Human-Robot Coexistence and Interaction in Open Industrial Cells[*]

Emanuele Magrini[a], Federica Ferraguti[b], Andrea Jacopo Ronga[b], Fabio Pini[c], Alessandro De Luca[a,*], Francesco Leali[c]

[a]*Dipartimento di Ingegneria Informatica, Automatica e Gestionale*
Sapienza Università di Roma, Via Ariosto 25, 00185 Roma, Italy
[b]*Dipartimento di Scienze e Metodi dell'Ingegneria*
Università di Modena e Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy
[c]*Dipartimento di Ingegneria "Enzo Ferrari"*
Università di Modena e Reggio Emilia, Via P. Vivarelli 10, 41125 Modena, Italy

Abstract

Recent research results on human-robot interaction and collaborative robotics are leaving behind the traditional paradigm of robots living in a separated space inside safety cages, allowing humans and robot to work together for completing an increasing number of complex industrial tasks. In this context, safety of the human operator is a main concern. In this paper, we present a framework for ensuring human safety in a robotic cell that allows human-robot coexistence and dependable interaction. The framework is based on a layered control architecture, that exploits an effective algorithm for online monitoring of relative human-robot distance using depth sensors. This methods allows to modify in real time the robot behavior depending on the user position, without limiting the operative robot workspace in a too conservative way. In order to guarantee redundancy and diversity at the safety level, additional certified laser scanners monitor human-robot proximity in the cell and safe communication protocols and logical units are used for the smooth integration with an indus-

trial software for safe low-level robot control. The implemented concept includes a smart human-machine interface, in particular to support in-process collaborative activities and for a contactless interaction in terms of gesture recognition of operator commands. Coexistence and interaction are illustrated and tested in an industrial cell, in which a robot moves a tool that measures the quality of a polished metallic part while the operator performs a close evaluation of the same workpiece.

1. Introduction

Recently, there has been an exceptional growth of attention by industrial end-users about the new possibilities opened by the feasibility of a safe Human-Robot Interaction (HRI), namely with robots and humans sharing a common workspace without fences on the factory floor and executing in collaboration a variety of useful tasks under safety premises [1]. In fact, HRI features span several functional aspects that are of interest in many different applications: teaching and programming of robot actions can be made more intuitive and friendly [2], semi-automatic operation of manipulators for tackling very complex tasks is enhanced thanks to on-the-fly human intervention [3], and operators may closely monitor the quality of products by working side-by-side with robots [4]. As a result, collaborative robotics has been considered one of the enabling technologies of the fourth industrial revolution, within the Industry 4.0 program [5, 6] and beyond.

The realization of these long-standing and great expectations has been made possible by the several research results obtained during the last few years within the robotics and automation scientific communities. In order to guarantee a certified level of safety and dependability during physical and cognitive HRI, a number of issues had to be considered in robot mechanical design, actuation and sensory equipment, as well as in algorithms that plan legible motion for the

humans and in human-aware robot control methods [7, 8].

In particular, this has involved novel mechanical designs of lightweight manipulators and of compact actuation (leading to industrial products such as the KUKA iiwa and the Universal Robots URx). These were aimed at reducing inertia/weight exposed to potential collisions and at exploiting the presence of compliant components (including harmonic drives and joint torque sensors) to absorb the energy of undesired impacts. Furthermore, a large variety of external, multi-modal sensors (cameras [9], laser [10], depth [11], structured light, and so on) has been introduced and extensively used, fusing information so as to allow fast and reliable recognition of human-robot proximity and gestures (see e.g. [12]).

In this framework, a general control architecture for handling physical HRI has been proposed in [13]. This consists of three nested functional layers of consistent behaviors that the robot must guarantee: *i)* safety in contact, which is typically realized through a sensorless model-based method for collision detection and reaction [14, 15]; *ii)* monitored coexistence, i.e., when robot and human work close to each other without the need of mutual contact or action coordination, requiring thus continuous obstacle avoidance capabilities by the robot controller, see, e.g., [16]; and *iii)* collaboration, when an explicit and intentional contact is being sought, with a controlled exchange of forces between human and robot [17].

From the cognitive side, it is very important to consider the primary role of users' trust during HRI [18]. Moreover, the operator has to be supported by a suitable Human-Machine Interface (HMI) that gives complete assistance and possibly augmented information about the process status and the next robot action, and should help in task-related decision making [19].

Despite the high performance demonstrated in research-oriented environments, few of these technical methods and scientific results have been transferred so far to real manufacturing applications, in particular where the use of conventional industrial robots is still a must (e.g., due to the heavy payload/tool weight that needs to be carried). While the first known implementations are in-

3

teresting, see, e.g., [20, 21] where automatic task speed reduction in response to human-robot closeness was implemented, they are mostly limited to simple operative conditions and principles. In particular, specific technological processes are disregarded, the size of safety zones between robot and human operator is often kept quite large, resulting in a waste of space of the robotic cell, and issues related to a full system integration are not addressed.

A first reason of the difficulty of transferring solutions that work in the lab to industrial settings is the need to comply with strict safety requirements, notably the robot standards ISO 10218-1 and ISO 10218-2 [22, 23], and the most recent technical specification ISO/TS 15066 [24]. A second bottleneck is that all additional sensors, components, and communication channels should be certified, discarding for the moment cheap and yet effective devices commonly used in research. As a third reason, most of the times the industrial robot in use has a closed control architecture, so that bidirectional access to some of the needed signals (or models) may be impossible at run time.

With the purpose of responding to some of these issues, the H2020 European project SYMPLEXITY [25] has considered a human-robot collaborative approach to perform polishing operations of metallic parts provided by the industrial end-users. In order to understand the rationale for a collaborative solution, the technological process and the current state-of-the-art of robotized solutions is briefly described next.

Polishing is a finishing process to enhance gradually the quality of a surface by subsequent removal of decreasing quantities of material until reaching the required quality [26]. Usually, polishing is a manual process that involves skilled operators, who carefully improve the surface quality following sequential steps. Each step requires dedicated tools, such as abrasive papers and stones, with a decreasing abrasive capacity and applied contact forces that decrease consequently. The high sensitivity naturally owned by humans allows an accurate and responsive control of applied forces, however making the polishing process difficult to automate.

Nevertheless, polishing of large surfaces, e.g., those of big molds used to

4

produce automotive components, are long processes which result in repetitive fatiguing operations, and this has called for a robotized approach. Different solutions have been developed based on industrial robots [27, 28, 29], with dedicated equipment (e.g., optical, for metrology [30]) and digital tools. The existing solutions are intended to replace the operators in the initial low-demanding phases of the polishing process or on areas with simpler geometry, where the target quality falls into the domain of automated system capabilities. The analysis of these solutions demonstrate that the human has still a key role in the polishing process; in particular, skilled operators are required in the last finishing steps with high-demanding quality requirements [31], and whenever it is important to evaluate the surface quality.

With the aim of identifying a solution for an effective execution of polishing processes, the project SYMPLEXITY has proposed a collaborative approach that involves physical and cognitive capabilities, both of the robot and the human, to perform polishing operations and evaluate intermediate and final quality of the surfaces. The considered robotic cell has a reconfigurable structure to manage different polishing tools and uses Abrasive Finishing (AF) and FluidJet Polishing (FP) technologies, depending on the Quality Assessment (QA) of the metallic surface of the workpiece. However, AF and FP are dangerous processes since they involve respectively an electrical high speed spindle and a lance to drive high pressure fluid flow against the component surface. Conversely, auxiliary Setup Operations (SO) on the work cell as well as QA operations are suitable for human collaboration since they do not require further high-power sources in addition to the specific power used for robot motion. Furthermore, human and robot coexistence during QA operations enhance the Manual Finishing (MF) by the operator, who could check in fact the surface quality while the robot performs the measuring process.

The collaborative concept proposed in this paper realizes the suitable level of *safe coexistence* within HRI, according to the layered control architecture of [13]. For the considered polishing task, this is implemented for those operations in the work flow where interaction turns out to be beneficial. To this purpose, we

deployed our effective method based on depth (RGB-D) sensors [11, 16] to com-
pute in real time the relative distance between a moving human and the robot
in action. This accurate distance information can be used to modify online the
motion of the industrial robot, so as to avoid any unintended contact or collision.
In its simplest version, the reaction strategy reduces the planned speed or stops
the robot according to safety specifications. Moreover, in order to enforce a form
of functional safety in our human-robot shared environment [32], allowing the
use of generically unsafe sensory and computational components for advanced
applications, we have integrated our distance evaluation algorithm with redun-
dant sensing hardware (two laser scanners) that monitors in parallel the cell,
established all communications only through a safe protocol (PROFISafe), and
used a safe-oriented proprietary robot control software (SafeMove by ABB). The
implemented HRI concept covers also *contactless collaboration* at the cognitive
level, with action commands recognized from human gestures and with the use
of an advanced HMI supporting the operator for in-process quality assessment,
also with the help of a database of previously collected polishing information
and situations.

The rest of the paper is organized as follows. Section 2 recalls the safety
requirements in place for human-robot collaboration, together with how our
three-layer control architecture for HRI fits into the general picture. Section 3
describes in detail the SYMPLEXITY collaborative polishing cell, including
its layout and the collaborative activity diagram break-down for the polishing
task. The safety layer design, including the cell monitoring sensors and the extra
safety hardware and communication used is discussed in Section 4. The concept
of depth space sensing is briefly presented in Section 5. Section 6 describes
our most efficient implementation of the human-robot distance computation
algorithm, which is at the core of the coexistence strategy. We report here the
experimental results on the monitoring performance in the SYMPLEXITY cell
and also the overall system behavior in the presence of a sensor failure. Finally,
Section 7 illustrates the human-machine interface, including handling of human
gestures. Findings and on-going work are summarized in Sec. 8.

2. Safety Requirements

Collaborative robotics is an approach allowing direct robot and operator interaction without traditional safeguarding under specific conditions. With the introduction of human-robot collaboration technologies, an even greater relevance is attributed to robot safety standards, which have been updated to address co-working scenarios. The international safety standards ISO 10218-1 and ISO 10218-2 [22, 23] have identified specific applications and criteria where collaborative operations can occur. More recently, the technical specification ISO/TS 15066 [24] has been introduced to specify safety requirements for collaborative industrial robot systems and the work environment, supplementing the requirements and the guidelines on collaborative industrial robot operations outlined in ISO 10218-1 and ISO 10218-2.

The safety standards ISO 10218-1/2 and the technical specification ISO/TS 15066 identify four forms of collaboration, which can be used either individually or in combination, depending the application concerned and the design of the robot system. These are summarized in Fig. 1 and described as follows [33].

1. **Safety-rated monitored stop - SMS.** The robot is stopped during the interaction with the operator in the collaboration space. This status is based on a monitoring system with specified safety performance and thus the drive can remain energized. This is the simplest type of collaboration. The robot and the operator can work, but not at the same time. This mode of operation is suitable for the manual placement of objects on the robot end-effector, in static visual inspection, for finishing or complex tasks where human presence is required, or when robots can help the operator with the positioning of heavy components [34]. This form of collaboration requires dedicated functionalities to monitor the robot standstill: in the so-called "Safe Standstill" (SST) mode, the robot movement is inhibited completely through dedicated redundant software and electronics-based safety technology [35]. These functionalities are integrated in collaborative robots, while they have been recently provided as

7

Figure 1: The four forms of collaboration identified by the robot safety standards ISO 10218-1/2 [22, 23] and by the technical specification ISO/TS 15066 [24].

an option for industrial robots [36].

2. **Hand guiding - HG**. The safety of the human-robot collaboration is assured by the robot being guided manually under control at an appropriately reduced speed. In this form of collaboration, the operator can teach the robot positions by moving the robot without the need for a teaching interface, e.g., a robot teach pendant [37]. The own weight of the robot arm is compensated to hold its position without user effort.

3. **Speed and separation monitoring - SSM.** Speed and motion path (i.e., the trajectory) of the robot are supervised and adjusted based on the speed and position of the operator in the safeguarded space. With reference to Fig. 2, the robot can operate at full speed when the human is in the green zone, at reduced speed in the yellow zone, and stops when the human moves into the red zone, which defines the minimum separation distance. Proximity of the operator to other robot-related hazards, like a manipulated object dropped accidentally by the robot, is treated similarly. All these areas are usually monitored by scanners or vision systems [34]. Non-isotropic distances to hazard are also considered, e.g., depending on the shape, speed, and direction of motion of a dangerous tool mounted on the robot end-effector. Suggestions and guidelines for implementing SSM are provided in [38], while in [39] the standard SSM has been improved for dynamically updating the robot speed limit depending on the separation distance to nearby users and the robot's direction of motion.

