pp. 68–78. ISSN: 17555817. DOI: 10.1016/j.cirpj.2020.04.003.

[54] Lidstone, J. et al. “A Survey of Right-Angle Power Tool Use in Canadian Automotive Assembly Plants”. In: *Applied ergonomics* 90 (2021), p. 103171. DOI: 10.1016/j.apergo.2020.103171.

[55] Gläser, D., Fritzsche, L., Bauer, S., and Sylaja, V. J. “Ergonomic Assessment for DHM Simulations Facilitated by Sensor Data”. In: *Procedia CIRP* 41 (2016), pp. 702–705. ISSN: 22128271. DOI: 10.1016/j.procir.2015.12.098.

[56] Ruiz Castro, P., Mahdavian, N., Brolin, E., Högberg, D., and Hanson, L. “IPS IMMA for Designing Human-Robot Collaboration Workstations”. In: ed. by Sascha Wischniewski & Thomas Alexander. Federal Institute for Occupational Safety and Health, 2017, pp. 263–273.

[57] Zhu, W., Fan, X., and Zhang, Y. “Applications and Research Trends of Digital Human Models in the Manufacturing Industry”. In: *Virtual Reality & Intelligent Hardware* 1.6 (2019), pp. 558–579. ISSN: 20965796. DOI: 10.1016/j.vrih.2019.09.005.

[58] Namwongsa, S., Puntumetakul, R., Neubert, M. S., Chaiklieng, S., and Boucaut, R. “Ergonomic Risk Assessment of Smartphone Users Using the Rapid Upper Limb Assessment (RULA) Tool”. In: *PloS one* 13.8 (2018), e0203394. DOI: 10.1371/journal.pone.0203394.

[59] Diego-Mas, J.-A., Alcaide-Marzal, J., and Poveda-Bautista, R. “Errors Using Observational Methods for Ergonomics Assessment in Real Practice”. In: *Human factors* 59.8 (2017), pp. 1173–1187. DOI: 10.1177/0018720817723496.

[60] Rahal, R. et al. “Caring About the Human Operator: Haptic Shared Control for Enhanced User Comfort in Robotic Telemanipulation”. In: *IEEE transactions on haptics* 13.1 (2020), pp. 197–203. DOI: 10.1109/TOH.2020.2969662.

[61] Shafti, A. et al. “Real-Time Robot-assisted Ergonomics”. In: *2019 International Conference on Robotics and Automation (ICRA)*. Montreal, QC, Canada: IEEE, 2019, pp. 1975–1981. ISBN: 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8793739.

[62] Makrini, I. E., Merckaert, K., Winter, J., Lefeber, D., and Vanderborght, B. “Task Allocation for Improved Ergonomics in Human-Robot Collaborative Assembly”. In: *Interaction Studies. Social Behaviour and Communication in Biological and Artificial Systems* 20.1 (2019), pp. 102–133. ISSN: 1572-0373. DOI: 10.1075/is.18018.mak.

[63] Zhe, C., Gines, H., Tomas, S., Shih-En, W., and Yaser, S. “OpenPose Realtime Multi Person 2D Pose Estimation Using Part Affinity Fields”. In: *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* (2019).

[64] Kim, W. et al. “Adaptable Workstations for Human-Robot Collaboration: A Reconfigurable Framework for Improving Worker Ergonomics and Productivity”. In: *IEEE Robotics & Automation Magazine* 26.3 (2019), pp. 14–26. ISSN: 1070-9932. DOI: 10.1109/MRA.2018.2890460.

[65] Karakikes, M. and Nathanael, D. *The Effect of Cognitive Workload on Decision Authority Assignment in Human-Robot Collaboration*. Preprint. In Review, July 2022. DOI: 10.21203/rs.3.rs-1809351/v1. (Visited on 11/02/2022).

[66] Tausch, A., Peifer, C., Kirchhoff, B. M., and Kluge, A. "Human–Robot Interaction: How Worker Influence in Task Allocation Improves Autonomy". In: *Ergonomics* 65.9 (Sept. 2022), pp. 1230–1244. ISSN: 0014-0139, 1366-5847. DOI: 10.1080/00140139.2022.2025912.

[67] Schmidbauer, C., Komenda, T., and Schlund, S. "Teaching Cobots in Learning Factories – User and Usability-Driven Implications". In: *Procedia Manufacturing* 45 (2020), pp. 398–404. ISSN: 23519789. DOI: 10.1016/j.promfg.2020.04.043.

[68] Bonci, A., Cen Cheng, P. D., Indri, M., Nabissi, G., and Sibona, F. "Human-Robot Perception in Industrial Environments: A Survey". In: *Sensors* 21.5 (Feb. 2021), p. 1571. ISSN: 1424-8220. DOI: 10.3390/s21051571.

[69] Kim, J., Candido, C., Thomas, L., and de Dear, R. "Desk Ownership in the Workplace: The Effect of Non-Territorial Working on Employee Workplace Satisfaction, Perceived Productivity and Health". In: *Building and Environment* 103 (July 2016), pp. 203–214. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2016.04.015.

