The International Journal of Advanced Manufacturing Technology (2020) 107:2313–2331
https://doi.org/10.1007/s00170-020-05076-5

ORIGINAL ARTICLE

Methods for considering safety in design of robotics applications featuring human-robot collaboration

José Saenz[1] · Roland Behrens[1] · Erik Schulenburg[1] · Hauke Petersen[1] · Olivier Gibaru[2] · Pedro Neto[3] · Norbert Elkmann[1]

Received: 11 November 2019 / Accepted: 4 February 2020 / Published online: 19 March 2020

Abstract
Collaborative robotics have a large potential for use in industrial applications. Nevertheless, this potential is currently unrealized and one of the reasons is the challenges in planning and designing while considering the safety requirements of these new types of applications. In this article, we will use an exemplary application to describe the many design decisions that are made during the planning of an industrial application featuring human-robot collaboration. Our approach uses model-based systems engineering concepts for considering safety-related aspects of the application during the design phase. Using a software implementation based on our method, we will then compare design results including required floor space and cycle time for the same exemplary application and discuss the implications of our approach for planning other robotics tasks. With our method, the required separation area around the robot was reduced by up to over 66% for a situation featuring a specific robot to be operated at 100% of its maximum possible speed.

Keywords Safety · Robotics · Collaborative

1 Introduction

There is a large industrial demand for new methods for designing robotics applications featuring human-robot collaboration (HRC applications). The current methods for planning HRC applications follow a nonlinear and highly iterative process with frequent unexpected drawbacks [17]. As a result, the complete systems are often costly and require large amounts of time to design, build, and validate. Furthermore, uncertainty during the design phase often leads to situations where a prototype is first built and validated in order to arrive at realistic estimates for the process parameters, leading to additional costs.

The design process, starting from the first sketch and finishing with the application running, needs a lot of effort in order to meet the safety requirements as specified by the existing health and safety regulations and standards [1, 19]. The legally relevant health and safety requirements in Europe are defined in the Machinery Directive 2006/42/EC [31], as well as various standards. The most relevant for industrial collaborative robotics include the ISO 12100 [7] and the ISO 10218-1 [2] and ISO 10218-2 [3], as well as the technical specification ISO/TS 15066 [6]. In our experience, system integrators and end-users of collaborative robots desire a more fluent and tool-based design process that supports the consideration of safety aspects from within the CAD planning environment. The interdependencies between the human operators, the technological components (robot, tools, and sensors), the application (the process, the environment, the role of humans), and the safety standards span a high-degree and complex solution space. There are currently no tools available at all which reduce the current complexity and allow for a streamlined design approach.

After an introduction to the state of the art, we will describe an exemplary application, highlighting how design decisions are currently made in engineering practice. In particular, we will highlight critical safety challenges and the flow of information throughout the design process. This includes understanding where safety-related information is sourced, which expertise is required for a particular step, and what software is used for storing and processing

✉ José Saenz
Jose.Saenz@iff.fraunhofer.de

[1] Fraunhofer IFF, Magdeburg, Germany

[2] ENSAM ParisTech, Lille, France

[3] CEMMPRE, University of Coimbra, Coimbra, Portugal

information. We will then introduce our approach for considering the safety of HRC applications, which incorporates model-based systems engineering concepts and is compatible with CAD/simulation tools in use today. Using the same example provided, we will compare results from our proposed approach with the current techniques for safety-related parameters such as size of minimum required separation distance and cycle time.

The main contributions of this work are twofold. From a theoretical standpoint, the main contribution is the proposed integrated methodology for considering safety-related features during the design of HRC applications featuring speed and separation monitoring. From a practical point of view, the contributions are on modeling safety systems in typical robotics simulation environments, and a streamlined workflow including the decision-making process for designing HRC systems.

2 State of the art—planning industrial applications featuring human-robot collaboration

Safety applied to collaborative robotics has been an active research topic in the last decades. In fact, it is impossible to imagine humans and robots sharing the same workspace and physically interacting without guaranteeing the physical integrity of nearby humans. In this context, the safety systems should be reliable to minimize the risk for both humans and equipment. In [28], a safety methodology for the control of collaborative robots combining design approaches for hardware and software of safety-related systems is proposed. Experiments demonstrated the effectiveness of the proposed methodology on a toilet-assist robot. Metrics are increasingly important in robotics, especially in relation to safety. In a recent study, a design metric based on maximum power flux density is proposed for the assessment of the severity of a transient physical contact between a robot manipulator and a human body region [29]. A power- and force-limiting type of collaborative application was considered to assist the design of both the robot manipulators and the application. In [30], robot motion generation algorithms for human-robot shared environments were investigated. A kinematic control strategy is proposed to enforce safety while maintaining robot productivity. Such methodology was experimentally validated on a dual-arm robot with 7-DOF per arm performing a manipulation task.

The planning of industrial HRC applications involves a large number of individual tasks and is traditionally executed by a team with different areas of expertise. Using a modified life cycle model in Fig. 1 that was first proposed in [21] for industrial HRC applications, we can specify that when we refer to "planning," we mean the work that takes place during the concept and development phases.

Figure 2 illustrates the general steps involved in the planning of HRC applications during the life cycle phases of concept and development, derived from the Machinery Directive 2006/42/EC. A good overview of this process and the role that safety plays is described in [23]. For our purposes, the starting point is a general idea of collaborative application, meaning that the tasks within the application is well known, and that there is a general idea of how the robot should assist the human operator (as described later in Section 3). This also means that an initial choice for the type of robot (type, payload, size, position) has been made, as well as an initial layout plan for how the robot should be positioned in the environment. Previous work [8, 21] describes a decision-making framework for generating the layout of an HRC application and allocating tasks between robots and humans, with the aim in reducing the time and effort for generating a process plan. While these questions are valid and the methods they propose are sound, their work is less relevant for the safety-related considerations of the layout. Other research in this field [20] focuses on describing robot motions as small building blocks and then combining these for planning assembly tasks. While this work provides an interesting overview of the state of the art for robot assembly planning, this approach does not consider safety in the sense of ISO 10218-1 and ISO 10218-2 and is only peripherally relevant for this work.

Following this initial concept of the application, the concrete details of the design are cross-referenced to ensure that the general and essential requirements are met. As an example, the designer checks that all safety-related components have performance level "d," category 3 according to ISO 13849 [4]. After the general requirements on the robotic system have been fulfilled, the designer can detail the process further. This includes a specification of which tasks are to be done by the robot and the human, what form of HRC is envisioned, and what corresponding safeguarding mode will be used for different steps. In the next step, the system limits need to be defined. This includes a specification of the restricted areas for the robot, as well as maximum robot speeds. At this point, it is possible to review

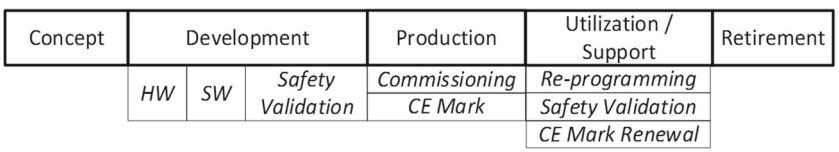

Fig. 1 Life cycle model for industrial HRC application, based on generic life cycle model from ISO/IEC/IEEE15288:2015

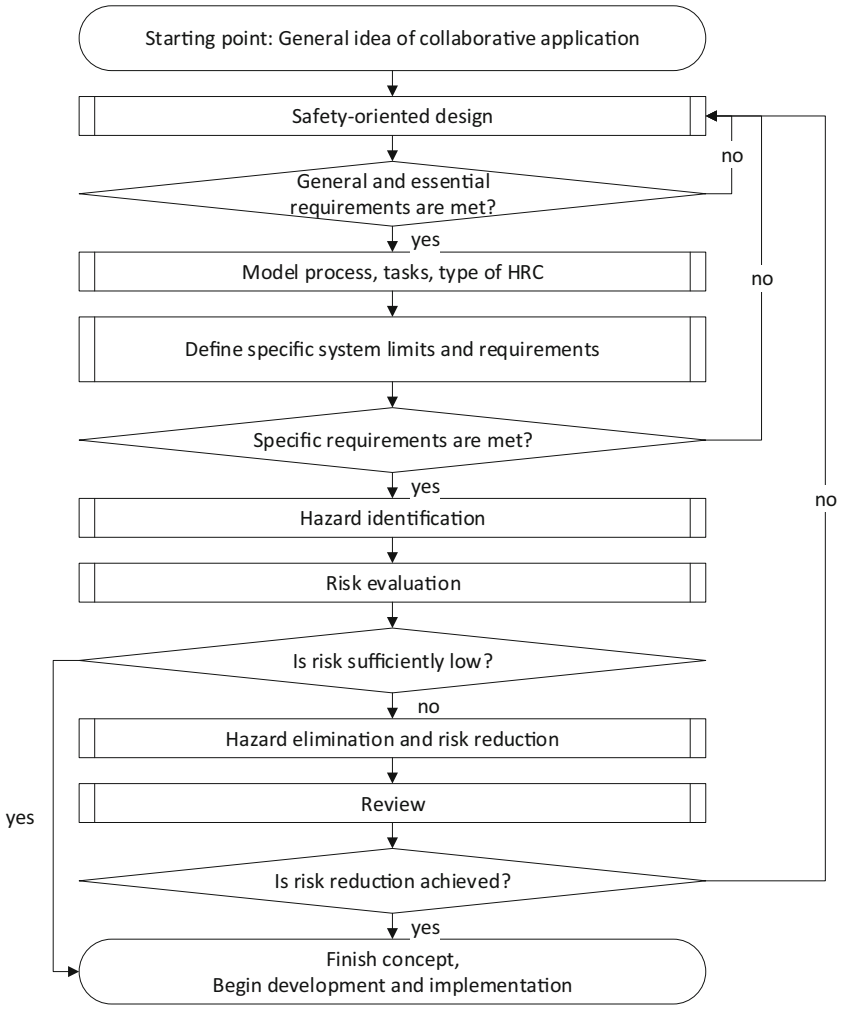

Fig. 2 Flow model of different phases during concept and design of a HRC application in manufacturing according to Machinery Directive 2006/42/EC

key economic parameters such as cycle time analysis. After reviewing the requirements, a hazard identification and risk evaluation are carried out. By this time, the design is quite advanced and it is possible to identify real, specific hazards.