4. **Power and force limiting - PFL.** Physical contact between the robot system (including the workpiece) and the human operator can take place either intentionally or unintentionally. A safe behavior is achieved by limiting the contact force to values at which damages or injuries are not to be expected. Collaboration based on limiting force requires often the use of robots designed specifically for this feature. The technical specification ISO/TS 15066 includes maximum values (biomechanical load limits) corresponding to onset of pain when the robot collides with the different body parts. A description of collision handling is presented in [40] in terms of

9

Figure 2: Safety zones for human-robot collaboration with speed and separation monitoring technology (Courtesy of ABB Robotics).

four possible robot responses to the contact. The simplest reaction is to activate the robot brakes after collision, with an immediate stop. Torque control mode with gravity compensation, torque and admittance reflex are improved strategies [14], which result in safer behaviors such as decreasing the impact energy through counter-motion in the opposite direction. A method for limiting the forces applied by an industrial robotic manipulator by detecting the collision with its surroundings without the use of external sensors has been proposed in [41].

The ISO 10218-1/2 safety standards underline the importance of hazard identification and require that a risk assessment be performed, especially for collaborative robots and for those operations that dynamically involve the operator and the robot, such as SSM and PFL. Although not safe under all conditions, a maximum reduced speed of 250 mm/s is considered to be low enough to allow an operator to react to unexpected robot motions during collaborative operations [22]. The technical specification ISO/TS 15066 provides additional information and further guidelines to evaluate the risk related to the four forms of collaboration [24]. The document specifies how to determine the admissible physical quantities for the collaboration forms SSM and PFL, such as minimum separation distances and limits of mechanical loadings over the human body, depending on the risk assessment.

10

2.1. Connection with our layered control architecture for pHRI

As mentioned in Sec. 1, we proposed in [13] a hierarchical control architecture for handling physical Human-Robot Interaction (pHRI), which consists of three nested layers, see Fig. 3(a).
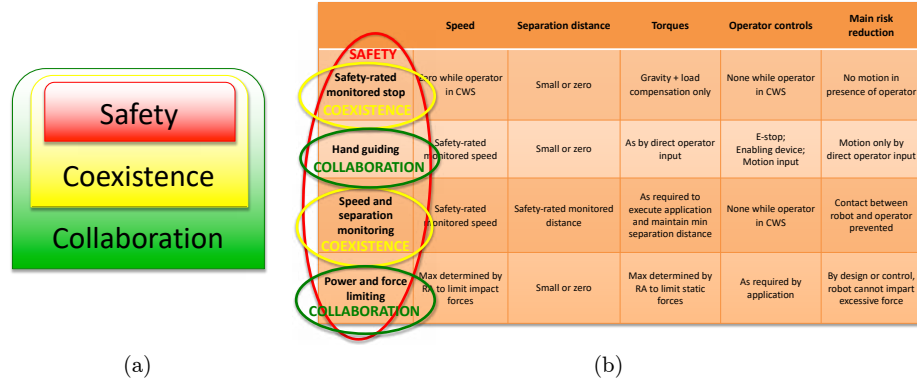

Figure 3: (a): The three nested layers of the hierarchical control architecture for pHRI proposed in [13]; (b) The mapping of the three control layers into the four modes of the ISO standards on robot safety.

We define safety, coexistence, and collaboration as functional behaviors that must be guaranteed in a consistent way during robot operation. Indeed, *Safety* is the most important feature of a robot that works close to human beings, and should always be enforced in any condition. As we have seen, depending on the risk assessment of a robotic task, one should limit the total instantaneous power of a robotic system in operation, the maximum speed of the robot TCP in the presence human environment, and the intensity of the exchanged contact forces.

Nonetheless, especially in research environments or for personal service applications, an effective pHRI may become unfeasible by the straight application of safety standards, or too constrained without a further classification of the intended scope of the human-robot interaction. This is why we have introduced also a layer for *Coexistence*, when the robot can share the workspace with other entities, most relevant with humans. In this case, human safety requirements

11

must be consistently guaranteed (i.e., obtaining a safe coexistence). An example of coexistence, sometimes called also coaction, is when robot and human operator work side-by-side without ever requiring a mutual contact. This is one of the applicative situations of interest in the present work.

Finally, we denoted the most demanding request in pHRI as *Collaboration*, namely the robot feature of performing a complex task with direct human interaction and coordination. This may occur in two different, not mutually exclusive modalities. In *physical* collaboration, there is an explicit and intentional contact with exchange of forces between human and robot. In *contactless* collaboration, there is no physical interaction: coordinated actions are guided or will follow from an exchange of information, which is achieved via direct communication, like with gestures and/or voice commands, or by recognizing human intentions. We refer to safe physical collaboration when this collaboration is consistent with safe coexistence, i.e., when the safety and coexistence features are guaranteed during physical collaboration phases. For example, if the human is collaborating with the robot using his/her right hand, contact between the robot and the left hand or the rest of the human body is undesired, and therefore such accidental contacts are treated as collisions that must be avoided. Similarly, if during a contactless collaboration the human enters the robot workspace, the human-robot system should be controlled so as to preserve safe coexistence.

The technical characteristics of the considered polishing application by abrasive finishing go well beyond the possibility of achieving a safe physical collaboration. Therefore, we limit ourselves to a situation of contactless collaboration through gestures. It is worth mentioning that the proposed hierarchical control architecture fits very well with the form of collaborations considered in the ISO standard [22] and in the technical specification [24]. The related mapping is illustrated in Fig. 3(b).

3. Collaborative Polishing Cell

In the SYMPLEXITY project we aimed at taking a step forward on symbiotic processes that involve both robot and human in the execution of complex tasks, in particular for polishing operations. In providing polished surfaces, the SYMPLEXITY approach exploits dedicated technologies with respect to the required quality of final surface. Thus, besides including polishing technologies to perform surface finishing operations, the approach integrates also an objective quality assessment phase, which drives the planning of the polishing sequence to produce the expected quality. Figure 4 summarizes the approach with a clear indication of the human central role in process planning with respect to the four technologies involved in the process, which are described as follows.

- *Abrasive Finishing*, AF. Although traditional manual polishing is based on abrasive tools, AF technology refers to dedicated equipment and approaches for robotic polishing. AF requires an electric spindle attached to the robot wrist to rotate or translate dedicated tool holders which drive abrasive tool tips. Tool holders have compliant kinematics drive by compressed air with the aim to adapt the contact forces to surface shape by means of dedicated procedures which control the pressure of air channels [42]. Quadrant *a* of Fig. 4 depicts the setup of robotic AF.

- *Fluidjet Polishing*, FP. It is a novel technology which exploits an abrasive mixture (water and aluminium oxide particles) as polishing agent. As in quadrant *b* of Fig. 4, the robot handles a dedicated lance with selected nozzle mounted at its end to shot the surface with medium pressure mixture flow [43].

- *Quality Assessment*, QA. Objective assessment of polished surfaces is a key point in finishing processes, since nowadays it is still a fully manual operation left to few experienced operators. SYMPLEXITY has proposed an on-line measurement technology to control the quality of the reference surface. CWS is the metrology head relying on interferometer techniques
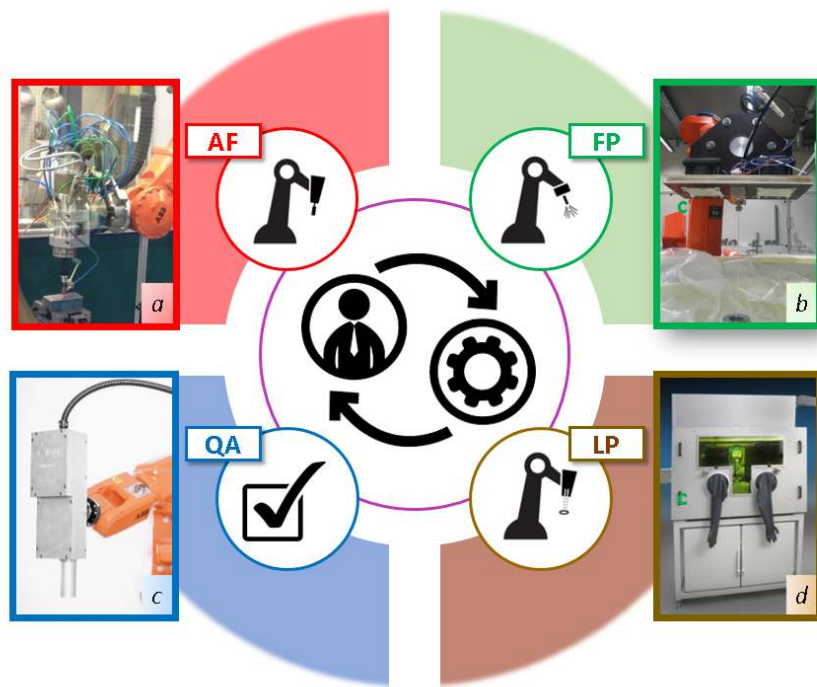
Figure 4: Polishing approach as suggested in SYMPLEXITY. The four quadrants distributed alongside the central circle illustrate the equipment used for the polishing technologies.

that provides objective data about the surface state. Quadrant c of Fig. 4 depicts the robot setup where CWS is moved by the robotic arm over the surface [44].

- *Laser Polishing*, LP. A dedicated laser polishing head is used to polish the surface through micro melting operations that reduce and smooth the extra material [45, 46]. The robot is enclosed in a protective cabin which holds harmful light emission due to laser source, quadrant d of Fig. 4.

Integration of the proposed technologies in a flexible production solution requires dedicated systems to allow automatic reconfiguration with respect to required operations, along both hardware and software solution [47]. Moreover, a digital model of the system is used to derive the polishing toolpath for the robot and with respect to the presented technologies. Consequently, dedicated computer aided supports for machine tool and robot programming, respectively CAM/CAD, are at the base of the SYMPLEXITY solution for providing the polishing toolpaths. On the other hand, human cognitive collaboration is fundamental to drive the entire process, as well as human coexistence in the robot working area to allow checking or executing of final operations, such as the fine polishing of not-compliant areas of the whole surface. Interaction through dedicated interfaces and especially safety countermeasures for human-robot collaboration are thus of great importance in the cell design and process definition [48].

3.1. Cell layout and safe collaborative concept

The present work focuses on the design of safety countermeasures to ensure the coexistence of operators during the execution of polishing operations, based on the technologies which has been presented previously. With respect to the design approach of traditional robotic workcells, a collaborative solution requires dedicated clarification phase to analyse who are the actors involved during collaborative polishing operations with the aim to provide a safe solution for the users. Accordingly to the four novel robotic polishing technologies involved, it

15

has been identified a theoretical activity diagram, summarized in Fig. 5, which provides the conceptual collaborative polishing process.

The LP technology requires dedicated equipment which makes it difficult to integrate in one polishing workcell. Consequently, the activity diagram and thus the collaborative robotic workcell refers to an integrated polishing solution for AF, FP and QA operations. In addition to these, auxiliary operations for workcell setup, SO (Setup Operations), and further manual finishing polishing actions, MF (Manual Finishing), are required to execute the process and reach the required quality on surface of the part. The activity diagram provides theoretical working sequence and identify which operations could be carried out through a human robot collaborative approach. The SO phase starts the process. The operator equips the workcell with required tools and positions the part to be polished in the working zone. In parallel, the robot can execute setup operations, such as part calibration and tools calibration, to respectively identify both the origin of the part and the end point of the tools [49, 50]. Subsequently, QA checks the quality of part surfaces to identify the initial state and thus drive the selection of the required polishing operations. If some tools are not available, another SO phase takes place, otherwise robotic polishing operation will start. AF and FP operations could be executed sequentially, individually, and repeated many times, according to the surface evaluation returned in the QA phase. After the robot polishing operations, the QA phase will determine the achieved quality and, possibly, a last refinement phase of MF will provide the final quality. Concerning human-robot collaboration, during the execution of AF and FP tasks it is not possible to have an operator inside the robot work space. In fact, the robot handles dangerous tools, such as an electrical spindle with high-speed rotating tools (producing sparkles and launching debris when in contact with the metallic surface) or a long streamlined lance shooting a pressurized abrasive mixture. A collaborative scenario is possible instead for the QA and SO operations. In the SO phase, for instance, a tool change operation that requires attaching a spindle (in no-running mode) to the robot can be done while the human is in the cell. Indeed, human-robot distance should be
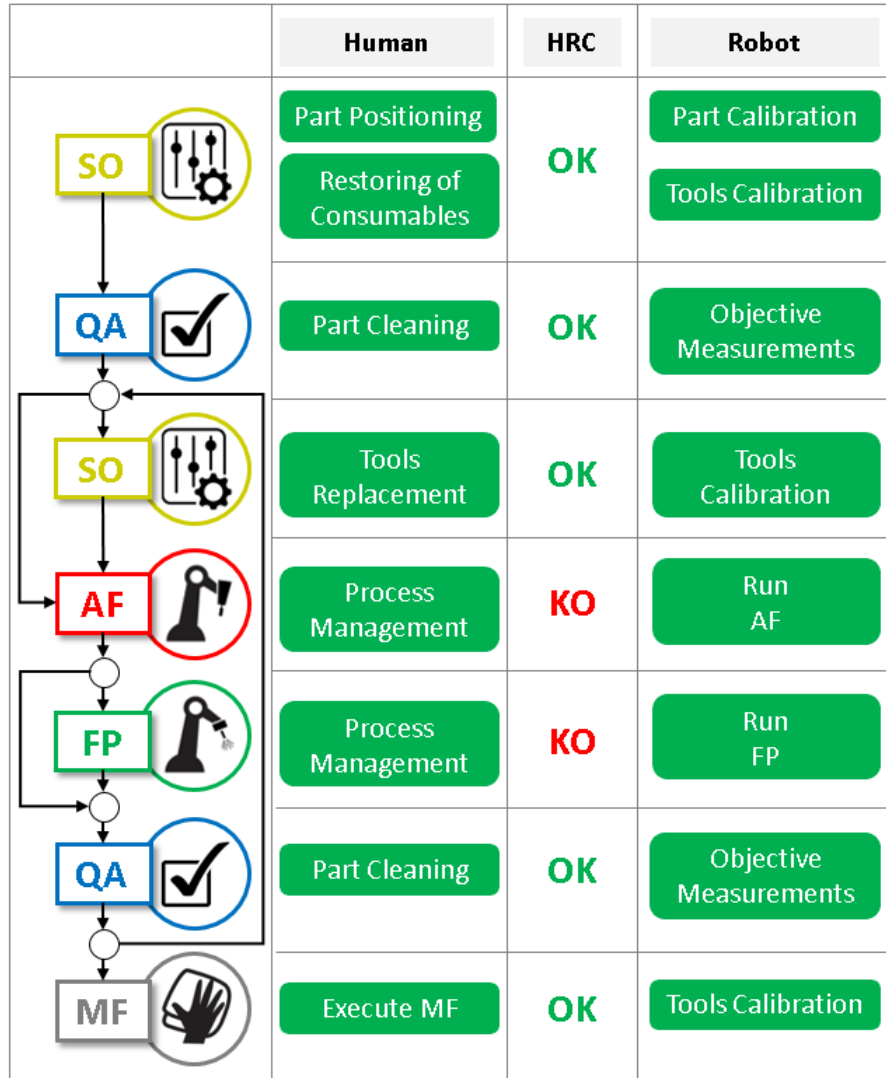
16