[70] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. "You Only Look Once: Unified, Real-Time Object Detection". In: (June 2015).

[71] Lepetit, V., Moreno-Noguer, F., and Fua, P. "EPnP: An Accurate O(n) Solution to the PnP Problem". In: *International Journal of Computer Vision* 81.2 (2009), pp. 155–166. ISSN: 0920-5691. DOI: 10.1007/s11263-008-0152-6.

[72] Pantano, M. et al. "Influence of Task Decision Autonomy on Physical Ergonomics and Robot Performances in an Industrial Human–Robot Collaboration Scenario". In: *Frontiers in Robotics and AI* 9 (Sept. 2022), p. 943261. ISSN: 2296-9144. DOI: 10.3389/frobt.2022.943261.

[73] McAtamney, L. and Nigel Corlett, E. "RULA: A Survey Method for the Investigation of Work-Related Upper Limb Disorders". In: *Applied ergonomics* 24.2 (1993), pp. 91–99. DOI: 10.1016/0003-6870(93)90080-S.

[74] Yazdanirad, S. et al. "Comparing the Effectiveness of Three Ergonomic Risk Assessment Methods-RULA, LUBA, and NERPA-to Predict the Upper Extremity Musculoskeletal Disorders". In: *Indian journal of occupational and environmental medicine* 22.1 (2018), pp. 17–21. ISSN: 0973-2284. DOI: 10.4103/ijoem.IJOEM_23_18.

[75] Morgeson, F. P. and Humphrey, S. E. “The Work Design Questionnaire (WDQ): Developing and Validating a Comprehensive Measure for Assessing Job Design and the Nature of Work”. In: *The Journal of applied psychology* 91.6 (2006), pp. 1321–1339. ISSN: 0021-9010. DOI: 10.1037/0021-9010.91.6.1321.

[76] Miller, L., Kraus, J., Babel, F., and Baumann, M. “More Than a Feeling Interrelation of Trust Layers in Human-Robot Interaction and the Role of User Dispositions and State Anxiety”. In: *Frontiers in Psychology* 12 (2021), p. 592711. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2021.592711.

[77] Gärling, T and Gärling, E. “Distance Minimization in Downtown Pedestrian Shopping”. In: *Environment and Planning A: Economy and Space* 20.4 (Apr. 1988), pp. 547–554. ISSN: 0308-518X, 1472-3409. DOI: 10.1068/a200547.

[78] Lyons, J., Hansen, S., Hurding, S., and Elliott, D. “Optimizing Rapid Aiming Behaviour: Movement Kinematics Depend on the Cost of Corrective Modifications”. In: *Experimental Brain Research* 174.1 (Sept. 2006), pp. 95–100. ISSN: 0014-4819, 1432-1106. DOI: 10.1007/s00221-006-0426-6.

[79] Török, G., Pomiechowska, B., Csibra, G., and Sebanz, N. “Rationality in Joint Action: Maximizing Coefficiency in Coordination”. In: *Psychological Science* 30.6 (June 2019), pp. 930–941. DOI: 10.1177/0956797619842550.

[80] Török, G., Stanciu, O., Sebanz, N., and Csibra, G. “Computing Joint Action Costs: Co-Actors Minimize the Aggregate Individual Costs in an Action Sequence”. In: *Open Mind* (Aug. 2021), pp. 1–13. ISSN: 2470-2986. DOI: 10.1162/opmi_a_00045.

[81] Dreyfuss, H. “People Come in Assorted Sizes”. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 8.4 (Aug. 1966), pp. 273–277. ISSN: 0018-7208, 1547-8181. DOI: 10.1177/001872086600800402.

[82] Dreyfuss, H. *Designing for People*. New York: Allworth Press, 2003. ISBN: 978-1-58115-312-5.

[83] Tusseyeva, I., Oleinikov, A., Sandygulova, A., and Rubagotti, M. “Perceived Safety in Human–Cobot Interaction for Fixed-Path and Real-Time Motion Planning Algorithms”. In: *Scientific Reports* 12.1 (Nov. 2022), p. 20438. ISSN: 2045-2322. DOI: 10.1038/s41598-022-24622-7.

[84] Fujita, M. et al. “What Are the Important Technologies for Bin Picking? Technology Analysis of Robots in Competitions Based on a Set of Performance Metrics”. In: *Advanced Robotics* (2019), pp. 1–15. ISSN: 0169-1864. DOI: 10.1080/01691864.2019.1698463.

[85] Solowjow, E. et al. “Industrial Robot Grasping with Deep Learning Using a Programmable Logic Controller (PLC)”. In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. Hong Kong, Hong Kong: IEEE, 2020, pp. 97–103. ISBN: 978-1-72816-904-0. DOI: 10.1109/CASE48305.2020.9216902.

[86] Roa, M. A. and Suárez, R. “Grasp Quality Measures: Review and Performance”. In: *Autonomous Robots* 38.1 (Jan. 2015), pp. 65–88. ISSN: 0929-5593, 1573-7527. DOI: 10.1007/s10514-014-9402-3.