Recent research efforts have focused on the use of model-based engineering methods to carry out the risk analysis for collaborative robotics applications [24]. Other work in this direction has either focused on use of the Failure Modes, Effects and Criticality Analysis (FMECA) technique [11], or the HAZard Operability (HAZOP) technique [10]. Unfortunately, these approaches do not address issues such as the size of safety zones and the overall effect of safety requirements on the environment, on the type of interaction, and on the overall process. Furthermore, these approaches do not sufficiently address the concept of requirements engineering to ensure conformity with the collaborative robotics standards ISO 10218-1 and 10218-2, as well as the ISO/TS 15066.

If remaining risks are too high, the designer can implement additional hazard elimination and risk-reduction measures. These can include additional safety sensors, changes to the process and/or environment. Here, quite a lot of work has focused on applying different safeguarding techniques [26], and on methods for calculating the required safety distances [9, 14, 16, 22]. Due to the sheer number of options, the designer has at their disposal to adapt an HRC application, there are a large number of possible variations for a given application. Furthermore, as indicated in the flow chart, if the final system still poses significant safety risks to the humans, the designer has to start all over again. The concept is only complete once the risk has been sufficiently reduced, opening the path for further development for implementation.

Systems engineering methodology has been applied to a number of industries and to an extent is currently being applied for specific questions in robotics, most notably for software engineering [18]. However, it has not yet been applied to the issue of safety during the lifecycle phases of concept, design, and validation. To our knowledge, any software tools that support the safety analysis of machinery are completely independent from other traditional design tools such as CAD/simulation tools. The use of software tools to

support safety analysis within CAD/CAE systems has not been described in the literature for HRC applications. In the literature, there have been instances where either ontologies or system models have been used to derive the risks of a system for a risk analysis [25] or for the purpose of carrying out a Failure Mode and Effect Analysis (FMEA) [27]. However, we have also not seen any instances where the configuration of safety sensors was used in simulations to perform analysis of safety-related design issues.

In the following sections, we would like to describe how all these individual tasks are carried out in engineering today in order to establish a baseline against which we can compare our approach. In order to do this, we will first introduce an exemplary application. Then, we will continue with an explanation of the current practice for designing the application.

3 Planning human-robot collaboration cells

In order to present our approach for considering safety when designing HRC applications, we will introduce an exemplary use-case that is currently performed manually, and which should be carried out with collaborative robots. The application has been derived from a real use-case from the automotive sector. Specific information regarding part geometry has been changed to protect confidential, proprietary information. These changes however are only cosmetic and do not affect the relevant properties of the application or the HRC solutions.

3.1 Depalletizing application

The task used here focuses on the manual assembly of brackets, as part of a pre-assembly station (not a part of the moving line). The brackets are delivered to the station in a wooden pallet and are stacked three layers deep in the pallets.

The operator walks from their assembly table to the pallet whenever a bracket is needed. The operator removes the brackets from the pallet, walks back to their assembly tables, and mounts fasteners to the bracket. When the pre-assembly tasks are finished, they place the finished part in a separate carrier, where it is picked up by logistics operators. This application is ergonomically challenging for operators, especially the task of reaching deep into the pallet to extract brackets from the lower layers.

In a first instance, manufacturing engineers would like to determine whether robots could be used for specific tasks and work alongside the human operators in this application. The general idea is for a robot to take over the task of removing the parts from the pallet and either directly handing them to the operator or placing them on the table where the operators can take them and perform the next assembly tasks.

The brackets weigh 7 kg each and are made of steel and coated in paint. The pallets are 1.2 m × 0.80 m and are 0.90 m deep. The parts within a layer are separated by rectangular foam pieces, which are laid between the parts.

Logistics operators bring the pallets into the workspace using a forklift. In normal operation, two pallets are sufficient for a single 8-h shift. Therefore, the pallets are brought into the workspace at the beginning of each shift and the empty ones removed at the end.

There are three separate workstations where operators can assembly the brackets. These are all equidistant from the two pallets that are positioned next to each other, in Fig. 3. The pallets are approached from one side by the operators. The rear side is the area for forklifts and logistics processes. The pallets are removed by forklift from this side.

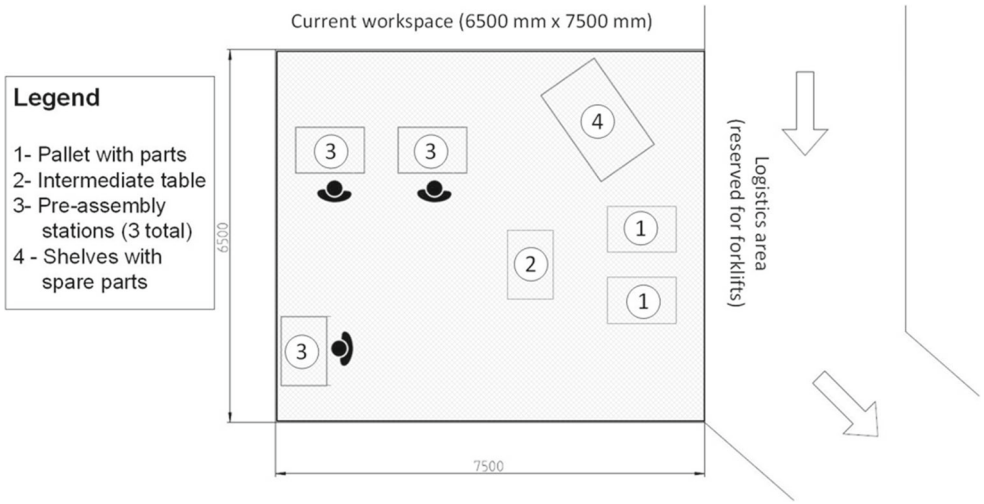

Fig. 3 General layout of the current station with logistics area to the right side of the assembly task floor area

3.2 Planning process with current methods

Building upon the generic flow model from Fig. 2, we will describe how the design process can be carried out with methodology and engineering tools available today. We use the following terms to identify the stakeholders involved in the design process:

1. Designer
2. Safety expert

The term designer refers to the engineer(s) tasked with planning the robotics application. They can be process, electrical, or mechanical engineers by training and are concerned with all aspects of the robotic application. This includes from the mechanical side of the layout planning, the mechanical design including tooling, and the overall process planning. Relevant electrical- and control-related aspects to be considered include:

1. The overall electrical plan of system
2. e-Stop functionality (stop/ restart)
3. Choice of electrical components
4. Integration into PLC or robot control system
5. Programming (robot control and PLC program if necessary)
6. Choice of electric and electronic components including sensors for process control and safety

The safety expert is responsible for ensuring overall safety of the system, with particular view towards human occupational health and safety. Therefore, they are concerned with:

1. The ergonomics of all tasks
2. The hazard identification and risk analysis
3. Approval of risk mitigation measures, in collaboration with the designers

We see a need that they are consulted by the designer (in the best case) at specific steps during the process of designing the application featuring HRC. In the worst-case, they are consulted at the end of the design process to review the status and offer suggestions for changes.

In the description of the design process, we will also make references to the types of software and/or documentation tools that the stakeholders use along the different phases of the design process.

3.2.1 Starting point: general idea of collaborative application

As a starting point, we assume that the designer has decided to use a standard industrial robot with a maximum payload of 22 kg and a maximum reach of at least 1.6 m (KUKA KR22-1612).

The designer starts with a CAD design and layout of the application. The designer initially thinks that Safety-Rated Monitored Stop (SRMS) or Speed and Separation Monitoring (SSM) is the correct method to safeguard the robot, since the parts have sharp edges that should not come in contact with a person.

The designer creates an Excel spreadsheet of the robot speeds for the individual movements, to get a general idea of the cycle time. Based on previous experience, the planner reviews the cycle times to see if they seem reasonable, and uses this first estimate to determine whether it is economical to continue with the design of the application.

3.2.2 Safety-oriented design

The designer starts by reviewing the Machinery Directive 2006/42/EC, and then reviews the type-C standards ISO 10218-1 and ISO 10218-2 to look for general requirements on the system. Since the designer has to do this often, they might have even created their own checklist to support them in this process.

They first review the robot system and look at the control system, the control devices, and the emergency stop functionalities as specified in the ISO 10218-1. The designer often simply assumes that the industrial robot will automatically fulfill all of these requirements if it is sold with a Declaration of Incorporation according to the Machinery Directive.

They then review the design of the work cell, looking at the ergonomics, lighting, and overall mechanics. This is by definition only a preliminary assessment, and will need to be revisited. Examples here include the requirement that the design avoids areas of shadow, irritating dazzle, or stroboscopic effects on moving parts.