Figure 5: Activity diagram related to the polishing process and allocation of collaborative operations between human and robot.

monitored to keep the operator away from the tool changer/robot during this
operation (crush hazard). Also, the spindle control would need to be interlocked
with safety signals to ensure that it cannot turn on when the operator is present.
As a result of this analysis, the HRC column in the diagram of Fig. 5 identifies
collaborative scenarios with *"OK"* and non-collaborative ones with *"KO"*.

A flexible usage of the proposed polishing technologies as well as the selection
of specific collaborative scenarios requires a reconfigurable solution to allow
the selection of specific polishing action, and the result achieved is depicted
in Fig. 6, where an exploded view of the HRC reconfigurable workcell for AF
and FP is presented. The workcell has a modular construction to easily adapt
the process. On the ground, there is a rigid baseplate (**#1**) used to transport
the workcell and to collect fluids generated during the FP operations. On the
baseplate, there is an integrated rail (**#2**) which moves the robot base (**#3**).
This solution augments the robot workspace; furthermore, the rail allows to use
the robot with the more rigid configurations which return better positioning
accuracy. The robot is equipped with a Force/Torque (F/T) sensor (**#4**) and
a quick change system (**#5**). The F/T sensor will be used to read the forces
and torques exchanged during the polishing processes. The quick change system
makes the automatic reconfiguration of the system over the process operations
possible. The quick change system is used to attach the robot end-effectors
placed on a dedicated depot (**#6**); the picture shows the AF spindle (**#7**) in
the depot position. Near the end-effectors depot, there is a tooling warehouse
that contains the AF tool holders (**#8**). The working zone is in the middle
of the cell, front side. A working table (**#9**) is used to place the parts (**#10**)
which will be processes by the robot and the human, following the collaborative
processes described previously. On the working table, it is possible to place a
movable cover (**#11**), to protect the environment during the FP process against
sprayed fluids, splashes and condensation. The protective cover is located above
the end-effectors parking (**#12**); the robot attaches the protective cover and
performs the movement to place the cover on the part. This cover has an open
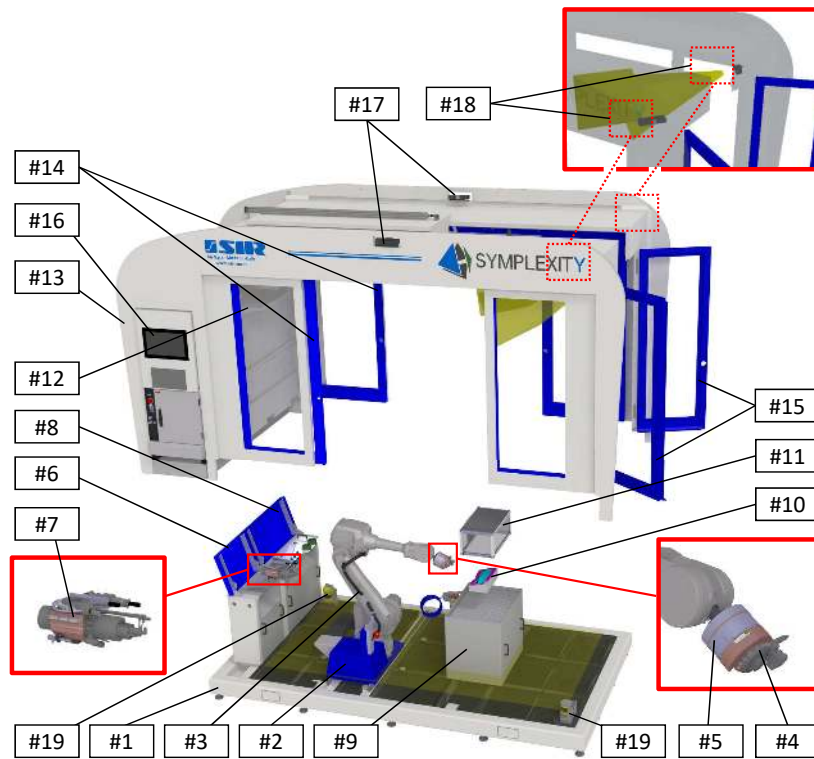passage on the top for inserting the Fluidjet lance.

18

Figure 6: Embodiment design of AF+FP collaborative workcell.

³⁹⁰ With respect to the forms of collaboration identified by [22, 23], as summarized in Sec. 2, and to the evaluation of dangerous operations, as in the activity diagram of Fig. 5, the QA and SO operations, as well as the MF operations, can be made compliant with the safety requirements for human-robot collaboration. Conversely, AF and FP operations are not suitable for a collaborative scenario, ³⁹⁵ and therefore the working area will be bounded by a protective cabin (**#13**). The cabin has two automatic sliding doors on the sides (**#14**) and two hinged doors on the front (**#15**); the doors will opens during collaborative phases and transform a closed space in an open environment accessible by the human. To change the cell configuration and activate a collaborative operation, the user ⁴⁰⁰ can request to enter inside the cabin from a HMI at the back side of the cabin (**#16**), or more naturally using dedicated gestures (see Section 7.1) that can be recognized by two external Microsoft Kinect V2 sensors, placed on the top of the sliding doors. The view in Figure 6 shows the external Kinect on the left side (**#17**). Starting from the initial embodiment design, the final developed ⁴⁰⁵ solution looks like in Fig. 7, as presented in a recent international fair.

4. Safety Layer Design

We present here the main hardware/software components concerning safety that have been used in the cell design.

4.1. Kinect depth sensors

⁴¹⁰ To monitor the robot workspace during collaborative operations, the reconfigurable workcell is equipped with two additional Kinect V2 depth sensors inside the cabin (**#18**), which are placed at the two top corners of the cabin on the side where the human is expected to work, as in Fig. 8. The two internal Kinects act as 3D depth camera, monitoring the workspace and providing ⁴¹⁵ input data to the algorithm that computes distances between the robot and human/obstacles during a coexistence phase in a very efficient way, as described in Section 6.

Figure 7: The final collaborative workcell used for the SYMPLEXITY demonstration (as presented at the AUTOMATICA 2018 fair).
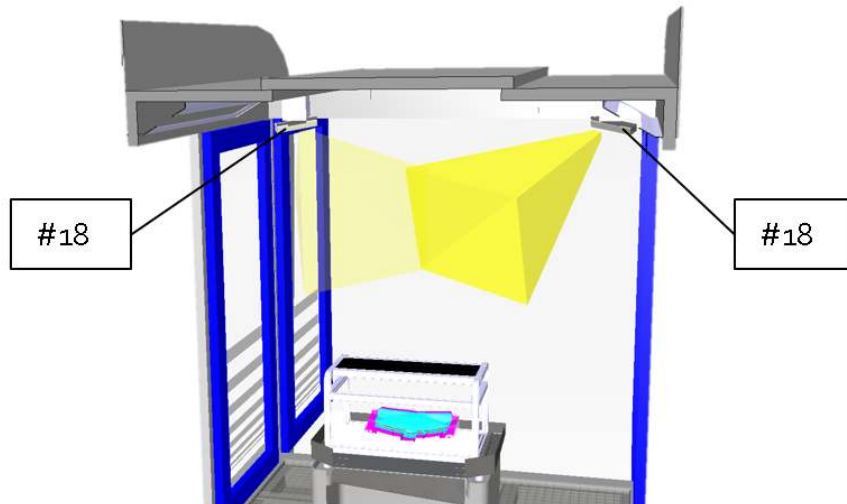

Figure 8: Placement of the two Kinect V2 sensors inside the cabin.

Although two sensors provide a redundant solution, one (or both) may occasionally fail. In particular, potential safety-related issues can be raised by a Kinect failure or severe occlusion, i.e., when wrong or no depths are being measured. Even the distance algorithm could fail in principle, e.g., in case of a bad communication quality between robot and host PC, providing no control signal to slow down (or stop) the robot in critical conditions. The occurrence of such faulty conditions can be detected and mitigating actions can be taken, as described later in Section 6.3, but these issues are anyway critical for the safe handling of human-robot coexistence.

As a matter of fact, the Kinect sensors are components which are not rugged enough for industrial applications —and even less since they lack a certification in terms of safety operation. In order to extend the safety of the proposed solution without abandoning the flexibility of use of the depth sensors, additional countermeasures have been identified which concern the workspace monitoring, the low-level robot control, and the integration and communication among peripherals. Extra monitoring capabilities and diversity/redundancy of components are in fact the preferred ways to mitigate the inclusion of unsafe (though high-performing) sensors and related algorithms into human-robot collaborative tasks [32].

4.2. Workspace monitoring

An additional safety protection has been integrated inside the cabin with the aim to prevent consequences on the operator from an unexpected behaviour of the depth sensors. Figure 9 illustrates the position of two laser scanners KEYENCE SZ-V32n placed on two opposite corners of the cabin, about 50 cm from to the floor (at the calf height). With the proposed placement, the laser scanners define invisible planes that detect the presence of moving or standing humans inside the cell. Through a dedicated interface, it is possible to program which are the elements in the cell that should not be identified as human intruders (e.g., the robot, the sliding plate on the track, and the working table). Note that the laser scanners are always active and work in parallel to the depth
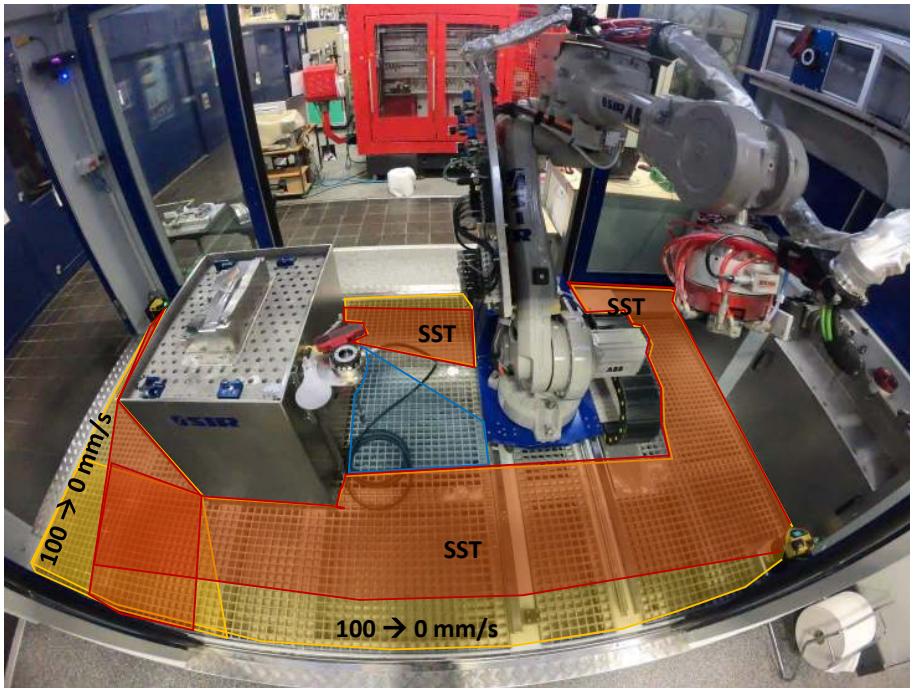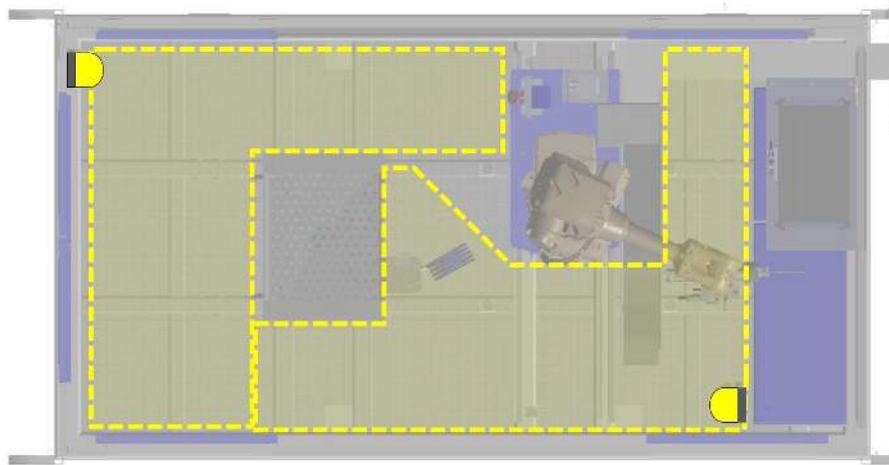
22