[87] Fujita, M. et al. “Bin-Picking Robot Using a Multi-gripper Switching Strategy Based on Object Sparseness”. In: *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. Vancouver, BC, Canada: IEEE, Aug. 2019, pp. 1540–1547. ISBN: 978-1-72810-356-3. DOI: 10.1109/COASE.2019.8842977.

[88] Mahler, J. et al. *Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics*. Aug. 2017. arXiv: 1703.09312 [cs]. (Visited on 08/19/2022).

[89] Morrison, D., Corke, P., and Leitner, J. *Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach*. May 2018. arXiv: 1804.05172 [cs]. (Visited on 08/19/2022).

[90] Zeng, A. et al. “Robotic Pick-and-Place of Novel Objects in Clutter with Multi-Affordance Grasping and Cross-Domain Image Matching”. In: *The International Journal of Robotics Research* 41.7 (June 2022), pp. 690–705. ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364919868017.

[91] Kleeberger, K., Bormann, R., Kraus, W., and Huber, M. F. “A Survey on Learning-Based Robotic Grasping”. In: *Current Robotics Reports* 1.4 (Dec. 2020), pp. 239–249. ISSN: 2662-4087. DOI: 10.1007/s43154-020-00021-6.

[92] Mahler, J. et al. “Learning Ambidextrous Robot Grasping Policies”. In: *Science Robotics* 4.26 (Jan. 2019), eaau4984. ISSN: 2470-9476. DOI: 10.1126/scirobotics.aau4984.

[93] Zeng, A. et al. “Robotic Pick-and-Place of Novel Objects in Clutter with Multi-Affordance Grasping and Cross-Domain Image Matching”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD: IEEE, May 2018, pp. 3750–3757. ISBN: 978-1-5386-3081-5. DOI: 10.1109/ICRA.2018.8461044.

[94] Satish, V., Mahler, J., and Goldberg, K. “On-Policy Dataset Synthesis for Learning Robot Grasping Policies Using Fully Convolutional Deep Networks”. In: *IEEE Robotics and Automation Letters* 4.2 (Apr. 2019), pp. 1357–1364. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2019.2895878.

[95] Sun, M. and Gao, Y. “GATER: Learning Grasp-Action-Target Embeddings and Relations for Task-Specific Grasping”. In: *IEEE Robotics and Automation Letters* 7.1 (Jan. 2022), pp. 618–625. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2021.3131378.

[96] Liu, Y., Qian, K., Xu, X., Zhou, B., and Fang, F. “Grasp Pose Learning from Human Demonstration with Task Constraints”. In: *Journal of Intelligent & Robotic Systems* 105.2 (June 2022), p. 37. ISSN: 0921-0296, 1573-0409. DOI: 10.1007/s10846-022-01650-z.

[97] Yun Jiang, Moseson, S., and Saxena, A. “Efficient Grasping from RGBD Images: Learning Using a New Rectangle Representation”. In: *2011 IEEE International Conference on Robotics and Automation*. Shanghai, China: IEEE, May 2011, pp. 3304–3311. ISBN: 978-1-61284-386-5. DOI: 10.1109/ICRA.2011.5980145.

[98] Sager, C., Janiesch, C., and Zschech, P. “A Survey of Image Labelling for Computer Vision Applications”. In: *Journal of Business Analytics* 4.2 (July 2021), pp. 91–110. ISSN: 2573-234X, 2573-2358. DOI: 10.1080/2573234X.2021.1908861.

[99] Zhang, Z. et al. *Understanding Physical Effects for Effective Tool-use*. June 2022. arXiv: 2206.14998 [cs]. (Visited on 08/22/2022).

[100] Pires, J., Godinho, T., and Araújo, R. “Using Digital Pens to Program Welding Tasks”. In: *Industrial Robot: An International Journal* 34.6 (Oct. 2007). Ed. by Madsen, O., pp. 476–486. ISSN: 0143-991X. DOI: 10.1108/01439910710832075.

[101] van Delden, S., Umrysh, M., Rosario, C., and Hess, G. “Pick-and-place Application Development Using Voice and Visual Commands”. In: *Industrial Robot: An International Journal* 39.6 (Oct. 2012), pp. 592–600. ISSN: 0143-991X. DOI: 10.1108/01439911211268796.

[102] Gregorio, D., Tonioni, A., Palli, G., and Di Stefano, L. “Semiautomatic Labeling for Deep Learning in Robotics”. In: *IEEE Transactions on Automation Science and Engineering* 17.2 (2020), pp. 611–620. ISSN: 1545-5955. DOI: 10.1109/TASE.2019.2938316.

[103] Reisch, R. et al. “Distance-Based Multivariate Anomaly Detection in Wire Arc Additive Manufacturing”. In: *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Miami, FL, USA: IEEE, Dec. 2020, pp. 659–664. ISBN: 978-1-72818-470-8. DOI: 10.1109/ICMLA51294.2020.00109.