3.2.3 General and essential requirements met

Here, the designer has also created an internal checklist, which is derived from the Machinery Directive 2006/42/EC and the ISO 10218-1. Many of these need to be revisited, as they are very concrete (e.g., requirements on labels and signs to indicate the robot's mode of operation) and cannot be fulfilled early in the design process. The most important general requirement regards the performance level of the robot used. Either the control system of the robot (and the corresponding safety functions such as motion, speed, and/or force control) is safety rated and certified to have performance level "d," category 3, or the risk assessment needs to prove that a lower category is possible. As a general rule of thumb, the designer normally starts all designs with a robot that has a performance level "d," category 3 safety rating.

3.2.4 Model process and assign tasks

In this step, the designer creates a simple listing of the tasks (Table 1) that are required for the process. In this case, individual tasks include movement to specific places in the workspace (e.g., table, pallet), as well as physical manipulation (e.g., picking and placing parts). Ideally, the designer considers different tasks along the entire system lifecycle, from commissioning to productive operation and maintenance tasks. For the purposes of this paper, we have limited ourselves to two separate tasks within the lifecycle of productive operation.

Using the methodology described by [26], we can also identify the type of HRC that we would like to use. As described in that work, this decision is based on the available and meaningful types of collaboration that suit the task. In this case, the designer chooses Speed and Separation Monitoring (SSM).

3.2.5 Model process and assign tasks

Based upon the floor plan and the task models previously defined, the designer specifies the system application limits and requirements. The limits include:

1. The workspace of the robot
2. The configuration of the robot to reach all the critical positions within the workspace
3. The speeds to be reached
4. The payloads carried

At the conclusion of this step, the designer checks that the system limits and requirements (e.g., for the specific safeguarding mode) are fulfilled. If not, the designer needs to change specific aspects of the design. Any changes made during this phase due to non-conformity to a specific requirement means that the designer has to start the process again from the beginning before continuing.

3.2.6 Hazard identification and risk evaluation

The application designer works together with a safety expert to identify hazards. They use the individual steps from the task model, and identify hazards per task step. Table 2 shows the specific risks identified.

The safety expert carries out the risk evaluation. In general, risk is the product of the severity of the damage and the probability of occurrence, and there are several methods for a quantitative evaluation of the risk [7]. In this example, the risk evaluation shows that there is a risk of collision and clamping during a large number of individual steps in the overall process and that risk mitigation methods need to be applied to them. Following the hierarchy of risk mitigation measures, technical safeguards were chosen since an inherently safe construction for the application is not feasible.

Table 1 Tabular listing of process tasks for two exemplary processes: standard operation (S ID) and operator removal of packing material from the pallet (P ID)

ID Nr	Process tasks	Task assignment		Task characteristics		
	Standard Operation	Operator	Robot	Shared workspace	Simultaneous co-work	Physical contact
S1	Identify part in pallet		x	Yes	No	No
S2	Move to pallet		x	Yes	No	No
S3	Pick up part from pallet		x	Yes	No	No
S4	Move to table		x	Yes	No	No
S5	Place part on the table		x	Yes	No	No
S6	Move to neutral position		x	Yes	No	No
S7	Enter collaborative workspace (table)	x		Yes	No	No
S8	Pick up part from table	x		Yes	No	No
S9	Leave collaborative workspace		x	Yes	No	No
	Operator removal of packing material					
P1	Identify that packing material needs to be removed, send operator signal		x	Yes	No	No
P2	Move to neutral position and stop		x	Yes	No	No
P3	Enter collaborative workspace near pallet	x		Yes	No	No
P4	Remove packing materials and/or separating layer	x		Yes	No	No
P5	Leave collaborative workspace	x		Yes	No	No

Table 2 Tabular listing of process tasks and associated hazards

ID Nr	Process tasks	Hazards
	Standard operation	
S1	Identify part in pallet	None
S2	Move to pallet	Collision
S3	Pick up part from pallet	Collision, clamping, crushing
S4	Move to table	Collision
S5	Place part on the table	Collision, clamping, crushing
S6	Move to neutral position	Collision
S7	Enter collaborative workspace (table)	Collision
S8	Pick up part from table	Collision
S9	Leave collaborative workspace	None
	Packing Material support	
P1	Identify that packing material needs to be removed, send operator signal	None
P2	Move to neutral position and stop	Collision
P3	Enter collaborative workspace near pallet	Collision, clamping, crushing
P4	Remove packing materials and/or separating layer	Collision, clamping, crushing
P5	Leave collaborative workspace	Collision, clamping, crushing

The S ID indicates a standard operation and the P ID indicates packing material support

3.2.7 Hazard elimination and risk mitigation

The risk mitigation process begins with the safety expert analyzing the results of the risk evaluation and specifying where changes need to be made. In this step, the safety expert also approves the use of SSM as the safeguarding mode. Following this, the designer uses the assumptions for the robot speed, the payload, and the extension to determine the braking time and the braking distance from manufacturer data sheets. The designer uses these, as well as initial assumptions from the risk mitigation measures (e.g., safety sensors), to define the size of the minimum required safety zone. In this step, there are also quite a few important configuration settings from the safety sensors that need to be defined or assumed in order to estimate the correct safety zones. These configuration settings are discussed with the electrical engineers, as these play a role in the electrical planning (e.g., which type of bus system is used). The separation distance is usually calculated with software outside of the design tool, for example, in a spreadsheet, using values taken from a robot data sheet [16, 22]. The equations for the calculation of the separation distance are described in Section 3. The calculation here represents a worst-case situation for a single, discreet action, and is then applied this over the boundary conditions of the application. A more nuanced study, considering dynamic effects, the changes in trajectory over time, etc., is not carried out due to lack of proper engineering tools.

The protective zone is represented by a combination of circles with the radius, $r = S_p$, which are centered at the extreme outer positions that the robot reaches. In this case, the designer chose three extreme positions, namely the furthest corners of the two pallets and on the table where the parts are laid, as in Fig. 4. The safety zone is represented by the combined area within the three circles.

Given the parameters in Table 3, and using a horizontally oriented laser scanner as the safety sensor, a minimum separation distance of 2.98 m is required around the robot tool at all times. Figure 4 represents the floor space that needs to be safeguarded by the laser scanners. Given the dimensions between the three points representing the furthest reach, the total area to be safeguarded can be calculated to be over 50 m^2.

3.2.8 Review

In this step, the designer reviews the final safety concept and validates the solutions against the requirements on the system. Figure 5 shows the layout with the calculated required minimum separation distance with the initial configuration. The designer quickly sees that this configuration is not acceptable, as the required safety distance is too large for the given workspace. We see a clear overlap of the safety distance and the logistics area why passing forklifts and operators working at assembly Table 1 or accessing shelves will cause the system to stop.

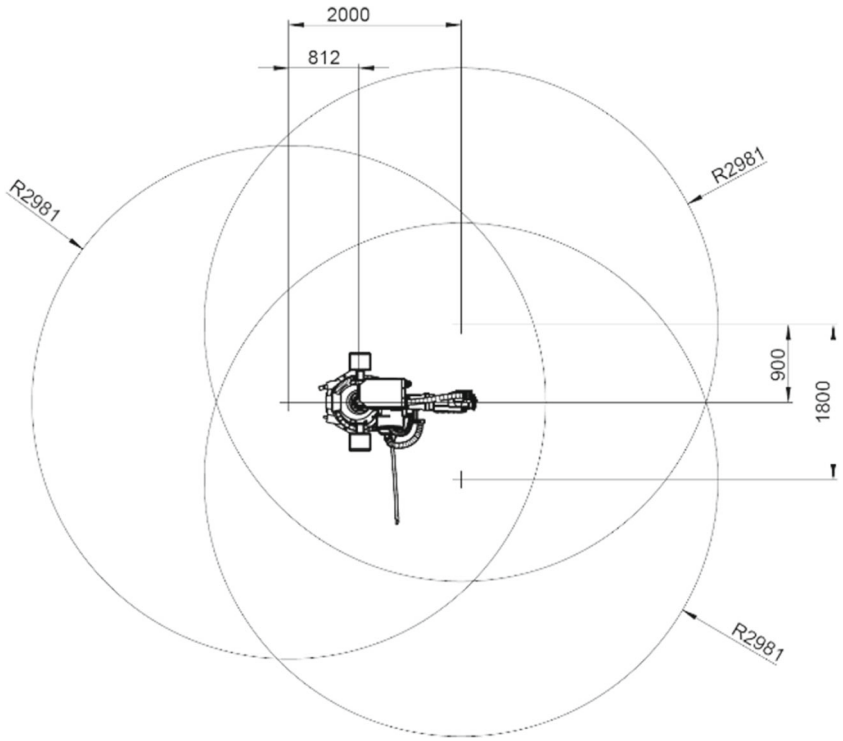

Fig. 4 Layout with robot with minimum required safety distance (R = 2918 mm) at three main positions for picking and placing with a KUKA KR22 robot operated at maximum speed and using two horizontally oriented laser scanners with a reaction time of 90 ms and a C-value of 850 mm

4 Proposal of methodology for design of collaborative robotics applications

The proposed approach applies model-based systems engineering practice to the issue of how to consider safety during the process of designing HRC applications. We therefore start with the specification of the requirements before defining an architecture for the new system. Then, we design the individual models we will need that physically describe not only the components but also the aspects related to safety (e.g., including attributes such as performance characteristics and sensor configuration). We will only briefly describe the modeling process for a single safety sensor, first clearly defining the scope of the modeling effort before describing the major attributes of the model and its connection to the overall system.