Figure 9: Top: The two laser scanners mounted close to the floor and their safety planes over the internal area of the cabin. Bottom: A 3D-view of the cell with the SST safety zones chosen in ABB SafeMove being highlighted. One of the two Kinects can be seen at the top left corner of the cabin.

sensing system. In case of a faulty behavior of a Kinect, the output from the two laser scanners will be evaluated by the cell controller: if an operator is present inside the cell, any collaborative action will stop until the recovery of the depth sensing system from its faulty condition.

4.3. SafeMove suite

The collaborative solution proposed in SYMPLEXITY had to include a 6-dof standard industrial robot, in order to lead to a workcell performance comparable to that needed in industrial environments. In this way, the achievements obtained using the workcell could be used to evaluate its performance with respect to real manufacturing needs expressed in terms of typical production parameters, say production rate or reconfiguration time. Conversely, the use of an industrial robot of medium-large size for collaborative operations poses serious risk conditions for the operator because its mass/inertia and large achievable speed does not allow a straight integration in safe human-robot interaction tasks.

This issue is well known in industrial scenarios. Thus, the main robot manufacturers have some form of software-based technology integrated into their robot controllers, which allow human-robot collaborative operations even with traditional high-payload manipulators [51]. Robot manufacturers such as FANUC, KUKA and ABB proposed, respectively, the Dual Check Safety technology [52, 53], the Safe Operation [54] and the SafeMove [36] solutions, with the aim to comply with safety requirements when implementing HRC systems. The main features involved are position, speed and zone check functions. In the solution proposed within our project, the robot selected for the workcell is an ABB IRB 4600, with a payload of 60 kg and a reachability of 2.05 m. The SafeMove option from ABB has been integrated, with a dedicated hardware and software configuration required to enable the robot movements. Safety zones leading to different robot behavior can be specified in the robot workspace, e.g., a Safe Stand Still (SST) zone (see Fig. 9), where the standing still state of the robot is supervised even if the servo and drive systems are in regulation [36]. Through an interface based on the RobotStudio software, it is possible to define

24

synchronization parameters in order to connect the robot with a safety cell controller. Moreover, safety zones and related speeds, as well as other safe-oriented signals, can be communicated within a hard real-time schedule between external devices and the robot controller.

4.4. Integration and peripherals control

Robot SafeMove functionality is required for collaborative operations when the cabin doors are open, in order to ensure that the robot speed is below the chosen safety threshold, which depends on the working area and the distance to the human operator. During AF and FP operations, the cabin doors remain always closed, and therefore the robot can move at high speed since the operator is not present the robot workspace. In order to switch off the SafeMove, additional activities are required in order to verify the state of cell peripherals and to ensure that the cabin is closed. Note that the laser scanners can still be active at the beginning of the AF/FP operations, as soon as the doors are closed and before starting the process, to double-check that no user is inside the cell.
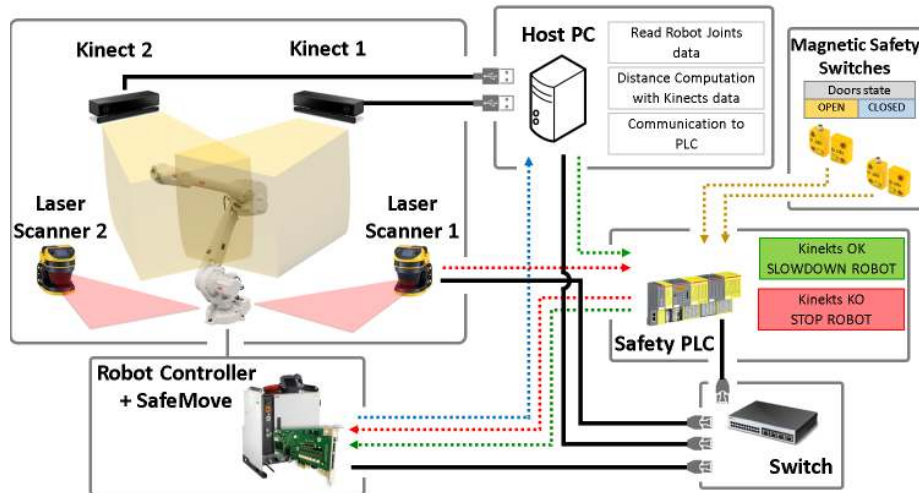

Figure 10: Control and communication architecture for safety.

A Safety PLC has been selected as cell controller with the aim to implement

safe logics under hard real time constraints. The PLC collects all the signals from safety devices that handle the state of the cell doors as well as the state of dangerous components, e.g., the AF and FP robot end-effectors. Safety pads are used to control if the doors are closed or, respectively, if the AF and FP tools are in the parking zones. In case a spindle is attached to the robot, there is also a sensor that monitors for spindle motion when the cell doors are open. As a result, the Safety PLC communicates with the robot SafeMove device to activate speed check with respect to the state of safety pads and sensors. Figure 10 illustrates the whole control and communication architecture. The Safety PLC acts on the cell devices and commands the robot behaviour through SafeMove. Black solid lines and coloured dotted lines represent respectively physical connections and data exchanged between devices. The PROFIsafe technology [55] is used over communication paths, from the laser scanners to the robot controller, the host PC, the Safety PLC, and to the door magnetic switches peripherals, integrating standard and safety data exchange on one cable (black channel principle) and providing a flexible functional safety to the system.

5. Depth Space Sensing

The distance between the robot and the nearby obstacles is an essential information needed to guarantee a safe human-robot coexistence. In our approach, distances between the robot body and all the obstacles in the camera field of view are evaluated in an efficient and fast way, analysing directly the images in depth space provided by the vision sensing (see Section 6).

Following the definition used in [11], the depth space is a non-homogeneous 2.5 dimensional space, where the first two dimensions represent the coordinate of the projection of a Cartesian point in the image plane, and the third one represents the distance between the point and the image plane. A device that provides images of the environment in depth space coordinates is called depth sensor. It can be usually modeled as a classic pin-hole camera and relies on different technologies, such as stereo vision, structured light, and time of flight.

26

In order to model the data processing of a pin-hole camera, two sets of parameters are required: the intrinsic parameters collected in a matrix $\mathcal{K}$, which describes the transformation from a Cartesian point into the image plane, and the extrinsic parameters organized in a matrix $\mathcal{E}$, which describes the coordinate transformation between the world and camera frame. We have

$$\mathcal{K} = \begin{pmatrix} fs_x & 0 & c_x \\ 0 & fs_y & c_y \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathcal{E} = \left(\left. {}^c\boldsymbol{R}_w \, \right| \, {}^c\boldsymbol{t}_w \right), \tag{1}$$

where f is the focal length of the camera, s_x and s_y are the pixel dimensions measured in meters, c_x and c_y are the coordinates of the image plane center, along the focal axis, expressed in pixel, and ${}^c\boldsymbol{R}_w$ and ${}^c\boldsymbol{t}_w$ are respectively the rotation matrix and translation vector between the world and camera frame. The depth information of the observed point, i.e., the distance between that Cartesian point and the camera image plane, is stored in the corresponding pixel of the depth image. It follows that only the depth of the closest point belonging to a projection ray is provided. This means that all points located beyond are occluded from the camera view and compose an uncertainty region in the Cartesian space called *gray area*, as shown in Fig. 11.

Consider an arbitrary point in the 3D space, expressed in the world frame as $\boldsymbol{P}_w = \left(\begin{array}{ccc} x_w & y_w & z_w \end{array} \right)^T$. From matrix $\mathcal{E}$ in eq. (1), its expression in the camera frame is

$$\boldsymbol{P}_c = \left(\begin{array}{ccc} x_c & y_c & z_c \end{array} \right)^T = {}^c\boldsymbol{R}_w\boldsymbol{P}_w + {}^c\boldsymbol{t}_w. \tag{2}$$

The corresponding projected point $\boldsymbol{P}_d = \left(\begin{array}{ccc} x_d & y_d & z_d \end{array} \right)^T$ in depth space is given then by

$$\begin{aligned} p_x &= \frac{x_c f s_x}{z_c} + c_x \\ p_y &= \frac{y_c f s_y}{z_c} + c_y \\ d_p &= z_c, \end{aligned} \tag{3}$$

where p_x and p_y are the pixel coordinates in the image plane and d_p is the corresponding depth value of the point.
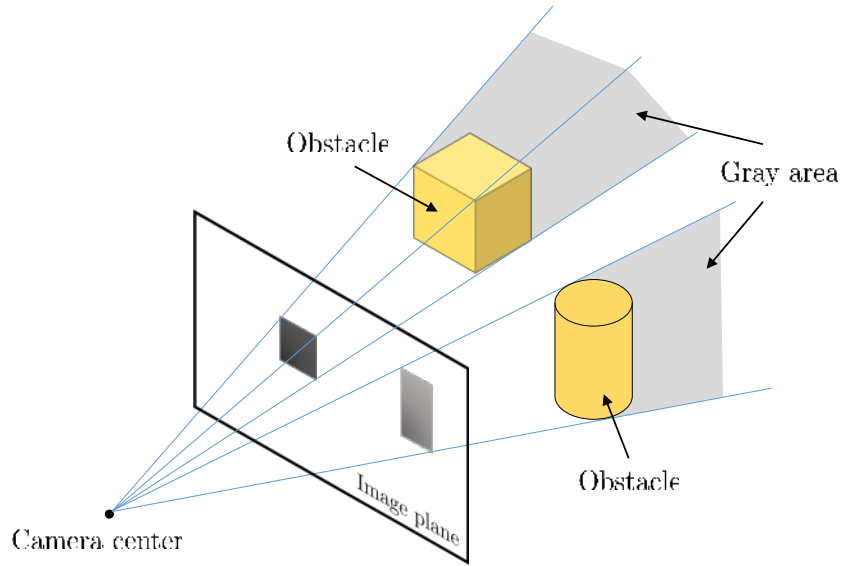
27

Figure 11: Generation of a depth image, with lighter intensities representing closer objects. Points occluded by the obstacle compose the gray area in the Cartesian space.

6. Real-Time Distance Computation in Depth Space

The proposed real-time distance computation algorithm [56] has been implemented as a Graphics Processing Unit (GPU) program in order to exploit the parallelism of this new graphic boards architecture. In particular, it is based on the CUDA framework for parallel programming within the NVIDIA environment, and on OpenGL library that provides hardware accelerated rendering functions. In relation to a common CPU, each core in a GPU is able to execute at the same time thousands of processes. This high degree of parallelism gives to any GPU-based algorithm huge performance improvements, thanks also to a high-speed memory closely interconnected to the GPU cores. The CUDA API library provides developers with access facilities to the GPU resource, with the possibility of writing programs similarly to the case of CPUs.

6.1. Image processing

To compute distances between (human) obstacles and robot, our approach relies on the processing of three 2.5D images, all having the same resolution:

28

- *Real depth image* is an image of the environment as captured by the depth sensor, see Fig. 12(a).

- *Virtual depth image* is an image containing only a projection of the robot in a virtual environment. The image is created using OpenGL to load a CAD model of the robot. Once the CAD has been loaded, we displace the virtual model by using the robot forward kinematics so as to match the actual robot configuration, see Fig. 12(b).