[104] Zhang, W., Ma, X., Cui, L., and Chen, Q. “3 Points Calibration Method of Part Coordinates for Arc Welding Robot”. In: *Intelligent Robotics and Applications*. Ed. by Xiong, C., Huang, Y., Xiong, Y., and Liu, H. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 216–224. ISBN: 978-3-540-88513-9.

[105] Nguyen, H. G., Habiboglu, R., and Franke, J. “Enabling Deep Learning Using Synthetic Data: A Case Study for the Automotive Wiring Harness Manufacturing”. In: *Procedia CIRP* 107 (2022), pp. 1263–1268. ISSN: 22128271. DOI: 10.1016/j.procir.2022.05.142.

[106] Sofiiuk, K., Petrov, I. A., and Konushin, A. “Reviving Iterative Training with Mask Guidance for Interactive Segmentation”. In: *2022 IEEE International Conference on Image Processing (ICIP)*. Bordeaux, France: IEEE, Oct. 2022, pp. 3141–3145. ISBN: 978-1-66549-620-9. DOI: 10.1109/ICIP46576.2022.9897365.

[107] He, K., Zhang, X., Ren, S., and Sun, J. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.90.

[108] Denninger, M. et al. “BlenderProc”. In: (2019). DOI: 10.48550/ARXIV.1911.01911.

[109] Brooke, J. *SUS - A Quick and Dirty Usability Scale*. CRC Press, 1996. ISBN: 978-0-7484-0460-5.

[110] Hart, S. G. and Staveland, L. E. “Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research”. In: *Advances in Psychology*. Vol. 52. Elsevier, 1988, pp. 139–183. ISBN: 978-0-444-70388-0. DOI: 10.1016/S0166-4115(08)62386-9.

[111] Mahler, J. et al. “Dex-Net 3.0: Computing Robust Robot Vacuum Suction Grasp Targets in Point Clouds Using a New Analytic Model and Deep Learning”. In: (2017). DOI: 10.48550/ARXIV.1709.06670.

[112] Yang, J., Gao, Y., Li, D., and Waslander, S. L. “ROBI: A Multi-View Dataset for Reflective Objects in Robotic Bin-Picking”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Prague, Czech Republic: IEEE, Sept. 2021, pp. 9788–9795. ISBN: 978-1-66541-714-3. DOI: 10.1109/IROS51168.2021.9635871.

[113] Kleeberger, K., Landgraf, C., and Huber, M. F. *Large-Scale 6D Object Pose Estimation Dataset for Industrial Bin-Picking*. Dec. 2019. arXiv: 1912.12125 [cs]. (Visited on 08/19/2022).

[114] Pantano, M. et al. “Design of a Collaborative Modular End Effector Considering Human Values and Safety Requirements for Industrial Use Cases”. In: *Human-Friendly Robotics 2021*. Ed. by Palli, G., Melchiorri, C., and Meattini, R. Vol. 23. Springer Proceedings in Advanced Robotics. Cham: Springer International Publishing, 2022, pp. 45–60. ISBN: 978-3-030-96358-3. DOI: 10.1007/978-3-030-96359-0_4.

[115] Bangor, A., Kortum, P. T., and Miller, J. T. “An Empirical Evaluation of the System Usability Scale”. In: *International Journal of Human-Computer Interaction* 24.6 (2008), pp. 574–594. DOI: 10.1080/10447310802205776.

[116] Mildenhall, B. et al. “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”. In: *Computer Vision – ECCV 2020*. Ed. by Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M. Vol. 12346. Cham: Springer International Publishing, 2020, pp. 405–421. ISBN: 978-3-030-58451-1 978-3-030-58452-8. DOI: 10.1007/978-3-030-58452-8_24.

[117] Ichnowski, J., Avigal, Y., Kerr, J., and Goldberg, K. *Dex-NeRF: Using a Neural Radiance Field to Grasp Transparent Objects*. Oct. 2021. arXiv: 2110.14217 [cs]. (Visited on 08/22/2022).

[118] Beschi, M., Faroni, M., Copot, C., and Pedrocchi, N. “How Motion Planning Affects Human Factors in Human-Robot Collaboration”. In: *IFAC-PapersOnLine* 53.5 (2020), pp. 744–749. ISSN: 24058963. DOI: 10.1016/j.ifacol.2021.04.167.

[119] Dragan, A. D., Lee, K. C., and Srinivasa, S. S. “Legibility and Predictability of Robot Motion”. In: *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Tokyo, Japan: IEEE, Mar. 2013, pp. 301–308. ISBN: 978-1-4673-3101-2 978-1-4673-3099-2 978-1-4673-3100-5. DOI: 10.1109/HRI.2013.6483603.

[120] Hall, E. T. *The Hidden Dimension*. New York: Anchor Books, 1990. ISBN: 978-0-385-08476-5.

[121] Stark, J., Mota, R. R., and Sharlin, E. "Personal Space Intrusion in Human-Robot Collaboration". In: *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. Chicago IL USA: ACM, Mar. 2018, pp. 245–246. ISBN: 978-1-4503-5615-2. DOI: 10.1145/3173386.3176998.