4.1 Goals of new methodology

As a first step in model-based systems engineering, we start by clearly defining the requirements on the system, in our case, a tool to support the safety analysis when designing HRC applications. Using the current design method for designing HRC applications as a starting point,

Table 3 Parameters used to determine size of minimum protective area for a discreet robot position

Parameter	Value	Comments
v_h	1600 mm/s	Standard assumption when not possible to measure the speed of approaching humans
v_r	1000 mm/s	Maximum robot speed
T_r	90 ms	Sensor reaction time from manufacturer data sheet
T_s	400 ms	From manufacturer data sheet
S_h	784 mm	Contribution to safety zone due to approaching human
S_r	90 mm	Contribution to safety zone due to robot speed
S_s	1242 mm	Extrapolated from manufacturer data sheet assuming combined axis 2 and 3 movement, 100% arm extension, 66% payload and 100% of maximum speed
C	850 mm	Assuming a horizontal orientation of the laser scanner
Z_d	10 mm	Measurement uncertainty from sensor system
Z_r	5 mm	Positioning uncertainty of robot
S_p	2981 mm	Necessary protective distance for SSM

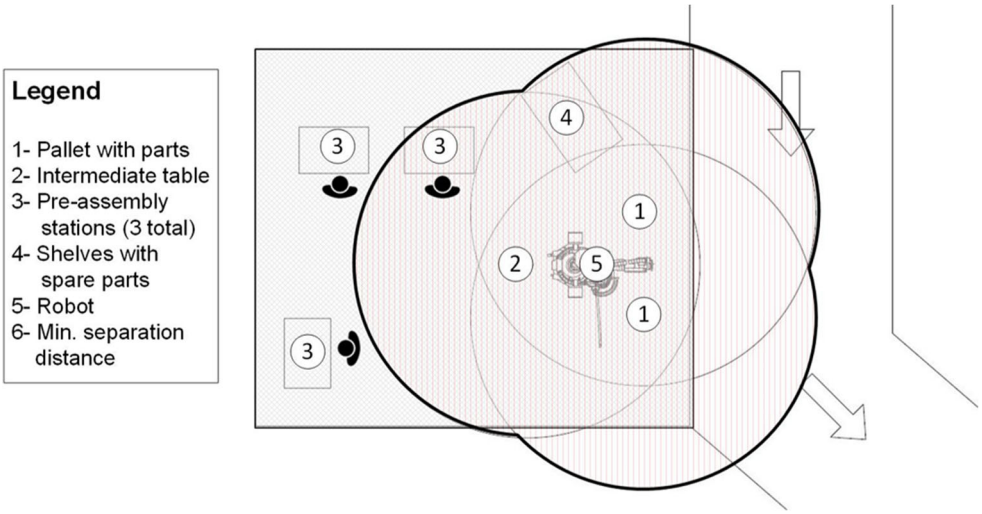

Fig. 5 Review of the application with a robot and the calculated minimum separation distances using traditional worst-case calculations. The required safety areas reach well into the logistics area, where forklifts pass by

a number of high-level requirements for an engineering tool and/or approach to streamline the design process have been formulated:

1. Requirements tracking to verify that the design fulfills the specified and explicitly formulated requirements
2. Support for consideration of safety-related questions (according to ISO 10218-1, ISO 10218-2, and ISO/TS 15066) in addition to other related standards and directives
3. Support for what-if analyses covering all aspects of an HRC application to improve/optimize designs and understand how changing parameters (e.g., robot speed)
4. Support for data round-tripping so that system models can be used together with existing engineering tools such as CAD and simulation software

4.2 Design targets

In order to compare the outcomes of the traditional methodology for designing HRC applications, we introduce a few design targets that a planner will typically seek to achieve. In general, it is essential that the design maintain worker safety at all times, as well as a set of production-oriented goals. In the literature, it is not always entirely clear what is meant by being safe. In some instances, it is implied that the absence of an injury under controlled experimental circumstances constitutes proof of safety. For the purposes of this paper, ensuring safety means that the system conforms to the relevant laws, guidelines, and standards. This means that a risk analysis according to ISO 12100 has been carried out, that risk mitigation measures have been identified, that the robotics components meet the requirements from ISO 10218-1, and the complete robotic system the requirements from ISO 10218-2 and the ISO/TS 15066. From the perspective of the designer, there is a specific set of production-oriented design targets that can be used to calculate a return-of-investment (ROI) for the system. This ROI can be used to evaluate whether a specific design will be implemented or whether the investigation will be stopped. In many cases, the most important reason for investigating a collaborative robotics solution is to create a significant improvement in operator ergonomics [15]. In these cases, the level of improvement to worker ergonomics can also be a deciding factor, beyond pure ROI considerations. The following key performance indicators are especially relevant for HRC applications:

1. Costs – The system costs are often considered across the entire lifecycle, from design to implementation (e.g., hardware costs, engineering and installation costs), operating costs, maintenance costs, and de-commissioning costs. The designer has the ability to influence these costs through selection of type of HRC, sensor, and robot choice.
2. Floor space – The required floor space is important, both because the overall flow of materials, transport spaces, and human spaces place strong constraints on how much space can be used, and because floor space is indirectly associated with a specific overall cost.
3. Cycle time – The achievable cycle time of an application is important both because it has a direct relation to the costs and overall production constraints (i.e., needs to fit into overall production cycle).

It is important to note that the priority for an individual production goal is very specific to the application in question and can vary. This implies that these production goals are part of the design space for the planner and the system design can be strongly influenced by which of these goals has the highest priority.

4.3 Architecture specification

In order to reach the goals specified in the previous sections, we propose a simple architecture that allows for data exchange with a CAD and simulation software tool. The engineering tool that we propose builds upon the physical (static and dynamic) models that already exist in the CAD and simulation software. We call it the Computer-Aided Safety (CAS) Tool.

The sequence diagram in Fig. 6 shows a part of the generic workflow that the designer uses when designing HRC applications and the software tools that are used. An important aspect is that the designer only accesses the CAS Tool through the CAD/simulation environment. They do not need to learn new software tools, and our system does not seek to duplicate readily available tools. Indeed, a key aspect from the architectural standpoint is, based on the normal and well-established workflow as described in Section 2, finding a way to build on existing tools and reduce situations where digital information is lost.

The CAS Tool proposed here therefore collects all relevant information needed for the safety-related evaluation and dimensioning of components initially from the CAD/Simulation tool. This includes model information about the robot types, their position in the layout, the physical parameters, and importantly information about the process through the program and the physical information (e.g., tooling and part information). It then appends these models with safety-related information that is made available through a number of sources, but remains firmly in the background. Wizards and input masks (Fig. 7) are used to gather safety-related information from the designer that is not readily available from data sheets (that are in a database within the CAS Tool).

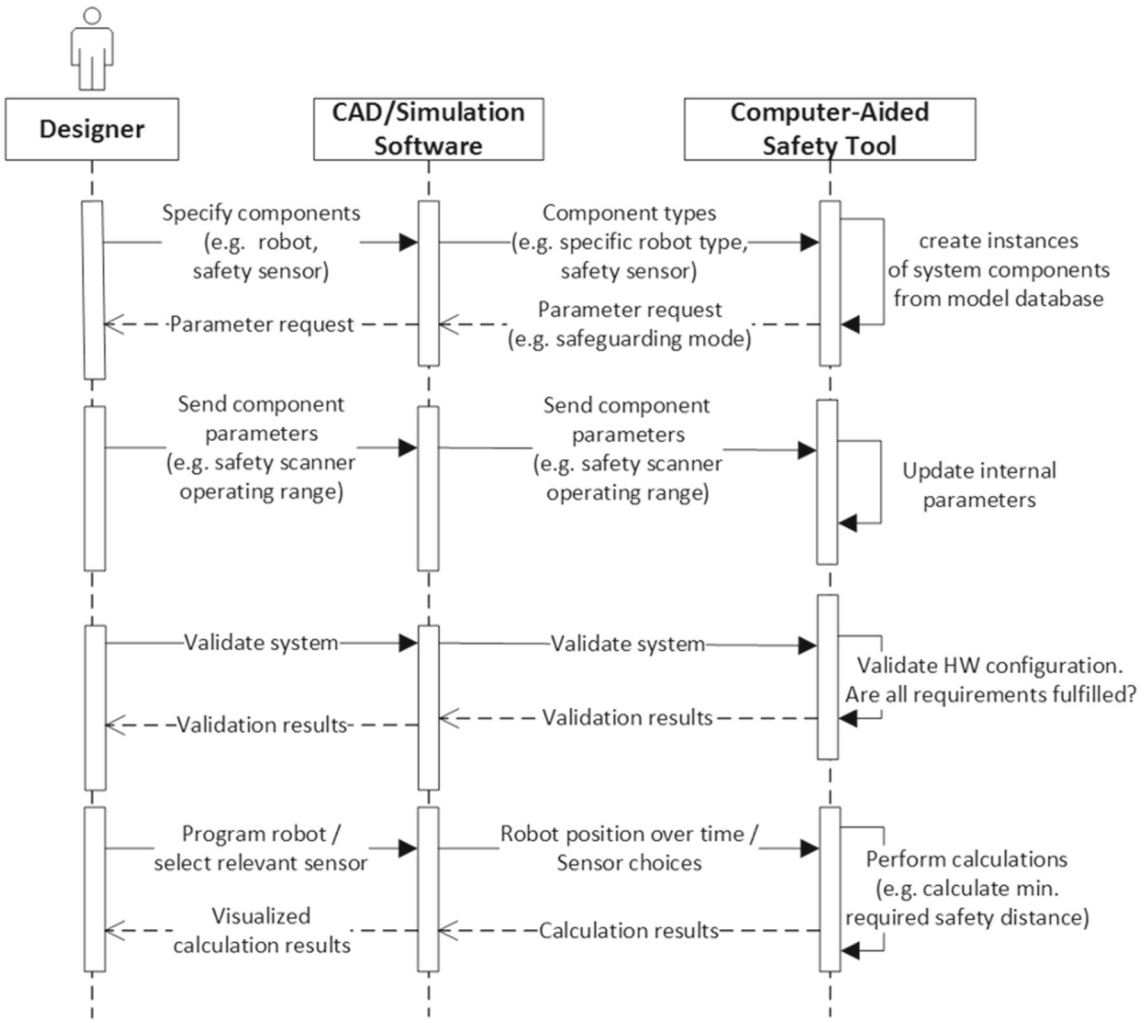

Fig. 6 Sequence diagram of the workflow of the designer using existing CAD/Simulation software tools and the proposed Computer-Aided Safety tool