- *Filtered depth image* is an image of the environment containing only the obstacles. It is obtained by subtracting from the *real depth image* the *virtual depth image* of the robot, see Fig. 12(c).
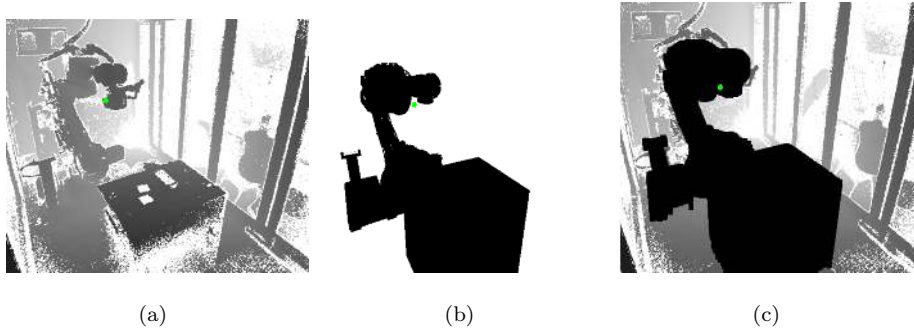

(a)　　　　　　(b)　　　　　　(c)

Figure 12: Real (left), virtual (center), and filtered (right) depth images.

The last filtering process removes the robot from the depth image. Thus, there are only obstacles in the filtered depth image, and the distance algorithm will not be confused by the detection of trivial robot self-collisions or proximities. In order to obtain a more conservative distance estimate, it is common practice to load a slightly expanded CAD model of the robot in the virtual depth image.

The total processing scheme shown in Fig. 13 is illustrated next. The depth sensor provides a new frame of the environment (at the frequency of 30 Hz for a Kinect V2 sensor) and loads the data into the GPU memory. In the meantime, a CAD model of the robot is loaded in a virtual environment, combining the information of the direct kinematics and the capabilities of the OpenGL library.

29

At this stage, matrix transformations are applied in order to obtain a virtual environment point of view that coincides with the depth sensor point of view. In this way, the virtual robot will overlap the real one.
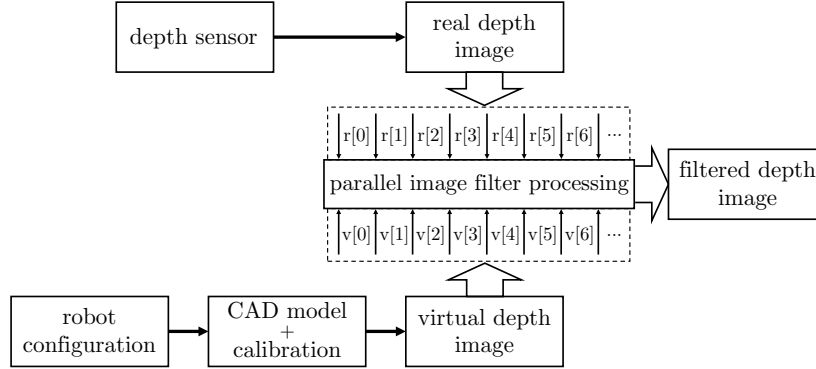

Figure 13: Robot depth images processing scheme.

The first matrix transformation $\boldsymbol{T}_{world}$ maps the coordinates of a point $\boldsymbol{p}_{CAD} = \begin{pmatrix} p_x & p_y & p_z \end{pmatrix}^T$ from a local reference frame (defined in the CAD model) to a world reference frame (often placed at the robot base). Next, a calibration matrix $\boldsymbol{T}_{camera}$ between the world and the camera sensor provides a second transformation to express the world coordinates in the camera frame. A perspective transformation matrix $\boldsymbol{T}_{clip}$ projects then these coordinates into clip-space coordinates. In particular, this transformation determines whether an object is too close to the camera or too far away to be handled. The last transformation $\boldsymbol{T}_{depth}$ determines the depth space coordinates of the point. Summarizing, the coordinates of a point in the virtual depth image are determined by applying the following sequence of transformations to the points of a 3D CAD robot model:

$$\boldsymbol{p}_v = \begin{pmatrix} p_{v,x} \\ p_{v,y} \\ d_v \end{pmatrix} = \boldsymbol{T}_{depth} \cdot \boldsymbol{T}_{clip} \cdot \boldsymbol{T}_{camera} \cdot \boldsymbol{T}_{world} \cdot \boldsymbol{p}_{CAD}, \qquad (4)$$

where $p_{v,x}$ and $p_{v,y}$ are the pixel coordinates in the image plane, and d_v is the corresponding depth.

Once the two 2.5D real and virtual images are ready, they are loaded into the GPU memory as two row vectors with depth information. Components in two vectors having the same index correspond to the same pixels in depth images. A parallel comparison of the depth information for any pair of corresponding components is then performed to filter out the robot. In particular, if a pixel belonging to the robot has a shorter depth than its corresponding pixel in the real depth image, then a maximum depth value is assigned to the corresponding pixel in the filtered depth image. Thus, for each pair of pixel coordinates (x, y) we have

$$d_f(x, y) = \begin{cases} d_r(x, y), & \text{if } d_r(x, y) < d_v(x, y) \\ \max \text{ depth}, & \text{if } d_r(x, y) \geq d_v(x, y), \end{cases} \tag{5}$$

where $d_f(x, y)$, $d_r(x, y)$, and $d_v(x, y)$ are the depth values in pixel coordinates (x, y) of the filtered, real, and virtual depth image, respectively. The image shown in Fig. 12(c) is a typical final result.

6.2. Distance computation

In [11], distances were computed between an obstacle point $\boldsymbol{O}$ and only a finite set of p 'control' points $\boldsymbol{P}_i$, $i = 1, \ldots, p$, distributed along the robot kinematic chain. Relying on a similar method, but exploiting at the same time the parallel computation capabilities of the CUDA architecture, we can now compute distances between *all* robot points $\boldsymbol{P}_D = \left(\begin{array}{ccc} p_{v,x} & p_{v,y} & d_v \end{array} \right)^T$ projected in the virtual depth image and *all* obstacle points $\boldsymbol{O}_D = \left(\begin{array}{ccc} p_{f,x} & p_{f,y} & d_f \end{array} \right)^T$ in the filtered depth image belonging to a *region of surveillance* centered in $\boldsymbol{P}$. Recalling the formulas in [11], we compute the distance

$$d(\boldsymbol{O}, \boldsymbol{P}) = \sqrt{v_x^2 + v_y^2 + v_z^2}, \tag{6}$$

with

$$\begin{aligned} v_x &= \frac{(p_{f,x} - c_x)d_f - (p_{v,x} - c_x)d_v}{f \, s_x} \\ v_y &= \frac{(p_{f,y} - c_y)d_f - (p_{v,y} - c_y)d_v}{f \, s_y} \\ v_z &= d_f - d_v, \end{aligned} \tag{7}$$

31

where $(p_{f,x}, p_{f,y})$ and $(p_{v,x}, p_{v,y})$ are the coordinates in the depth space of the points $\boldsymbol{O}$ and $\boldsymbol{P}$, d_f and d_v are their respective depths w.r.t. the camera, c_x and c_y are the pixel coordinates of the center of the image plane (on the focal axis), f is the focal length of the camera, and s_x and s_y are the dimensions of a pixel in meters. The last five parameters are the camera intrinsic parameters, which can be usually retrieved from the device manufacturer.

Since we don't know how long/deep an obstacle is, two possible cases arise, as illustrated in Fig 14. If the obstacle point has a smaller depth than the point of the robot $(d_f < d_v)$, we assume for later use that the depth of the obstacle is set at $d_f = d_v$, namely the minimum distance with respect to the occluded points in the related gray area is considered, thus in a more conservative way.
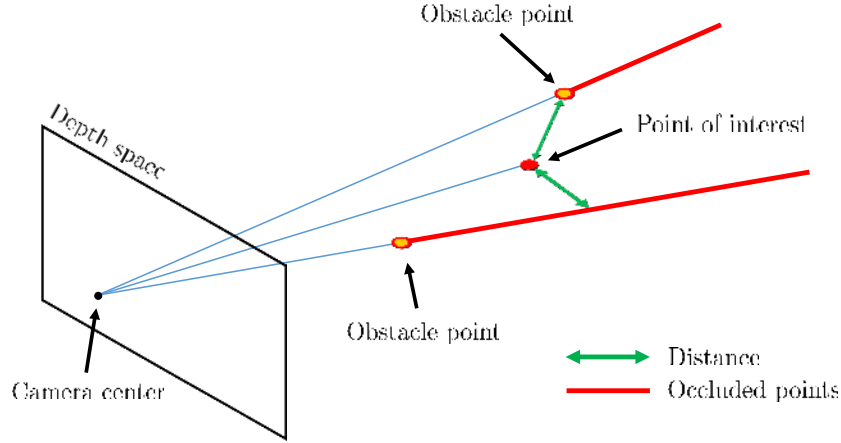

Figure 14: The two possible cases of obstacle point depth d_f smaller or larger than the depth d_v of the point of interest on the robot.

The Cartesian surveillance region, constituted by a cube in 3D space centered at $\boldsymbol{P}$ and with side 2ρ, will have dimensions in the image plane given by

$$x_s = \rho \frac{f s_x}{d_v - \rho}, \qquad y_s = \rho \frac{f s_y}{d_v - \rho}. \tag{8}$$

Thus, the distance evaluation process considers only pixels in the filtered depth image lying in the area

$$\boldsymbol{S} = \left[p_{f,x} - \frac{x_s}{2}, p_{f,x} + \frac{x_s}{2}\right] \times \left[p_{f,y} - \frac{y_s}{2}, p_{f,y} + \frac{y_s}{2}\right]. \tag{9}$$

32

Since computations for each pixel in the filtered image are completely independent, distances can be evaluated concurrently by each GPU thread, speeding up the algorithm. The entire processing is illustrated in Fig. 15. Note also that distance evaluations can be made also at a faster rate than depth sensor measurement frequency, rather as soon as a new robot position data is made available. In [56], with measured robot configurations coming in at 200 Hz, the algorithm was shown to run at about the same frequency (170 Hz).
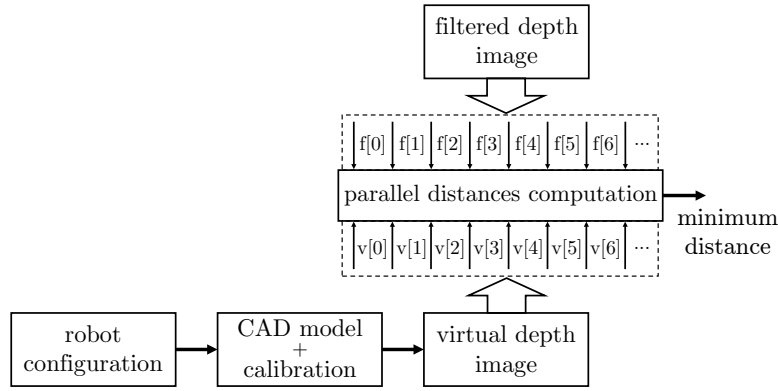

Figure 15: Distance computation processing scheme.

When robot-obstacle distances have been evaluated, they can be used to slow down (or possibly stop) the robot in order to avoid collision with obstacles and ensure a safe coexistence with humans. Associated to the distance between the obstacle point $\boldsymbol{O}$ and a robot point $\boldsymbol{P}$, as obtained from (6–7), a velocity scaling factor $0 \leq \alpha \leq 1$ can be defined as

$$
\alpha = \begin{cases} \dfrac{|d_{near}(\boldsymbol{O}, \boldsymbol{P})| - d_{min}}{d_{max} - d_{min}}, & \text{if } d_{min} \leq |d_{near}(\boldsymbol{O}, \boldsymbol{P})| \leq d_{max} \\ 1, & \text{if } |d_{near}(\boldsymbol{O}, \boldsymbol{P})| > d_{max}, \end{cases} \tag{10}
$$

where $d_{near} = min_{\boldsymbol{O} \in \boldsymbol{S}}\, d(\boldsymbol{O}, \boldsymbol{P})$ is the local minimum distance from an obstacle in the explored area $\boldsymbol{S}$ in (9), $d_{min} \geq 0$ is the minimum safe distance allowed before the robot should stop, and d_{max} is the distance beyond which the robot velocity needs not to be scaled. The safe distance d_{min} can also be modified

33

online as a function of the robot tool speed.

Figure 16 shows the results obtained during a coexistence experiment lasting 8 seconds. When the human approaches the robot inside the cell, the algorithm returns distances between the robot and the moving (human) obstacle for each RGB-D sensor. The plotted quantities $d_1(t)$ and $d_2(t)$ are the computed evolutions of the two minima among all distances evaluated (at time t) using the depth images, respectively from the left and right Kinects in the cell. The horizontal line (in red) is the threshold $d_{max} = 0.25$ m below which the robot velocity will be reduced.