[122] Kheng Lee Koay, Walters, M., and Dautenhahn, K. "Methodological Issues Using a Comfort Level Device in Human-Robot Interactions". In: *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.* Nashville, TN, USA: IEEE, 2005, pp. 359–364. ISBN: 978-0-7803-9274-8. DOI: 10.1109/ROMAN.2005.1513805.

[123] Takayama, L., Dooley, D., and Ju, W. "Expressing Thought: Improving Robot Readability with Animation Principles". In: *Proceedings of the 6th International Conference on Human-robot Interaction - HRI '11*. Lausanne, Switzerland: ACM Press, 2011, p. 69. ISBN: 978-1-4503-0561-7. DOI: 10.1145/1957656.1957674.

[124] Koppenborg, M., Nickel, P., Naber, B., Lungfiel, A., and Huelke, M. "Effects of Movement Speed and Predictability in Human-Robot Collaboration". In: *Human Factors and Ergonomics in Manufacturing & Service Industries* 27.4 (July 2017), pp. 197–209. ISSN: 10908471. DOI: 10.1002/hfm.20703.

[125] Aeraiz-Bekkis, D., Ganesh, G., Yoshida, E., and Yamanobe, N. "Robot Movement Uncertainty Determines Human Discomfort in Co-worker Scenarios". In: *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*. Singapore: IEEE, Apr. 2020, pp. 59–66. ISBN: 978-1-72816-139-6. DOI: 10.1109/ICCAR49639.2020.9108085.

[126] Lasota, P. A. and Shah, J. A. "Analyzing the Effects of Human-Aware Motion Planning on Close-Proximity Human–Robot Collaboration". In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 57.1 (Feb. 2015), pp. 21–33. ISSN: 0018-7208, 1547-8181. DOI: 10.1177/0018720814565188.

[127] Sisbot, E. A. and Alami, R. "A Human-Aware Manipulation Planner". In: *IEEE Transactions on Robotics* 28.5 (Oct. 2012), pp. 1045–1057. ISSN: 1552-3098, 1941-0468. DOI: 10.1109/TRO.2012.2196303.

[128] Mainprice, J. et al. "Planning Human-Aware Motions Using a Sampling-Based Costmap Planner". In: *2011 IEEE International Conference on Robotics and Automation*. Shanghai, China: IEEE, May 2011, pp. 5012–5017. ISBN: 978-1-61284-386-5. DOI: 10.1109/ICRA.2011.5980048.

[129] Zhao, X. and Pan, J. “Considering Human Behavior in Motion Planning for Smooth Human-Robot Collaboration in Close Proximity”. In: *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. Nanjing: IEEE, Aug. 2018, pp. 985–990. ISBN: 978-1-5386-7980-7. DOI: 10.1109/ROMAN.2018.8525607.

[130] Faroni, M., Beschi, M., and Pedrocchi, N. “Safety-Aware Time-Optimal Motion Planning with Uncertain Human State Estimation”. In: *IEEE Robotics and Automation Letters* (2022), pp. 1–8. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2022.3211493.

[131] Hopko, S. K. and Mehta, R. K. “Trust in Shared-Space Collaborative Robots: Shedding Light on the Human Brain”. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* (June 2022), p. 001872082211090. ISSN: 0018-7208, 1547-8181. DOI: 10.1177/00187208221109039.

[132] Calinon, S. and Lee, D. “Learning Control”. In: *Humanoid Robotics: A Reference*. Springer Netherlands, 2017, pp. 1–52.

[133] Saveriano, M., An, S.-i., and Lee, D. “Incremental Kinesthetic Teaching of End-Effector and Null-Space Motion Primitives”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA, USA: IEEE, May 2015, pp. 3570–3575. ISBN: 978-1-4799-6923-4. DOI: 10.1109/ICRA.2015.7139694.

[134] Zhu, Z. and Hu, H. “Robot Learning from Demonstration in Robotic Assembly: A Survey”. In: *Robotics* 7.2 (Apr. 2018), p. 17. ISSN: 2218-6581. DOI: 10.3390/robotics7020017.

[135] Lee, D. and Ott, C. “Incremental Kinesthetic Teaching of Motion Primitives Using the Motion Refinement Tube”. In: *Autonomous Robots* 31.2-3 (Oct. 2011), pp. 115–131. ISSN: 0929-5593, 1573-7527. DOI: 10.1007/s10514-011-9234-3.

[136] Eiband, T., Willibald, C., Tannert, I., Weber, B., and Lee, D. “Collaborative Programming of Robotic Task Decisions and Recovery Behaviors”. In: *Autonomous Robots* (Oct. 2022). ISSN: 0929-5593, 1573-7527. DOI: 10.1007/s10514-022-10062-9.

[137] Edmonds, M. et al. “Feeling the Force: Integrating Force and Pose for Fluent Discovery through Imitation Learning to Open Medicine Bottles”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vancouver, BC: IEEE, Sept. 2017, pp. 3530–3537. ISBN: 978-1-5386-2682-5. DOI: 10.1109/IROS.2017.8206196.