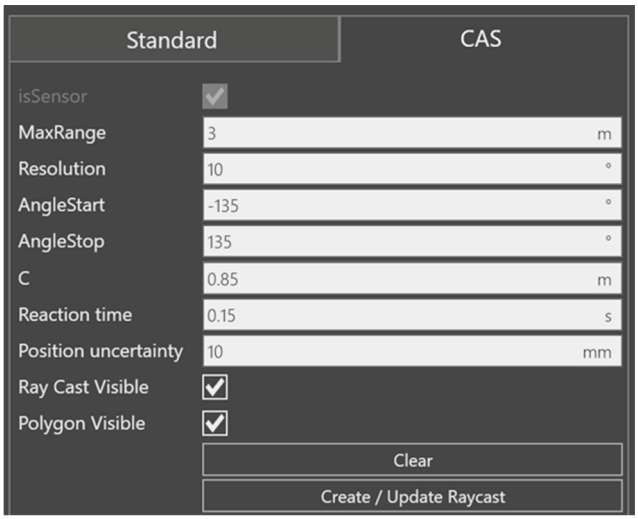

Fig. 7 Exemplary input mask for configuration of a generic laser scanner

4.4 Definition of system models

In this section, we will briefly discuss the system models that are used as the basis for the CAS Tool. As a first step, we created a simple model of all the components involved in a collaborative robotics application. We build on existing models that are available either through robotics ontologies [13] or through other model-based software activities [18], and extended to include the information relevant for the safety evaluation and calculation of required separation distances. As shown in the previous section, the overall architecture is designed to use model information already available from a CAD/simulation software and append it with safety information. The key system characteristics that we want to study and understand are:

1. What risk mitigation measures are possible and valid?
2. What is the required minimum separation distance for a specific safety sensor and for the chosen system configuration?
3. Do all chosen sensor and robot parameters (e.g., sensor resolution, sensor reaction time) fulfill the requirements of the specific application?
4. Is the configuration and position of my sensor valid to monitor the required safety zone?
5. How large are the (instantaneous) safety zones and how do process parameters (payload, robot program, speeds) that the designer can change influence their size?

We will address these questions individually in the following sections, highlighting how the model is appended to include the relevant information to answer these questions.

4.4.1 Validity of risk mitigation measures

As a first step, we would like for the designer to be able to choose from a pool of safety sensors that have been shown to be valid for use in safety-related applications. The simplest means to determine whether a specific sensor is valid for use is to check whether the sensor has the required performance level according to ISO 13849 or the required safety integrity level (SIL) according to IEC 61508 [12]. Therefore, the first model parameters to be included for any safety sensor in our library of components are for these. The values for these parameters can be appended manually to our library of components or they can be read from an e-data sheet provided by the manufacturer. With this information in the model, the designer can run a simple check when using a component from our library whether the component has the required performance level and category or SIL. The required value is derived from the risk analysis, and the default required value would be performance level "d," category "3." While this initial check may be considered trivial, it requires the user to define a real, specific sensor type (manufacturer, model), which is relevant for further evaluation.

4.4.2 Definition of required minimum separation distance

As a next step, we enable the designer to determine the required minimum separation distances based upon the operating parameters for the process. This would be a sharp contrast to the methodology currently in use, which relies on a discreet worst-case situation and applies it across the entire process.

The main equations to consider are from the ISO/TS 15066 with regard to speed and separation monitoring. In particular, the main equation for determining the minimum required separation distance, $S_p(t)$, can be described as the sum of a number of individual terms in the following equation.

$$S_p(t) = S_h + S_r + S_s + C + Z_d + Z_r \quad (1)$$

Previous work explains well the individual terms and how they are determined [16, 17]. In our case, we are especially interested in how the choice of specific sensors plays a role in the calculation of this value and how to model the sensors for use in our overall methodology. As an example, we will discuss a generic laser scanner and then explain how the parameters and relations can vary when considering a specific product.

The first term in the equation, S_h, is related to the distance that a human can move during the time it takes for the sensor to react and for the robot to stop. In the absence of a system to measure the speed and direction of human motion, we can use a constant value of 1600 mm/s for approach

speed of the human as defined in the ISO 13855 [5]. In this case, the separation distance attributable to the speed of the approaching human, can be calculated as follows:

$$S_h = 1600(T_r + T_s) \tag{2}$$

whereby T_r the reaction time of the sensor and T_s represents the braking time of the robot. The next term in (1) is the distance attributable to the robot's motion after an object has entered the safety zone up to the point where the emergency stop command is sent to the robot:

$$S_r\,(t) = \int_{t_0}^{t_0+T_r} v_r\,(t)\,dt. \tag{3}$$

In the case of a constant velocity, (3) can be simplified to:

$$S_r\,(t) = T_r v_r. \tag{4}$$

The third main component of the minimum protective distance S_S represents the braking distance of the robot. This distance can be calculated from manufacturer data sheets, and is dependent on the payload, the configuration of the arm, and the robot's speed. An example of this calculation is provided in [16].

The value C is the overreach constant and indicated how far a body part can penetrate the safety area before it is sensed. This value is defined in the ISO 13855 and is primarily determined by the type of sensor that is used. In the case of a laser scanner, the value is also influenced by the orientation of the laser scanner, θ_{LS}, the height at which the sensor has been mounted, H_{LS}, and the resolution of the sensor, d_{LS} whereby $\theta_{LS} = 0°$ corresponds to the laser scanner in a horizontal position, with the scanning field parallel to the floor, and $\theta_{LS} = 90°$ corresponds to the laser scanner in a vertical position, with the scanning field perpendicular to the floor.

$$C_{\text{scanner}} = \begin{cases} 1200 - (0.4H_{LS}), & \theta_{LS} = 0° \\ 8\,(d_{LS} - 14), & \theta_{LS} = 90° \end{cases} \tag{5}$$

In the case where $\theta_{LS} = 0°$, it is also important to note that the minimum allowed value for C_{scanner} is 850 mm. Combining (2), (4), (5), and (1), we see that the following sensor parameters:

1. Sensor reaction time T_r
2. Sensor resolution d_{LS}
3. Height of the sensor H_{LS}
4. Orientation of the sensor θ_{LS}

All these parameters play a role in determining the required size of the separation distance.

$$S_p\,(t) = \begin{cases} 1.6\,(T_r + T_s) + (T_r v_r) + S_s + (1200 - (0.4H_{LS})) \\ \qquad +Z_d + Z_r, & \theta_{LS} = 0° \\ 1.6\,(T_r + T_s) + (T_r v_r) + S_s + (8\,(d_{LS} - 14)) \\ \qquad +Z_d + Z_r, & \theta_{LS} = 90° \end{cases} \tag{6}$$

As a first instance, it is possible to model generic sensors featuring these key safety-related attributes. To support real world implementation, specific models of individual sensors are needed and further configuration parameters such as multi-sampling or constraints between reaction time, sensor resolution, and scanning range also need to be included in the model. In this case, we also see that we have static attributes (e.g., those that can be derived from a data sheet and that are not independent of a user-specific configuration or setting) and dynamic attributes that are chosen by the user.

As an example, the SICK laser scanner microScan3 requires users to specify a number of configuration parameters to determine the reaction time, T_r, of the sensor. In particular, the following parameters are required to determine the sensor's reaction time:

1. Scan cycle time T_s
2. Set interference protection T_i
3. Set multi-sampling n
4. Time for processing and output T_o

$$T_r = (T_s + T_i)\,n + T_o \tag{7}$$

The values for the scan cycle time are themselves determined from a look-up table and are dependent on the range and resolution of the sensor.

$$T_s = f\,(d, r) \tag{8}$$

This relationship puts designers in a conundrum, as they are required to make assumptions for the range without knowledge of the size of the safety zone they need to oversee, which is the outcome of Equation 5.1 and for which the range is an input. Therefore, some means for checking the validity of the designer's assumptions is also necessary, and will be addressed in Section 4.4.3.