6.3. Handling safety issues of depth sensors

We consider next safety-related issues resulting from a failure of one (or both) of the depth sensors present inside the cell and/or of the associated algorithm devoted to compute in real time the distance between the robot and the human. Potential safety-related issues can be raised by:

- depth sensor failures, as caused by hardware/driver faults, cable disconnection, severe camera occlusions or bad lighting conditions;

- distance computation algorithm failures, as a result of a bad geometric filtering of the robot CAD model from the depth image (e.g., due to a poor calibration between robot and sensors), bad quality of the communication between the robot and the PC hosting the algorithm (e.g., Ethernet cable unplugged), or excessive noise in the Kinect data.

Software countermeasures have been implemented to improve the robustness of the measurement chain and of the algorithm used for cell monitoring. A possible device failure can be recognized by introducing a further image processing step. In fact, when the sensor stops working or the cable is unplugged, the provided image will no longer change and its refresh rate F goes to zero. Thus, a hardware failure will be detected as soon as the image refresh rate $F < F_{th}$, i.e., drops below a suitably tuned threshold $F_{th} > 0$. Moreover, depth sensors provide 'black' pixels when their associated depth values are not valid, namely,
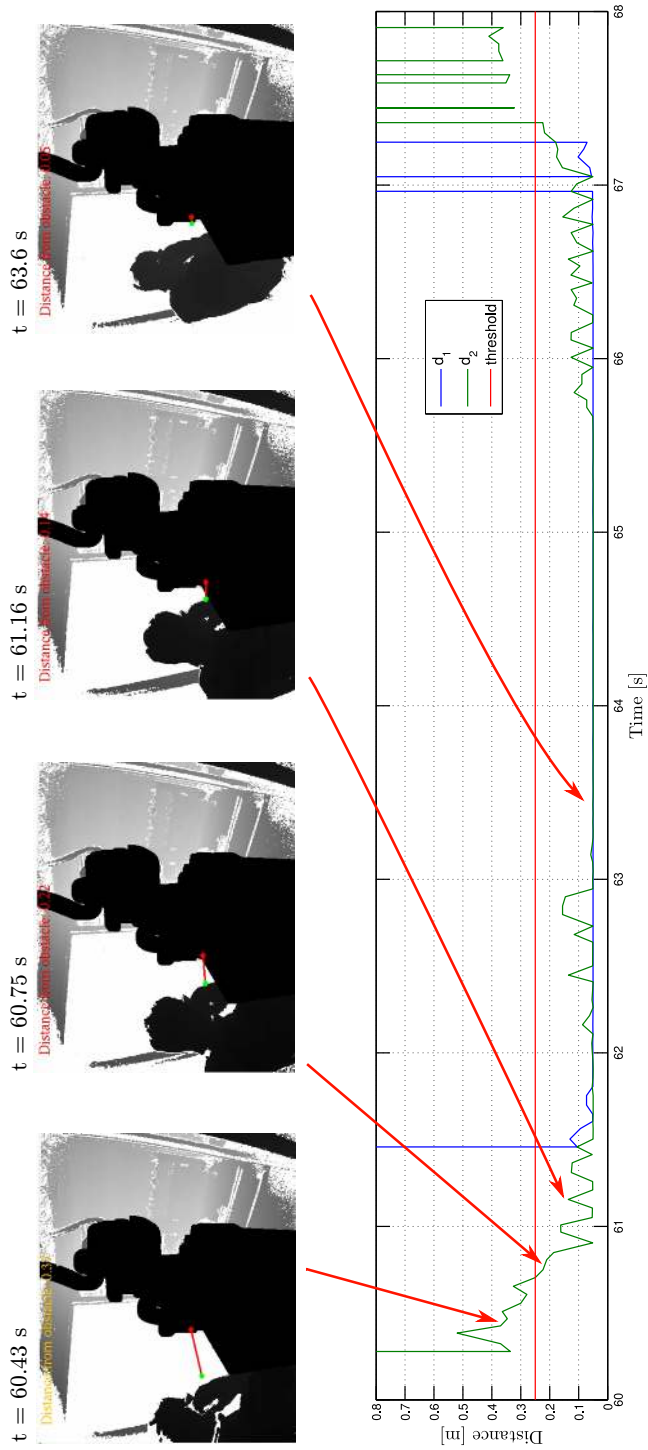
34

Figure 16: Minimum distances between robot and human during the coexistence experiment, as computed by the algorithm with the left (blue line) and the right (green line) Kinect. At the top, snapshots of the depth views with actual minimum distances of the right depth sensor.

when the camera is severely occluded or in bad lighting conditions. In such cases, the occlusion is detected as soon as the number of black pixels p_B in the image I becomes larger than a threshold $n_{th} > 0$, i.e., $n_B = \sum_I p_B \geq n_{th}$.

As for failures of the distance computation algorithm, an uncalibrated sensor, a bad calibration procedure, or a wrong CAD model lead to inappropriate filtering of the robot from the depth image. Typically, this will produce a (minimum) distance signal with several discontinuities over short intervals of time, which can be detected by monitoring and averaging the last few distance samples. A similar effect is produced if the robot controller (see Fig. 10) is not providing timely the correct values of the robot joint angles to the distance algorithm[1]. In any event, the PROFIsafe protocol, through which the robot and the host PC are connected, will ensure that the robot stops in case of bad communications.

As soon as a fault is detected in any of the above situations, an emergency signal can be sent to the Programmable Logic Controller (PLC) of the robotic cell, which will in turn immediately stop the robot. However, if the robot enters in such an emergency stop, the user will need to restart the entire process. To avoid this, the two additional laser scanners present in the cell are used in order to understand in which part of the working area the human is located, and possibly prevent an unnecessary emergency stop when the risk of collision is still negligible. We remark that, when the internal Kinect sensors or the software interpreting their data fail, the laser scanners will be unaffected (see also the communication paths in Fig. 10. The robot will eventually stop if the human position determined by the scanners is critical.

Figure 17 shows how safety issues raised by failures in the depth sensing and processing are handled. A speed scaling strategy is adopted when the Kinect

---

[1]Such faulty behavior would lead to more parts of the robot being identified as obstacles, possibly resulting in the activation of the SSM robot stopping function. This suggests that also robot joint angles should be part of the safety system, though not all robot vendors at present do so in their safety interfaces.
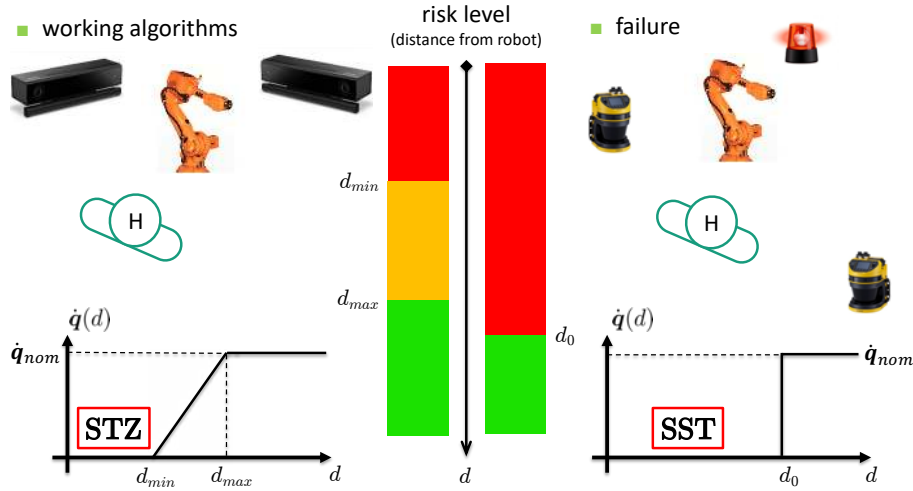
Figure 17: Safe handling of depth sensors. Left: Speed scaling solution with working Kinect cameras; the robot slows down when the minimum distance to obstacles is $d \in (d_{min}, d_{max}]$, and eventually stops using the Safe Tool Zone function when $d \le d_{max}$. Right: Solution with laser scanners when the Kinect cameras fail; the robot enters in emergency stop through the Safe Stand Still function as soon as the laser scanners detect a human in the cell ($d \le d_0$).

depth cameras are working properly. The robot speed is scaled down when human distance from the robot decreases[2], or is set to zero if the human is too close to the robot. We note that in this case the robot is stopped using the Safe Tool Zone (STZ) function provided by ABB SafeModule [36], which will not lead to an emergency stop. However, when a Kinect failure is recognized and the laser scanners detect a human in the cell, the robot will be forced to enter in an emergency stop using the Safe Stand Still (SST) function, also provided by the SafeModule.

---

[2]Sometimes the robot speed is not scaled down continuously in the intermediate range of distances $(d_{min}, d_{max}]$, but rather set in the entire interval at a constant speed d_{int} which is slower than the nominal one —see the experiments in Section 6.4.

37

6.4. Experimental results

We present here experimental results on human-robot coexistence and on communication via gestural commands, directly collected on the SYMPLEXITY cell while an Abrasive Finishing (AF) task was under preparation. Indeed, no contacts or other physical human-robot collaboration are allowed in this case. Other representative results of the same type are shown in the video accompanying the paper.

In the chosen scenario, a human operator activates the coexistence phase by a suitable gesture (i.e., opening the cell doors), and then enters in the AF cell where a 6R ABB-IRB 4600-60 robot, commanded using its native code, is in motion. The workspace is monitored by two Microsoft$^®$ Kinect V2 depth sensors, positioned at a distance of about 3 m facing the robot. The Kinects provide 512×424 depth images at 30 Hz rate. The hardware platform that performs all the needed computations is a 64-bit Intel$^®$ Core i7 CPU, equipped with 16GB DDR3 RAM. The implementation of the real-time distance computation runs on a high performance graphic card with a NVIDIA GTX1070 GPU, organized in 1920 CUDA cores and capable of 30720 concurrent threads. In (8), we have set the parameter $\rho = 0.5$ m, defining a Cartesian surveillance region constituted by a cube with a side of about 1 m. The complete algorithm runs in $3 \div 4$ ms, depending on the relative robot-human position. The host PC and the robot are connected through a PLC using the Ethernet communication protocol of the ABB controller. The minimum and maximum distance in (10) have been set to $d_{min} = 0.05$ m and $d_{max} = 0.25$ m, respectively.

In the first experiment, when the human enters the cell, the robot end-effector is moving with a Cartesian speed of $v_{nom} = 50$ mm/s. With reference to Fig. 18, when the distance between the human and the robot becomes smaller than the set threshold, i.e., $d_{near} = min(d_1, d_2) \leq 0.25$ m, the monitoring algorithm detects this situation issuing a signal and slows down the TCP to a constant speed $v_{slow} = 20$ mm/s, reached in about 550 ms. Moreover, if the distance decreases further, i.e., $d_{near} = min(d_1, d_2) \leq 0.05$ m, the monitoring algorithm activates the Safe Tool Zone (STZ) function of the ABB SafeModule,
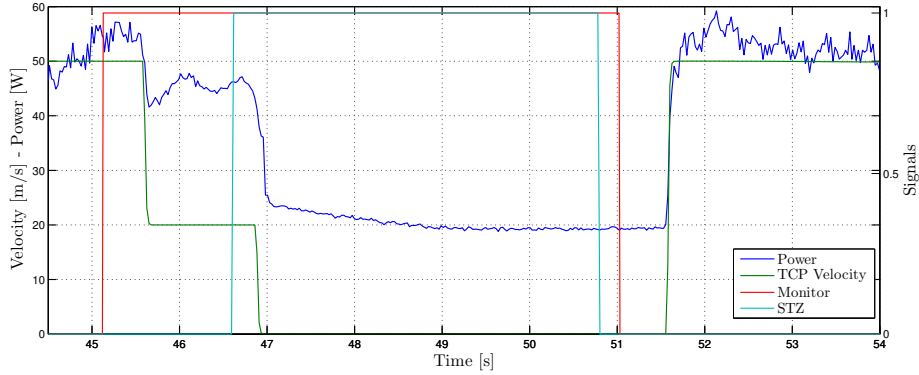
38

Figure 18: Coexistence experiment with working Kinect cameras. As soon as $d_{near} \leq 0.25$, the monitoring algorithm activates a signal (red line) which slows down the TCP speed (green line) to $v_{slow} = 20$ mm/s. When $d_{near} \leq 0.05$, the Safe Tool Zone (STZ - light blue line) function is activated and the robot stops. The robot does not activate the motor brakes and the power (blue line) remains always different from zero.

with a 'maximum allowed' velocity of 0 mm/s. The outcome is that the robot stops (after a delay of about 360 ms) without entering in an emergency stop. This is shown by the robot power (blue line) which remains different from zero; this is the power needed to compensate at rest the gravity acting on the manip-⁷⁰⁰ ulator, since motor brakes have not been activated. As soon as the human is far enough, the robot resumes autonomously the task (at the instant $t = 51.5$ s).

In the second experiment, the Kinect cameras have been disabled to emulate a device/algorithm fault. As in the previous case, the human enters the AF cell while the robot is in motion with a faster end-effector speed $v_{nom} = 100$ mm/s. ⁷⁰⁵ As shown in Fig. 19, the laser scanners detect the presence of a human (at $t = 29.66$ s) and activates the Safe Stand Still (SST) emergency function using the SafeModule. The robot enters in emergency stop, and the TCP speed drops to 0 mm/s after a delay of about 100 ms. However, here the power goes to zero as well, due to the automatic activation of the motor brakes by the ABB ⁷¹⁰ controller. In contrast to the previous experiment, the user needs then to restart the robot controller in order to resume the original task.