[138] *Directive 2006/42/EC of the European Parliament and of the Council of 17 May 2006 on Machinery, and Amending Directive 95/16/EC*. 2006.

[139] Byner, C., Matthias, B., and Ding, H. "Dynamic Speed and Separation Monitoring for Collaborative Robot Applications – Concepts and Performance". In: *Robotics and Computer-Integrated Manufacturing* 58 (Aug. 2019), pp. 239–252. ISSN: 07365845. DOI: 10.1016/j.rcim.2018.11.002.

[140] Vogel, C. and Elkmann, N. "Novel Safety Concept for Safeguarding and Supporting Humans in Human-Robot Shared Workplaces with High-Payload Robots in Industrial Applications". In: *Proceedings of the Companion of the 2017 IEEE International Conference on Human-Robot Interaction*. Vienna Austria: ACM, Mar. 2017, pp. 315–316. ISBN: 978-1-4503-4885-0. DOI: 10.1145/3029798.3038314.

[141] Saenz, J. et al. "Methods for Considering Safety in Design of Robotics Applications Featuring Human-Robot Collaboration". In: *The International Journal of Advanced Manufacturing Technology* 107.5-6 (Mar. 2020), pp. 2313–2331. ISSN: 0268-3768, 1433-3015. DOI: 10.1007/s00170-020-05076-5.

[142] Saenz, J., Schulenburg, E., Behrens, R., and Elkmann, N. "Experiences in Applying a New Approach to Designing Safe HRC Applications". In: *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*. Tokoname, Japan: IEEE, Mar. 2021, pp. 139–143. ISBN: 978-1-66543-862-9. DOI: 10.1109/ISR50024.2021.9419566.

[143] Kopp, T., Schäfer, A., and Kinkel, S. "Kollaborierende Oder Kollaborationsfähige Roboter? Welche Rolle Spielt Die Mensch-Roboter-Kollaboration in Der Praxis?" In: *Industrie 4.0 Management* 2020.2 (Apr. 2020), pp. 19–23. ISSN: 23649208. DOI: 10.30844/I40M_20-2_S19-23.

[144] Pantano, M., Regulin, D., Lutz, B., and Lee, D. "A Human-Cyber-Physical System Approach to Lean Automation Using an Industrie 4.0 Reference Architecture". In: *Procedia Manufacturing* 51 (2020), pp. 1082–1090. ISSN: 23519789. DOI: 10.1016/j.promfg.2020.10.152.

[145] International Organization for Standardization. *ISO 20218-1, Robotics — Safety Design for Industrial Robot Systems — Part 1: End-effectors*. Tech. rep. Geneva: ISO, 2018.

[146] "Methodik Und Organisation Des Kostenmanagements Für Die Produktentwicklung". In: *Kostengünstig Entwickeln Und Konstruieren*. Ed. by Ehrlenspiel, K., Kiewert, A., and Lindemann, U. VDI-Buch. Berlin, Heidelberg: Springer Berlin

Heidelberg, 2007, pp. 35–121. ISBN: 978-3-540-74222-7. DOI: 10.1007/978-3-540-74223-4_4.

[147] Salunkhe, O., Fager, P., and Fast-Berglund, Å. "Framework for Identifying Gripper Requirements for Collaborative Robot Applications in Manufacturing". In: *Advances in Production Management Systems. The Path to Digital Transformation and Innovation of Production Management Systems*. Ed. by Lalic, B., Majstorovic, V., Marjanovic, U., von Cieminski, G., and Romero, D. Vol. 591. Cham: Springer International Publishing, 2020, pp. 655–662. ISBN: 978-3-030-57992-0 978-3-030-57993-7. DOI: 10.1007/978-3-030-57993-7_74.

[148] Verein Deutscher Ingenieure. *VDI 2225-3, Konstruktionsmethodik - Technisch-wirtschaftliches Konstruieren - Technisch-wirtschaftliche Bewertung*. Tech. rep. Düsseldorf: VDI, 1998.

[149] Likert, R. "A Technique for the Measurement of Attitudes". In: *Archives of Psychology* 22 140 (1932), p. 55.

[150] Saenz, J. et al. "COVR Toolkit – Supporting Safety of Interactive Robotics Applications". In: *2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS)*. Magdeburg, Germany: IEEE, 2021, pp. 1–6. ISBN: 978-1-66540-170-8. DOI: 10.1109/ICHMS53169.2021.9582659.

[151] Lee, D. T. and Schachter, B. J. "Two Algorithms for Constructing a Delaunay Triangulation". In: *International Journal of Computer & Information Sciences* 9.3 (June 1980), pp. 219–242. ISSN: 0091-7036, 1573-7640. DOI: 10.1007/BF00977785.

[152] Moya, A., Bastida, L., Aguirrezabal, P., Pantano, M., and Abril-Jiménez, P. "Augmented Reality for Supporting Workers in Human–Robot Collaboration". In: *Multimodal Technologies and Interaction* 7.4 (Apr. 2023), p. 40. ISSN: 2414-4088. DOI: 10.3390/mti7040040.