The authors would like to point out how the determination of specific sensor configuration parameters in the design phase essentially front-loads some of the overall engineering work. The settings for a safety laser scanner are normally set during the commissioning phase of a system. Using this methodology, the scanner configuration necessarily needs to be determined during the design phase in order to fully understand the effect that configuration has on the safety-related parameters. This methodology has the added advantage, if implemented correctly, that the configuration

only needs to be determined one time during the design phase, and the used values can be exported to the sensor later during the commissioning phase. Indeed, this method would not only streamline the overall workflow, but it would be necessary to ensure that the real system as built conforms to the assumptions made in the design phase of the system.

4.4.3 Validity of sensor positioning in environment

A key question that designers of robotics applications need to know is whether the sensors they choose are positioned correctly and are able to monitor the space required. Continuing with the example of using a laser scanner oriented horizontally to monitor a workspace around a stationary robot, a designer will want to know where to best place the sensor. From the previous calculation, it is possible to display the minimum required safety distance around the entire robot's trajectory for a complete working cycle as a 2D polygon, projected on the surface of the ground (Fig. 8). Also from the previous step, the designer has configured specific parameters of the laser scanner including the range and the scanning angle (start and finish). Using visualization features from the simulation software, it is possible to display the field of view of the laser scanner. The designer can make this field of view visible and perform a visual check whether the two fields (i.e., the 2D projection of the minimum separation distance for a complete cycle and the laser scanner's field of view) overlap. In this manner, the designer has the possibility to check the validity of the sensor's position and field of view. In the example in Fig. 8, layout with robot, two pallets to empty, and the intermediate table, the red lines represent the range and angle of view of the two laser scanners placed in the scene. The yellow polygon around the robot represents the required separation distance over the entire robot program. The designer can see that the laser scanner range is insufficient to monitor the entire safety zone and needs to make changes. A simple change would be to enlarge the range of the sensor from 3 to 4 m.

Another useful feature is the ability to check whether environment specific objects such as fixtures, tables, or columns are in the field of view of the sensor and require special attention. This is therefore a situation where the models are used in the CAD/simulation environment to visualize information relevant for the design process.

4.4.4 Changing process parameters to meet specific design targets

The advantage of considering the safety aspects of collaborative robotics during the design phase is the ability to adapt the parameters of the entire system to meet specific design targets before building the system. Different options and tradeoffs can be made visible and the design team can discuss these possibilities with management prior to larger investments. A systems engineering approach to the challenge of designing HRC applications necessarily looks at all elements of the system, from the components (robot, gripper, safety sensors, etc.), the process (robot speeds, work pieces, tooling, etc.), the environment, and human factors. The approach we suggest enables the designer to make changes to all of these individual components to see how

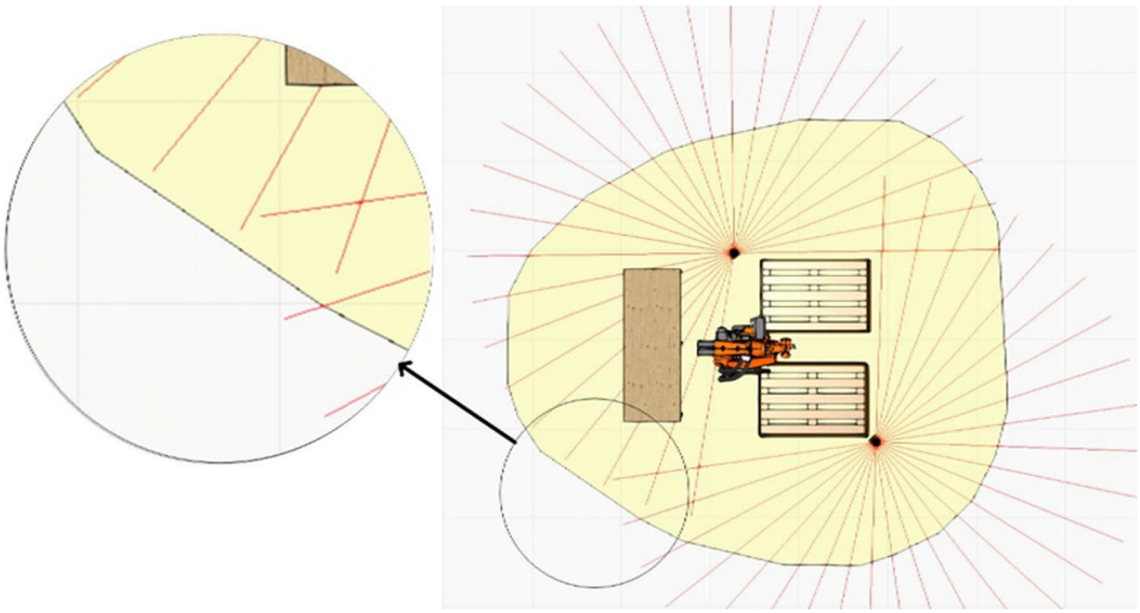

Fig. 8 Layout with robot, two pallets to empty, and the intermediate table. The red lines represent the range and angle of view of the two laser scanners placed in the scene. The yellow polygon around the robot represents the required separation distance over the entire robot program

the overall system will react and in particular to understand the system performance (e.g., achievable cycle times, required floor space). The physical and process-oriented models of the system within the CAD/simulation will have been appended with further safety-related information.

5 Experiments and results

The approach presented in this article has been implemented for use as a plug-in to be used with the commercially available robot simulation software, Visual Components. The software was implemented according to the overall architecture described previously and with clear specification of where the data for the component models comes from (e.g., from the component data sheet, the simulation tool, or the user).

All mechanical and robot information was extracted from the simulation. This includes robot joint speeds, joint configurations, kinematic specification, and payload information. It also includes position information of all components. In many cases, the attributes required for calculation of minimum separation distance were not available from the simulation. This includes information such as sensor configuration details and other parameters about the robot such as maximum payload or maximum robot reach. This data is stored in an internal database with component model information, whereby different methods for populating the database with information are available.

The application described in Section 3 was simulated in Visual Components and the plug-in as described in this article was used for determining the size of the minimum separation distance and for determining other safety-related information. Figure 9 shows three screenshots from the simulation of the robot and sensor system with the aim of supporting the designer with supplemental information about the safety. The application with a robot, 2 pallets (on the right of the robot), and the table (on the left of the robot) is shown for three discreet positions during the robot's program. The red polygon indicates the instantaneous size of the minimum required separation distance, and the yellow polygon (which is the same in all three images) represents the total separation distance over the entire program.

A first comparison relates to the separation distance calculated by a spreadsheet (using worst-case assumptions as described in Section 3.2.7) and the distance as calculated using the robot's programmed speeds. Here we see that the minimum separation distance calculated using our approach is significantly smaller than with traditional methods based on simple worst-case assumptions (Fig. 10).

While the required separation area around the robot for the given program is much smaller than the worst-case calculation, the designer can easily see (Fig. 11) that it still does not fulfill their design goals of fitting within the existing work cell without intruding into the logistic areas. Furthermore, access to the shelves (position 4) and one of the pre-assembly stations is not allowed, as these are firmly within the safety zone of the robot. This means the designer needs to make more changes in order to reach their design goals of getting the system to fit in the existing floor space.

As we stated earlier, the entire application can be viewed as a single system with various parameters available for change in order to reach specific design targets. In this case, the designer would like for a solution to fit into the existing workspace, so as not to disrupt other processes such as the forklift driving areas and the work of the operators on assembly Tables 1, 2, and 3. Options the designer can consider include:

1. Changes to the robot:
   Type (with different braking parameters, payload, reach, etc.)

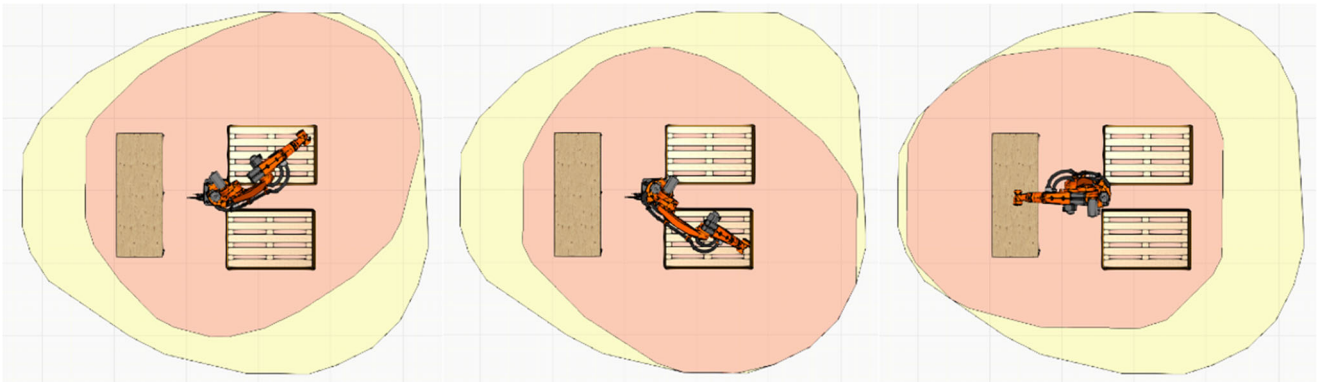

Fig. 9 Screenshots from simulation of robot in application. The yellow polygon represents the minimum required separation distance over the entire programmed path. The red polygon represents the instantaneous required separation distance based on the robot's current speed at that moment along the programmed path