In order to evaluate the performance in terms of delays between obstacle
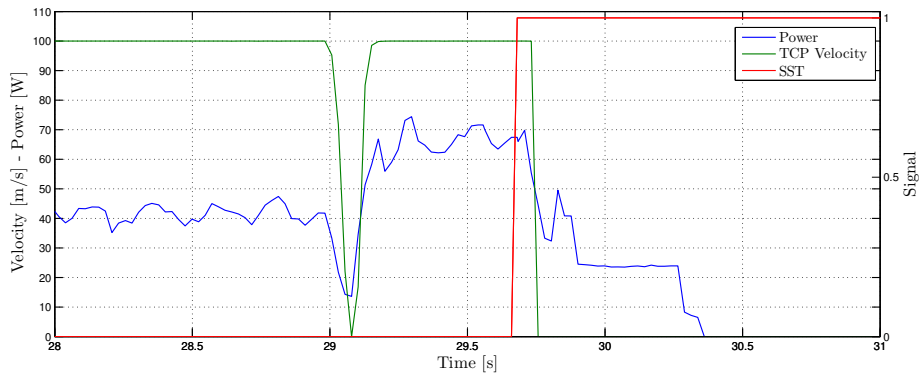
39

Figure 19: Coexistence experiment emulating a failure of the Kinects. As soon as the laser scanners detect an obstacle, the cell PLC sends a signal (red line) to the SafeModule which activates the Safe Stand Still (SST) function (light blue line). The robot enters in emergency stop activating the motor brakes. The TCP speed (green line) as well as the power (blue line) drop both to zero.

detection and robot reaction, different coexistence experiments have been performed using the previous setup, with and without using Kinect cameras. The ₇₁₅ main numerical results regarding the average time delays are given in Tab. 1 and Tab. 2., respectively. The measured reaction times from full TCP speed until robot stop when using the laser scanners are consistent with the known $10 - 30$ ms of latency between signal activation and start of a deceleration in the ABB SafeMove modules.

TCP speed [mm/s]	average [s]
$100 \rightarrow 20$	0.553
$50 \rightarrow 20$	0.537
$20 \rightarrow 0$	0.339

Table 1: Average time delays to slow down or stop the robot, with working Kinect cameras.

TCP speed [mm/s]	average [s]
$100 \rightarrow 0$	0.081

Table 2: Average time delay to stop the robot, using laser scanners.

40

7. Human-Machine Interface

The Human-Machine Interface (HMI) that governs the SYMPLEXITY cell is composed of three parts: the pre-process interface, the in-process interface, and the after-process interface. The general layout is shown in Fig. 20. In particular:
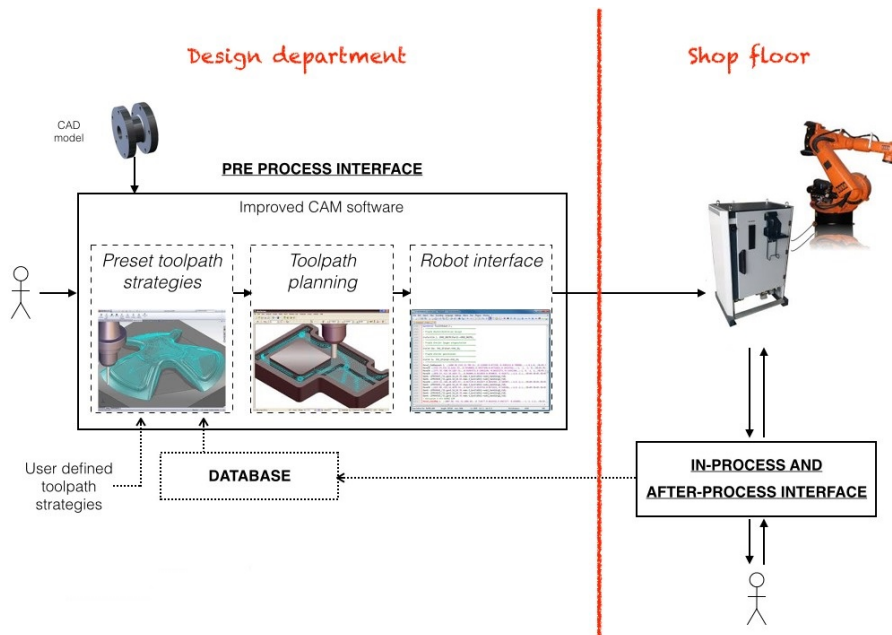

Figure 20: Layout of the SYMPLEXITY HMI system.

- The *pre-process interface* consists of an offline software that is used by CAM specialists to automatically compute the nominal toolpath strategy for each polishing process (AF, FP, LP) or for QA. The main software consists of using PowerMill by AUTODESK [57] with toolpath and tools adapted to polishing and contactless measurements. There is also a plugin to prototype the specific features of SYMPLEXITY, to drive the robot cells and its different processes. Additional features of the pre-process include a live view of the robot movements, smartwatch and smartphone notifications, and augmented reality for visualization of QA results.

41

- The *in-process interface* runs on board of the robotic cell and is employed to start the polishing process and have an on-line feedback of the processing (Fig. 21). The main functionalities of the in-process interface are:

  - Start of a new polishing session. At the beginning of each polishing session, the operator uploads the toolpath strategy provided by the pre-process interface. All the polishing steps that have to be performed are shown within the in-process interface.

  - Selection of the execution modality. The operator can choose if the the polishing steps have to be executed in cascade or if only one or more steps have to be performed individually.

  - Visualization of the relevant (editable and not editable) parameters for the current polishing step. The editable parameters can be changed by the operator.

  - Visualization of the workpiece surface that is undergoing the current polishing step.

  - Availability of the tool in the warehouse. If the tool requested for the execution of a polishing step is missing, the start of the step is prevented and an indication of the missing tools is shown to the operator.

  - Visualization of the state of the cell doors (open/closed) depending on the state of the polishing task.

  - Intervention by the user. The user can decide to perform a polishing step manually or to pause the process. In the first case, a collaborative human-robot operation can be started and the HMI will support the safe monitoring of the collaborative steps in execution (see [58] for more details).

  - Visualization of the QA results.

  - Log of parameters and visualization of messages related to information, warning or errors into the process.

42

Figure 21: Overview of the in-process interface for the abrasive finishing process.

- At the end of the polishing, the *after-process interface* displays a summary of the task performed, including the surface QA. The interface allows to access the QA data files and to interactively display the evaluated parameters for the whole measured area or selected (zoomed) parts. It visualizes measurement results, thresholds for go/no-go decisions, and indications on the final quality value, generating different levels of reports.

- The three interfaces are interconnected through a *data management* system that collects all the data related to the task. In particular, the pre-process interface interacts with the database to save the nominal toolpath strategy that will be used in the polishing process and to query for suggestions about possible polishing strategies. Indeed, the HMI of the database has been designed to provide main functionalities that allow users to:

  - add new entries in the database, store there results from the pre-process phase, and retrieve them for a new polishing session; when the operator selects the desired module or list of modules to be executed, the corresponding procedural modules are loaded into the robot controller and the robot executes the polishing process;

43

— obtain suggestions from the database about the combination of values for the polishing parameters that best matches the input part to be polished and the desired final quality. Depending on the current quality value detected on the piece and the target final quality, different polishing strategies are suggested.

7.1. Gesture Recognition

Communication between the user and the robotic cell can also be established using simple gestures by the operator. The gesture recognition provides indeed a safe contactless interaction with the robotic cell, safety being structurally guaranteed by the cabin dividing the human and the robot. The tool for gesture recognition has been included in the HMI.

Two (Kinect) depth sensor devices have been placed outside the robotic cell, each above and in correspondence to one of the two opening doors, see (**#17**) in Fig. 6. This placement allows to easily track the body position of the user and recognize the open or closed state of both hands (drawn respectively as a green or red circles in Fig. 22). The skeleton tracking feature of the Kinect is used and optimized to locate and distinguish users that are standing or sitting while facing the Kinect, and to follow then their actions. No specific pose or calibration action needs to be taken for a user to be tracked.

Recognition of gestures can then be performed, by choosing for instance the following set of intuitive gestures (Fig. 22):

- Gestures for activating (Fig. 22(a)) and deactivating (Fig. 22(b)) communication, which are needed to avoid false positives in the recognition of unintentional gestures (e.g., of people that are simply walking in the surrounding area of the camera range).

- Gestures to start (Fig. 22(c)) and stop (Fig. 22(d)) robot motion during the execution of a task program. The action consists in raising both hands up, with their common state being respectively open (to start) or closed (to stop).
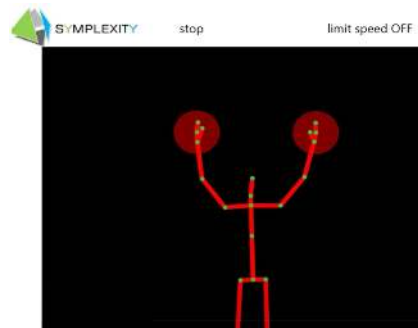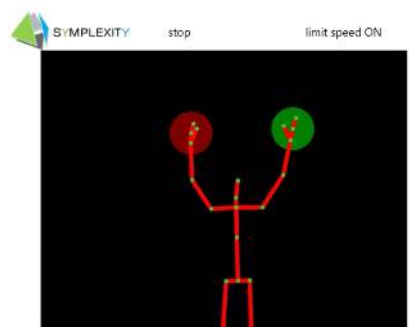
44

(a) Activating gesture

(b) Deactivating gesture
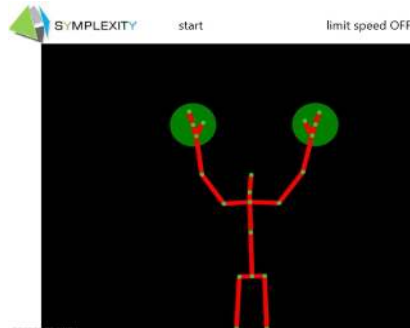
(c) Gesture for starting robot motion

(d) Gesture for stopping robot motion

(e) Gesture for turning ON limitation of speed

(f) Gesture for turning OFF limitation of speed

Figure 22: Gestures selected for the contactless interaction between operator and robotic cell (green circle = open hand, red circle = closed hand).

- Gestures to turn on (Fig. 22(e)) and turn off (Fig. 22(f)) the limitation
₈₁₀ on the robot end-effector speed, in relation to a threshold value chosen
a priori. This is achieved when both hands are raised up, as before, but
with opposite states: speed limitation is turned on when the right hand is
open and the left is closed, while is turned off in the mirrored situation.
This particular hand position has to be maintained for about five seconds
₈₁₅ to prevent ambiguities with unintentional movements.

Figure 23: Examples of gestures being performed by a user to command the robot.

These gestures (or other equally intuitive) guarantees high performance in
the interpretation of the desired commands by the operator. The proposed
method has been tested with several users (see, e.g., Fig. 23) at various dis-
tances from the RGB-D camera, obtaining always satisfactory results. The
₈₂₀ effectiveness of the recognition method can also be appreciated in the last part
of the accompanying video clip. However, the lighting-related conditions could
be a limiting factor in gesture recognition. In particular, light sources contain-

46

ing infrared components, e.g., with high sunlight exposure, introduce (further) noise in the depth images retrieved by the Kinect sensor, leading to a loss of performance.

8. Conclusions

Human-robot interaction and collaborative robotics are major trends in today's robot technology that enable solutions in which the operator can directly interacts with the robot, as opposed to traditional safeguarding where access to the robot system is prevented in most of the operational conditions. Interaction with access to the workspace and coexistence of human and robot clearly require safety principles to be implemented, like specific energy-limited robots or adding sensors and control layers to standard industrial robot systems.

In this paper, we designed a safety framework to ensure coexistence of an operator in a robotic cell in which a standard industrial robot is in motion. In particular, we defined safety measures and requirements to allow coexistence of operators during the execution of selected phases of a polishing operation of metallic surfaces, based on abrasive finishing and fluid jet polishing technologies. The concept was enriched by the actual implementation of the software and hardware architecture into the manufacturing application, including also a smart human-machine interface monitoring the entire process.

While physical human-robot contacts are not allowed by the polishing task, an innovative coexistence modality and human-robot communication with gestural commands were demonstrated for the collaborative phases of setup operations of the cell/tools and quality assessment of the polished workpiece. The human in intervention is activated by gestures. In the cell, two RGB-Depth sensors are devoted to monitor the workspace and compute online distances between robot and human, both in motion. The computational algorithm works very efficiently in the so-called depth space of the sensors and returns distances that are used in real time to slow down (and possibly stop) the robot, so as to avoid collision with any dynamic obstacle. To enhance the level of safety in human-robot

47

coexistence, and with the aim of protecting the operator from faulty behaviour of the depth sensors, we integrated the depth-space monitoring algorithm with redundant sensing hardware, i.e., two laser scanners working in parallel in the cell, established communications through a safe channel (PROFISafe), and used a safe-oriented proprietary robot control software (SafeMove by ABB). The performance obtained was very satisfactory on all tests. We note, however, that the implemented cell is still a research project, and not all safety functions have achieved the safety performance requirements of the industrial robot safety standards.