[153] Juelg, C., Hermann, A., Roennau, A., and Dillmann, R. "Efficient, Collaborative Screw Assembly in a Shared Workspace". In: *Intelligent Autonomous Systems 15*. Ed. by Strand, M., Dillmann, R., Menegatti, E., and Ghidoni, S. Vol. 867. Cham: Springer International Publishing, 2019, pp. 837–848. ISBN: 978-3-030-01369-1 978-3-030-01370-7. DOI: 10.1007/978-3-030-01370-7_65.

[154] Universal Robots. *Stop performance categories CB3*. 2021. URL: https://www.universal-robots.com/articles/ur/safety/what-is-cb3-stop-performance-categories/.

[155] Diedrich, C. et al. *Information Model for Capabilities, Skills & Services: Definition of Terminology and Proposal for a Technology-Independent Information Model for Capabilities and Skills in Flexible Manufacturing*. Discussion Paper. Plattform Industrie 4.0, Oct. 2022, p. 43.

[156] Backhaus, J. and Reinhart, G. "Digital Description of Products, Processes and Resources for Task-Oriented Programming of Assembly Systems". In: *Journal of Intelligent Manufacturing* 28.8 (2017), pp. 1787–1800. ISSN: 0956-5515. DOI: 10.1007/s10845-015-1063-3.

[157] Froschauer, R., Köcher, A., Meixner, K., Schmitt, S., and Spitzer, F. "Capabilities and Skills in Manufacturing: A Survey Over the Last Decade of ETFA". In: (Apr. 2022).

[158] Booth, A., Sutton, A., Clowes, M., and Martyn-St James, M. *Systematic Approaches to a Successful Literature Review*. Third edition / Andrew Booth, Anthea Sutton, Mark Clowes, Marrissa Martyn-St James. Los Angeles: SAGE, 2021. ISBN: 978-1-5297-1184-4.

[159] Tranfield, D., Denyer, D., and Smart, P. "Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review". In: *British Journal of Management* 14.3 (Sept. 2003), pp. 207–222. ISSN: 1045-3172, 1467-8551. DOI: 10.1111/1467-8551.00375.

[160] Cutting-Decelle, A. F. et al. "ISO 15531 MANDATE: A Product-process-resource Based Approach for Managing Modularity in Production Management". In: *Concurrent Engineering* 15.2 (2007), pp. 217–235. ISSN: 1063-293X. DOI: 10.1177/1063293X07079329.

[161] Schlenoff, C. et al. "An IEEE Standard Ontology for Robotics and Automation". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vilamoura-Algarve, Portugal: IEEE, Oct. 2012, pp. 1337–1342. ISBN: 978-1-4673-1736-8 978-1-4673-1737-5 978-1-4673-1735-1. DOI: 10.1109/IROS.2012.6385518.

[162] Neto, A. B. d. O., Silva, J. A., and Barreto, M. E. "Prototyping and Validating the CORA Ontology: Case Study on a Simulated Reconnaissance Mission". In: *2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)*. Rio Grande, Brazil: IEEE, Oct. 2019, pp. 341–345. ISBN: 978-1-72814-268-5. DOI: 10.1109/LARS-SBR-WRE48964.2019.00066.

[163] Schäfer, P. M. et al. “Flexible Robotic Assembly Based on Ontological Representation of Tasks, Skills, and Resources”. In: *Proceedings of the Eighteenth International Conference on Principles of Knowledge Representation and Reasoning*. Ed. by Erdem, E., Bienvenu, M., and Lakemeyer, G. Hanoii, Vietnam: International Joint Conferences on Artificial Intelligence Organization, 2021, pp. 702–706. ISBN: 978-1-956792-99-7. DOI: 10.24963/kr.2021/73.

[164] Pantano, M., Eiband, T., and Lee, D. “Capability-Based Frameworks for Industrial Robot Skills: A Survey”. In: *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. Mexico City, Mexico: IEEE, Aug. 2022, pp. 2355–2362. ISBN: 978-1-66549-042-9. DOI: 10.1109/CASE49997.2022.9926648.

[165] Quigley, M., Faust, J., Foote, T., and Leibs, J. “ROS: An Open-Source Robot Operating System”. In: vol. 3. ICRA Workshop on Open Source Software. 2009, p. 5.

[166] Huck, T. P., Münch, N., Hornung, L., Ledermann, C., and Wurll, C. “Risk Assessment Tools for Industrial Human-Robot Collaboration: Novel Approaches and Practical Needs”. In: *Safety Science* 141 (2021), p. 105288. ISSN: 09257535. DOI: 10.1016/j.ssci.2021.105288.

[167] Vicentini, F., Askarpour, M., Rossi, M. G., and Mandrioli, D. “Safety Assessment of Collaborative Robotics Through Automated Formal Verification”. In: *IEEE Transactions on Robotics* 36.1 (2020), pp. 42–61. ISSN: 1552-3098. DOI: 10.1109/TRO.2019.2937471.