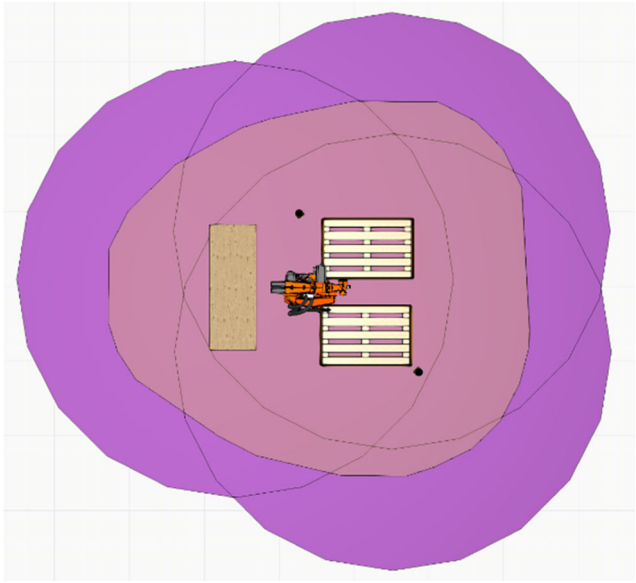

Fig. 10 Comparison on separation distance calculated with worst-case assumptions (purple) as three circles at furthest reaching positions and with our approach (yellow polygon) for KR 22 robot with 100% POV and with laser scanner as safety sensor

Program:

Speed (either for the entire process or for specific motions)

Changes to limit the max extension

2. Changing the sensor:

Type (with associated reaction time and overreach values)

Parameters for use (such as resolution)

3. Changing the process and layout:

This includes adding fencing elements, the position of components in the workspace, or changes to the task itself (e. g., the order of tasks, the responsibility for a task).

Using traditional methods, the designer would need to maintain a number of separate documents including a CAD layout, spreadsheets of safety distances, and a library of pdf files with robot and sensor data. Using our approach, the designer can try all of the various options listed above from within the robot simulation environment and can immediately see the effects of any specific changes. Only valid system configurations can be used due to the internal models, and the component data is saved in a database and readily available for further use. In order to illustrate this, we have carried out a series analyses with four configurations (Table 4). The analysis results in terms of cycle time and required floor space are shown in Table 5.

The final solution favored by the designer could look like the layout shown in Fig. 12. Two light curtains are used to stop the robot when operators approach from the left or forklifts come in from the right. The complete system fits within the existing workspace. Interestingly, changes to the robot speed in this configuration have little effect on the overall size of the workspace. The choice of sensor and its configuration (high resolution, lower C-value) play the largest role in reducing the size of the required separation distance. It should be noted that these four configurations only represent a small number of the options available to the designer. The cycle times are not completely indicative of the process, as the program as it is simply has the robot

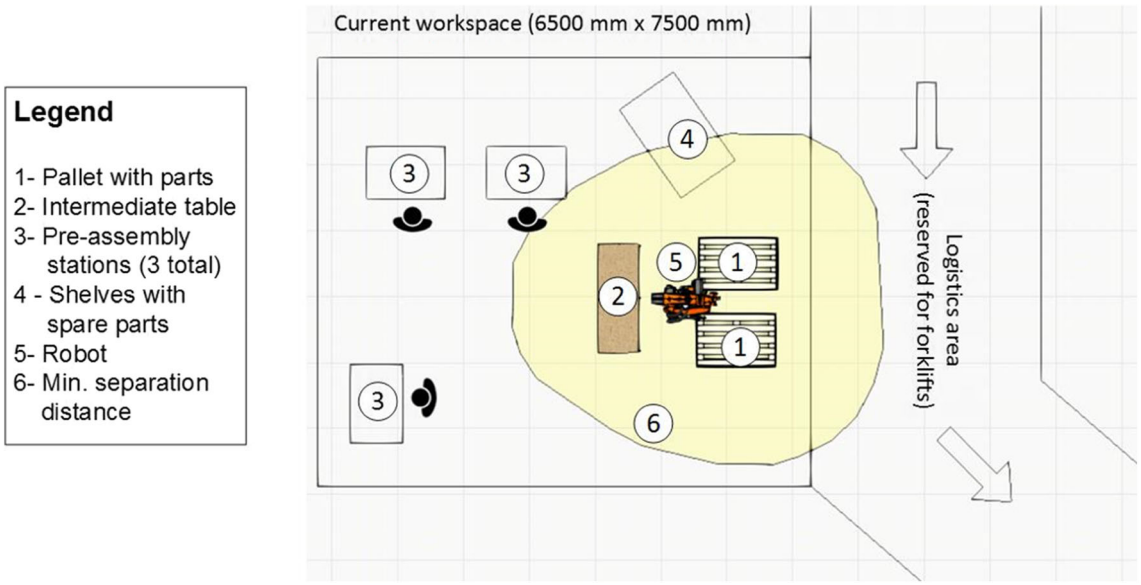

Fig. 11 Required separation distance (yellow polygon) around the robot based on a simple program with robot moving to three extreme positions to empty the two pallets and place the parts on the table

Table 4 System parameters for four test configurations

System configuration parameters	Configuration 1	Configuration 2	Configuration 3	Configuration 4
Robot POV [%]	100 %	100 %	100 %	33 %
Robot payload [%]	66 %	66 %	66 %	66 %
Sensor type	Laser scanner	Light curtain	Light curtain	Laser scanner
Sensor resolution [mm]	70	40	20	70
Sensor reaction time [ms]	90	21	30	90
Sensor C-value [mm]	850	210	50	850
Sensor Z-value [mm]	0	0	0	0

moving to the three furthers positions, but they do show a trend for comparison between configurations with different POV values.

5.1 Discussion

In this section, we would like to discuss the implications and limitations of the use of our proposed method for the consideration of safety during the design of industrial applications featuring human-robot interaction. The methods we propose aim to support the designer of new and existing applications, making information about safety available prior to the final commissioning. Indeed, the system consisting of a collaborative robot with tooling and workpieces, a process, working in a specific environment is very complex and the interdependencies of all the system components are not always understandable or readily traceable. Such a complex system offers many opportunities for adaption of individual or multiple parameters and components to meet specific design goals. Therefore, we believe that the most important advantage of using our method is the possibility to perform "what-if" analyses of the application in a simulation, considering safety-related aspects such as the required size of the minimum separation distance. Parameters that contribute to this include sensor choice, sensor configuration and parameterization, the robot program (speeds and trajectories), and other layout considerations.

Table 6 shows the software that different responsible persons use along the design process (as described in Sections 3 and 4). This represents a sharp contrast to the software used traditionally. With our approach, only the phases of hazard identification, risk evaluation, and hazard elimination still require documentation and software outside of the robot simulation tool with the proposed plug-in. As the state of the art has shown, there is already other research looking at methods to automate the risk analysis, and it is plausible to incorporate their work into this overall workflow.

The required separation area around the robot was over 66% less when using light curtains and over 55% less when using a laser scanner for maximum robot speeds. The discrepancies in this case come through the assumption that the robot will move all the time at its maximum speed. Although the robot program in the simulation was set to 100% for all movements, due to the acceleration and deceleration ramps and the distances for the individual joints, the individual joints were seldom able to reach such a high speed. This becomes evident when assuming a slower robot speed of 33% POV as in the 4th configuration tested. In this case, the area as calculated by the CAS tool is only 12,9% less than the worst-case calculation, indicating a convergence between the assumed and simulated speed of the robot.

As an outlook, we suggest that it should be possible to validate the real physical system based on the simulation

Table 5 Cycle time and size of separation distance around robot for four tested configurations

	Cycle time [s]	Required separation area (worst-case calculation) [m^2]	Required separation area (calculated by CAS Tool) [m^2]
Configuration 1	11	50.72	22.74
Configuration 2	11	35.67	11.22
Configuration 3	11	28.54	9.52
Configuration 4	19	25.99	22.65

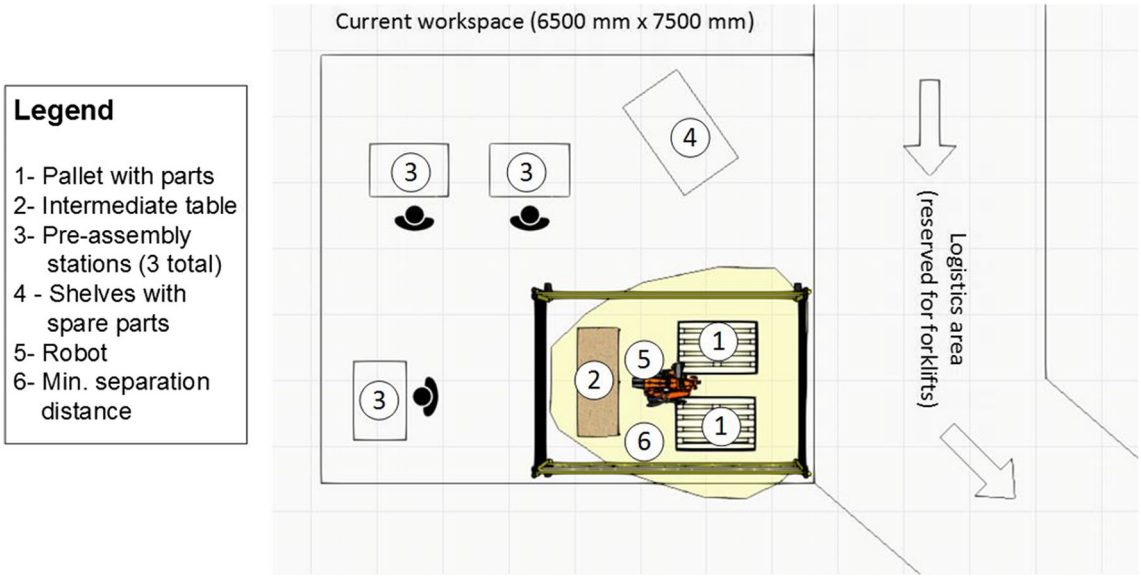

Fig. 12 Final selected configuration with a KR22 robot, 2 light curtains, and 2 fences to limit access to the pallets and the table

results, eliminating the need for physical measurements. This would require that the simulation of the robot (e.g., braking behavior) has been tested and proven to be correct for specific robot models, as well as methods to ensure that the implementation and commissioning correspond to the simulated parameters (e.g., the positioning of sensors). Given a new robot program, it would also be possible to calculate the required minimal protective distance online. As long as the system parameters are known, this could offer a simple means for validating that the safety is respected for robotic movements that are not completely pre-planned, but rather generated in real time. In this sense, it also does not matter who created the new motion commands, either an AI-based agent or a human programmer. The system simply checks what minimum separation distance is required based on the planned motions and sensor parameters, and then checks to ensure that these safety zones are not violated. Future work will focus both on the extension of the model library to more sensor types and manufacturers and on the validation of the simulation to allow for more flexibility in safe HRC applications.