In closing, we mention that a second robotic cell was developed within our SYMPLEXITY project, in which a more extended polishing task with physical human-robot interaction and controlled force exchange has been implemented, as described in [58]. A different technology was used there, namely Laser Polishing, which requires a sufficiently good quality of the workpiece surface to start with. To this end, an external station was equipped with a lightweight UR10 robot mounting a force/torque sensor and with a metrology system. When needed, the operator can safely perform manual pre-polishing on the part held by the robot, physically pushing/pulling on the robot body so as to reorient the workpiece as desired.

References

References

[1] M. Hägele, K. Nilsson, J. Pires, R. Bischoff, Industrial robotics, in: B. Siciliano, O. Khatib (Eds.), Handbook of Robotics (2nd Ed.), Springer, 2016, pp. 1385–1421.

[2] S. Ong, A. Yew, N. Thanigaivel, A. Nee, Augmented reality-assisted robot programming system for industrial applications, Robotics and Computer-Integrated Manufacturing 61 (2020) 101820.

[3] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, P. Fraisse, Collaborative manufacturing with physical human–robot interaction, Robotics and Computer-Integrated Manufacturing 40 (2016) 1 – 13.

[4] F. Ferraguti, F. Pini, T. Gale, F. Messmer, C. Storchi, F. Leali, C. Fantuzzi, Augmented reality based approach for on-line quality assessment of polished surfaces, Robotics and Computer-Integrated Manufacturing 59 (2019) 158 – 167.

[5] Y. Lu, Industry 4.0: A survey on technologies, applications and open research issues, J. of Industrial Information Integration 6 (2017) 1–10.

[6] S. Robla-Gomez, V. Becerra, J. Llata, E. Gonzalaz-Sarabia, C. Torre-Ferrero, J. Perez-Oria, Working together: A review on safe human-robot collaboration in industrial environments, IEEE Access 5 (2017) 26754–26773.

[7] A. De Santis, B. Siciliano, A. De Luca, A. Bicchi, An atlas of physical human-robot interaction, Mechanism and Machine Theory 43 (3) (2008) 253–270.

[8] S. Haddadin, S. Parusel, R. Belder, A. Albu-Schäffer, It is (almost) all about human safety: A novel paradigm for robot design, control, and planning, in: F. Bitsch, J. Guiochet, M. Kaâniche (Eds.), Computer Safety, Reliability, and Security (SAFECOMP 2013), Vol. 8153 of Lecture Notes in Computer Science, Springer, 2016, pp. 202–215.

[9] M. Bdiwi, M. Pfeifer, A. Sterzing, A new strategy for ensuring human safety during various levels of interaction with industrial robots, CIRP Ann. - Manufacturing Technology 66 (1) (2017) 453–456.

[10] P. Lasota, G. Rossano, J. Shah, Toward safe close-proximity human-robot interaction with standard industrial robots, in: Proc. IEEE Int. Conf. on Automation Science and Engineering, 2014, pp. 339–344.

49

[11] F. Flacco, T. Kröger, A. De Luca, O. Khatib, A depth space approach for evaluating distance to objects – with application to human-robot collision avoidance, J. of Intelligent and Robotic Systems 80 (Suppl. 1) (2015) 7–22.

[12] O. Mazhar, B. Navarro, S. Ramdani, R. Passama, A. Cherubini, A real-time human-robot interaction framework with robust background invariant hand gesture detection, Robotics and Computer-Integrated Manufacturing 60 (2019) 34 – 48.

[13] A. De Luca, F. Flacco, Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration, in: Proc. IEEE Int. Conf. on Biomedical Robotics and Biomechatronics, 2012, pp. 288–295.

[14] S. Haddadin, A. Albu-Schäffer, A. De Luca, G. Hirzinger, Collision detection and reaction: A contribution to safe physical human-robot interaction, in: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2008, pp. 3356–3363.

[15] S. Haddadin, A. De Luca, A. Albu-Schäffer, Robot collisions: A survey on detection, isolation, and identification,, IEEE Trans. on Robotics 33 (2) (2017) 1292–1312.

[16] F. Flacco, A. De Luca, Real-time computation of distance to dynamic obstacles with multiple depth sensors, IEEE Robotics and Automation Lett. 2 (1) (2017) 56–63.

[17] E. Magrini, F. Flacco, A. De Luca, Control of generalized contact motion and force in physical human-robot interaction, in: Proc. IEEE Int. Conf. on Robotics and Automation, 2015, pp. 2298–2304.

[18] I. Maurtua, A. Ibarguren, J. Kildal, L. Susperregi, B. Sierra, Human–robot collaboration in industrial applications: Safety, interaction and trust, Int. J. of Advanced Robotic Systems 14 (4) (Jul-Aug 2017) 1–10.

50

[19] D. Gorecky, M. Schmitt, M. Loskyll, D. Zülke, Human-machine-interaction in the Industry 4.0 era, in: Proc. 12th IEEE Int. Conf. on Industrial Informatics, 2014, pp. 289–294.

[20] A. Mohammed, B. Schmidt, L. Wang, Active collision avoidance for human-robot collaboration driven by vision sensors, Int. J. of Computer Integrated Manufacturing 30 (9) (2017) 970–980.

[21] P. Long, C. Chevallereau, D. Chablat, A. Girin, An industrial security system for human-robot coexistence, Industrial Robot 45 (2) (2018) 220–226.

[22] ISO 10218-1-2011, Robots and robotic devices – Safety requirements for industrial robots. Part 1: Robots (since July 1, 2011).
URL http://www.iso.org

[23] ISO 10218-2-2011, Robots and robotic devices – Safety requirements for industrial robots. Part 2: Robot systems and integration (since July 1, 2011).
URL http://www.iso.org

[24] ISO TS 15066:2016, Robots and robotic devices – Collaborative robots (since February 15, 2016).
URL http://www.iso.org

[25] SYMPLEXITY, Symbiotic Human-Robot Solutions for Complex Surface Finishing Operations.
URL www.symplexity.eu

[26] R. Bogue, Finishing robots: A review of technologies and applications, Industrial Robot 36 (1) (2009) 6–12.

[27] E. Kalt, R. Monfared, M. Jackson, Development of an intelligent automated polishing system, in: Proc. 16th Int. Conf. of the European Society for Precision Engineering and Nanotechnology (EUSPEN 2016), 2016.
URL dspace.lboro.ac.uk/2134/21696

[28] O. Borisov, V. Gromov, S. Kolyubin, A. Pyrkin, A. Bobtsov, V. Salikhov, A. Klyunin, I. Petranevsky, Human-free robotic automation of industrial operations, in: Proc. 42nd Ann. Conf. of the IEEE Industrial Electronics Society, 2016, pp. 6867–6872.

[29] A. Kharidege, D. Ting, Z. Yajun, A practical approach for automated polishing system of free-form surface path generation based on industrial arm robot, Int. J. of Advanced Manufacturing Technology 93 (9–12) (2017) 3921–3934.

[30] I. Ferralli, T. Blalock, M. Brunelle, T. Lynch, B. Myer, K. Medicus, Manufacturing and metrology for IR conformal windows and domes, in: Proc. of SPIE, Vol. 10179 (Window and Dome Technologies and Materials XV), 2017.

[31] P. Chotiprayanakul, D. Liu, G. Dissanayake, Human-robot-environment interaction interface for robotic grit-blasting of complex steel bridges, Automation in Construction 27 (2012) 11–23.

[32] F. Vicentini, N. Pedrocchi, L. Molinari Tosatti, Safenet of unsafe devices – extending the robot safety in collaborative workspaces, in: Proc. 10th IEEE Int. Conf. on Informatics in Control, Automation and Robotics, 2013, pp. 276–283.

[33] F. Platbrood, O. Gornemann, Safe robotics - Safety in collaborative robot systems, SICK AG, Waldkirch, Germany (2017).

[34] A. Vysocky, P. Novak, Human-robot collaboration in industry, Modern Machinery Science J. (2016) 903–906.

[35] S. Kock, J. Bredahl, P. J. Eriksson, M. Myhr, K. Behnisch, Taming the robot, ABB Review 4 (2006) 11–14.

[36] ABB - SafeMove2 Product Overview.
    URL   https://library.e.abb.com/public/

```
381b5a79303249e2ae2fd654e0f0dab7/ABB%20SafeMove2%20-%
20Overview%20-%20External%202017.pdf
```

[37] F. Ferraguti, C. Talignani Landi, C. Secchi, C. Fantuzzi, M. Nolli, M. Pe-
samosca, Walk-through programming for industrial applications, Procedia
Manufacturing 11 (2017) 31–38.

[38] J. A. Marvel, R. Norcross, Implementing speed and separation monitoring
in collaborative robot workcells, Robotics and Computer-Integrated Man-
ufacturing 44 (2017) 144 – 155.

[39] C. Byner, B. Matthias, H. Ding, Dynamic speed and separation monitoring
for collaborative robot applications – concepts and performance, Robotics
and Computer-Integrated Manufacturing 58 (2019) 239 – 252.

[40] S. Haddadin, Physical safety in robotics, in: R. Drechsler, U. Kühne (Eds.),
Formal Modeling and Verification of Cyber-Physical Systems, Springer,
2015, pp. 249–271.

[41] P. Aivaliotis, S. Aivaliotis, C. Gkournelos, K. Kokkalis, G. Michalos,
S. Makris, Power and force limiting on industrial robots for human-robot
collaboration, Robotics and Computer-Integrated Manufacturing 59 (2019)
346 – 360.

[42] A. D. Wilbert, B. Behrens, C. Zymla, O. Dambon, F. Klocke, Robotic fin-
ishing process - an extrusion die case study, CIRP Journal of Manufacturing
Science and Technology 11 (2015) 45–52.

[43] V. Grün, R. Börret, M. Speich, D. Wiedemann, Fluid jet polishing of steel
moulds, in: Proc. of SPIE - The International Society for Optical Engi-
neering, Vol. 10009, 2016.

[44] L. Baath, B. Rosen, Towards new interferometer technology for surface
metrology, in: Proc. of the 12th Int. Conf. of the European Society for
Precision Engineering and Nanotechnology, Vol. 1, 2012, pp. 158–161.

[45] J. Flemmer, I. Ross, E. Willenborg, H. Fröba, Machine tool and cam-nc data chain for laser polishing complex shaped parts, Advanced Engineering Materials 17 (3) (2015) 260–267.

[46] C. Nüsser, J. Kumstel, T. Kiedrowski, A. Diatlov, E. Willenborg, Process- and material-induced surface structures during laser polishing, Advanced Engineering Materials 17 (3) (2015) 268–277.

[47] F. Leali, F. Pini, M. Ansaloni, Integration of cam off-line programming in robot high-accuracy machining, in: IEEE/SICE Int. Symposium on System Integration, SII 2013, 2013, pp. 580–585.

[48] V. Villani, F. Pini, F. Leali, C. Secchi, Survey on human-robot collaboration in industrial settings: Safety, intuitive interfaces and applications, Mechatronics 55 (2018) 248–266.

[49] F. Pini, F. Leali, M. Ansaloni, Offline workpiece calibration method for robotic reconfigurable machining platform, in: 19th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2014, 2014.

[50] F. Leali, M. Pellicciari, F. Pini, A. Vergnano, G. Berselli, A calibration method for the integrated design of finishing robotic workcells in the aerospace industry, Communications in Computer and Information Science 371 (2) (2013) 37–48.

[51] Major Robot OEMs Fast-Tracking Cobots.
URL https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/Major-Robot-OEMs-Fast-Tracking-Cobots/content_id/5033

[52] C. Goldsberry, Worker safety remains an important ingredient in automated systems, Welding Design and Fabrication 81 (2).

[53] FANUC Safety functions for robots.
URL https://www.fanuc.eu/uk/en/robots/accessories/safety

[54] KUKA Safe Operation.
URL https://www.kuka.com/en-de/products/robot-systems/software/hub-technologies/kuka_safeoperation

[55] J. Rofar, M. Franekova, Functional safety specification of communication profile - PROFIsafe, Advances in Electrical and Electronic Engineering 5 (1) (2006) 158–161.

[56] E. Magrini, A. De Luca, Human-robot coexistence and contact handling with redundant robots, in: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2017, pp. 4611–4617.

[57] AUTODESK PowerMill: CAM software solution for high-speed machining and multi-axis CNC.
URL https://www.autodesk.com>products>powermill>features

[58] C. Gaz, E. Magrini, A. De Luca, A model-based residual approach for human-robot collaboration during manual polishing operations, Mechatronics 55 (2018) 234–247.