[168] Gualtieri, L., Rauch, E., and Vidoni, R. “Development and Validation of Guidelines for Safety in Human-Robot Collaborative Assembly Systems”. In: *Computers & Industrial Engineering* 163 (2022), p. 107801. ISSN: 03608352. DOI: 10.1016/j.cie.2021.107801.

[169] García-Domínguez, A., Marcos, M., and Medina, I. “A Comparison of BPMN 2.0 with Other Notations for Manufacturing Processes”. In: AIP Conference Proceedings. Cadiz, Spain: AIP, 2012, pp. 593–600. DOI: 10.1063/1.4707613.

[170] Erasmus, J., Vanderfeesten, I., Traganos, K., and Grefen, P. “The Case for Unified Process Management in Smart Manufacturing”. In: *2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC)*. Stockholm: IEEE, 2018, pp. 218–227. ISBN: 978-1-5386-4139-2. DOI: 10.1109/EDOC.2018.00035.

[171] Schonberger, D., Lindorfer, R., and Froschauer, R. “Modeling Workflows for Industrial Robots Considering Human-Robot-Collaboration”. In: *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. Porto: IEEE, 2018, pp. 400–405. ISBN: 978-1-5386-4829-2. DOI: 10.1109/INDIN.2018.8471999.

[172] Froschauer, R. and Lindorfer, R. “Workflow-Based Programming of Human-Robot Interaction for Collaborative Assembly Stations”. In: (2019). DOI: 10.3217/978-3-85125-663-5-14.

[173] Lindorfer, R., Froschauer, R., and Schwarz, G. “ADAPT - A Decision-Model-Based Approach for Modeling Collaborative Assembly and Manufacturing Tasks”. In: *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. Porto: IEEE, 2018, pp. 559–564. ISBN: 978-1-5386-4829-2. DOI: 10.1109/INDIN.2018.8472064.

[174] Pantano, M. et al. “Novel Approach Using Risk Analysis Component to Continuously Update Collaborative Robotics Applications in the Smart, Connected Factory Model”. In: *Applied Sciences* 12.11 (2022), p. 5639. DOI: 10.3390/app12115639.

[175] Ionescu, T. B. “Leveraging Graphical User Interface Automation for Generic Robot Programming”. In: *Robotics* 10.1 (Dec. 2020), p. 3. ISSN: 2218-6581. DOI: 10.3390/robotics10010003.

[176] Steinmetz, F., Wollschlager, A., and Weitschat, R. “RAZER—A HRI for Visual Task-Level Programming and Intuitive Skill Parameterization”. In: *IEEE Robotics and Automation Letters* 3.3 (July 2018), pp. 1362–1369. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2018.2798300.

[177] Maynard, H. B., Stegemerten, G. J., and Schwab, J. L. *Methods-Time Measurement*. McGraw-Hill, 1948. ISBN: 978-1-258-35098-7.

[178] Paul, R. P. and Nof, S. Y. “Work Methods Measurement—a Comparison between Robot and Human Task Performance”. In: *International Journal of Production Research* 17.3 (May 1979), pp. 277–303. ISSN: 0020-7543, 1366-588X. DOI: 10.1080/00207547908919615.

[179] Zimniewicz, M. “Deliverable 3.2 - SHOP4CF Architecture”. In: (2020), p. 32.

[180] Zimniewicz, M. “Deliverable 3.3 - SHOP4CF Architecture 2”. In: (2021), p. 58.

[181] Cirillo, F. et al. “A Standard-Based Open Source IoT Platform: FIWARE”. In: *IEEE Internet of Things Magazine* 2.3 (Sept. 2019), pp. 12–18. ISSN: 2576-3180, 2576-3199. DOI: 10.1109/IOTM.0001.1800022.

[182] Chitta, S., Sucan, I., and Cousins, S. “MoveIt! [ROS Topics]”. In: *IEEE Robotics & Automation Magazine* 19.1 (Mar. 2012), pp. 18–19. ISSN: 1070-9932. DOI: 10.1109/MRA.2011.2181749.

[183] Pahl, C. “Containerization and the PaaS Cloud”. In: *IEEE Cloud Computing* 2.3 (May 2015), pp. 24–31. ISSN: 2325-6095. DOI: 10.1109/MCC.2015.51.

[184] Brown, T. B. et al. “Language Models Are Few-Shot Learners”. In: (2020). DOI: 10.48550/ARXIV.2005.14165.

[185] Lambrecht, J., Kleinsorge, M., Rosenstrauch, M., and Krüger, J. “Spatial Programming for Industrial Robots through Task Demonstration”. In: *International Journal of Advanced Robotic Systems* 10.5 (May 2013), p. 254. ISSN: 1729-8814, 1729-8814. DOI: 10.5772/55640.

[186] Astad, M. A., Hauan Arbo, M., Grotli, E. I., and Tommy Gravdahl, J. “Vive for Robotics: Rapid Robot Cell Calibration”. In: *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*. Delft, Netherlands: IEEE, Nov. 2019, pp. 151–156. ISBN: 978-1-72813-787-2. DOI: 10.1109/ICCMA46720.2019.8988631.