Table 6 Overview of software used by responsible role during the design phases of an HRC application in manufacturing according to our proposed methodology

Design phase according to MD 2006/42/EC from Fig. 2	Responsible role	Software used with our approach
Starting point: General idea of collaborative application	Designer	CAD/Simulation
Safety-oriented design	Designer	CAD/Simulation
General and essential requirements met	Designer	CAD/Simulation with plug-in
Model process, tasks, type of HRC	Designer, safety expert	CAD/Simulation
Define system limits and requirements	Designer	CAD/Simulation with plug-in
Are specific requirements met?	Designer	CAD/Simulation
Hazard identification	Designer, safety expert	CAD/Simulation → document/spreadsheet
Risk evaluation	Safety expert	Document/spreadsheet → other safety evaluation tool
Hazard elimination and risk mitigation	Safety expert, designer	Document/spreadsheet → other safety evaluation tool → CAD/Simulation
Review	Designer	CAD/Simulation

6 Conclusion

This paper proposed a method to model the safety aspects of HRC applications featuring speed and separation monitoring as a means to speed up the design process (save time) and to reduce the size of the factory layout dedicated to the separation distance in 66%. Such HRC applications demonstrated to be characterized by their complexity due to the interdependencies between different system parameters (hardware, sensor range, or robot speed). The proposed methodology was compared with the current practice. Results demonstrate that using our proposed approach, it was possible to quickly make changes to the application, particularly the components, their placement, and to the process itself (i.e., adapting the robot speed) to work towards specific design goals such as minimum cycle time or minimal footprint of the application. The exemplary use-case comparison highlights the difference between the worst-case calculations carried out on a spreadsheet with very rough estimates versus the more granular approach provided through the CAS tool integrated in the simulation.

As an outlook, we suggest that our overall approach of modeling safety-related attributes of the complete system can have wide-reaching consequences for future robotics applications that are flexible and feature online changes to the program during run-time. Currently, a validation of the complete system is required after it has been commissioned in order to ensure that the desired safety effects are indeed achieved by the system.

Funding information This work is part of the MR_KOOP project funded by the Investitionsbank of Saxony-Anhalt under grant agreement 1704/00050.

References

1. Petersen H, Behrens R, Saenz J, Schulenburg E, Vogel C, Elkmann N (2018) Reliable planning of human-robot-collaboration featuring speed and separation monitoring. In: 9th International Conference on Safety of Industrial Automated Systems - SIAS 2018
2. International Standard Organisation: ISO 10218-1:2011: Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots
3. ISO 10218-2:2011 (2015) International Standard Organisation: ISO 10218-2:2011: Robots and robotic devices – safety requirements for industrial robots – part 2: robot systems and integration
4. ISO 13849:2015 (2015) International Standard Organisation: ISO 13849:2015: Safety of machinery – safety related parts of control Systems – part 1: general principles for design
5. ISO 13855:2010 (2010) International Standard Organisation: ISO 13855:2010: Safety of machinery – positioning of safeguards with respect to the approach speeds of parts of the human body
6. ISO/TS 15066:2016 (2016) International Standard Organisation: ISO/TS 15066:2016: Robots and robotic devices – collaborative robots
7. ISO12100: 2010. International Standard Organisation: ISO12100: 2010: Safety of machinery – general principles for design – Risk assessment and risk reduction (2010)
8. Bikas C, Argyrou A, Pintzos G, Giannoulis C, Sipsas K, Papakostas N, Chryssolouris G (2016) An automated assembly process planning system, vol 44. 6th CIRP Conference on Assembly Technologies and Systems (CATS)
9. Byner C, Matthias B, Ding H (2019) Dynamic speed and separation monitoring for collaborative robot applications – concepts and performance. Robot Comput-Integr Manuf 58:239–252
10. Guiochet J (2016) Hazard analysis of human-robot interactions with HAZOP-UML. Saf Sci 84:225–237
11. Guiochet J, Motet G, Baron C, Boy G (2004) Toward a human-centered uml for risk analysis. In: Johnson CW, Palanque P (eds) Human error, safety and systems development. Springer US, Boston, pp 177–191
12. IEC. 41 (2010) International Electrotechnical Commission: IEC 61508-1:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems - part 1: general requirements
13. IEEE (2015) IEEE standard ontologies for robotics and automation
14. Marvel JA, Norcross R (2017) Implementing speed and separation monitoring in collaborative robot workcells. Robot Comput-Integr Manuf 44:144–155
15. International Federation of Robotics (2018) Demystifying collaborative industrial robots
16. Saenz J, Vogel C, Penzlin F, Elkmann N (2017) Safeguarding collaborative mobile manipulators - evaluation of the valeri workspace monitoring system. Procedia Manuf 11:47–54. 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy
17. Saenz J, Elkmann N, Gibaru O, Neto P (2018) Survey of methods for design of collaborative robotics applications- why safety is a barrier to more widespread robotics uptake. In: Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering, ICMRE 2018. ACM, New York, pp 95–101
18. Schlegel C, Steck A, Brugali D, Knoll A (2010) Design abstraction and processes in robotics: from code-driven to model-driven engineering. In: Ando N, Balakirsky S, Hemker T, Reggiani M, von Stryk O (eds) Simulation, modeling, and programming for autonomous robots. Springer, Berlin, pp 324–335
19. Salmi T, Väätäinen O, Malm T, Montonen J, Marstio I (2014) Meeting new challenges and possibilities with modern robot safety technologies. In: Zaeh MF (ed) Enabling manufacturing competitiveness and economic sustainability. Springer International Publishing, Cham, pp 183–188
20. Schröter D, Kuhlang P, Finsterbusch T, Kuhrke B, Verl A (2016) Introducing process building blocks for designing human robot interaction work systems and calculating accurate cycle times. Procedia CIRP 44:216–221. 6th CIRP Conference on Assembly Technologies and Systems (CATS)
21. Tsarouchi P, Spiliotopoulos J, Michalos G, Koukas S, Athanasatos A, Makris S, Chryssolouris G (2016) A decision making framework for human robot collaborative workplace generation. Procedia CIRP 44:228–232. 6th CIRP Conference on Assembly Technologies and Systems (CATS)
22. Vicentini F, Giussani M, Tosatti LM (2014) Trajectory-dependent safe distances in human-robot interaction. In: Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), pp 1–4
23. Blecha P, Blecha R, Bradáč F (2011) Integration of risk management into the machinery design process. In: Jabloński R, Březina T (eds) Mechatronics. Springer, Berlin

24. Askarpour M, Mandrioli D, Rossi M, Vicentini F (2016) SAFER-HRC: Safety Analysis Through Formal vERification in Human-Robot Collaboration. In: Skavhaug A, Guiochet J, Bitsch F (eds) Computer safety, reliability, and security. SAFECOMP 2016. Lecture Notes in Computer Science, vol 9922. Springer, Cham
25. Awad R, Fechter M, van Heerden J (2017) Integrated risk assessment and safety consideration during design of HRC workplaces. In: IEEE Emerging Technologies and Factory Automation (ETFA). Limassol, Cyprus
26. Behrens R, Saenz J, Vogel C, Elkmann N (2015) Upcoming technologies and fundamentals for safeguarding all forms of human-robot collaboration 8th International Conference Safety of Industrial Automated Systems (SIAS 2015). Königswinter, Germany
27. Scippacercola F, Pietrantuono R, Russo S, Silva NP SysML-based and Prolog-supported FMEA. In: 2015 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). Gaithersburg, MD, pp 174–181
28. Lee S, Yamada Y, Ichikawa K, Matsumoto O, Homma K, Ono E (2014) Safety-function design for the control system of a human-cooperative robot based on functional safety of hardware and software. IEEE/ASME Trans Mechatron 19(2):719–729
29. Vemula B, Matthias B, Ahmad A (2018) A design metric for safety assessment of industrial robot design suitable for power- and force-limited collaborative operation. Int J Intell Robot Appl 2(2):226–234
30. Zanchettin AM, Ceriani NM, Rocco P, Ding H, Matthias B (2016) Safety in human-robot collaborative manufacturing environments metrics and control. IEEE Trans Autom Sci Eng 13(2):882–893
31. Directive 2006/42/EC of the European Parliament and of the Council of 17 May 2006 on machinery, and amending Directive 95/16/EC