POLITECNICO DI TORINO

Master of Science in Mechanical Engineering

Master of Science Thesis

Human-Robot Collaboration in Automotive Systems: Properties and Scenarios of Applications

Supervisor

Prof. Eugenio Brusa

Candidate

Antonio Santarpino

Centro Ricerche Fiat

Company tutor

Dott. Giulio Vivo

Academic year 2019/2020

INDEX

ABSTRACT

This thesis has been developed in collaboration with Centro Ricerche Fiat (CRF) S.C.p.A, Orbassano (TO), on site with supervision of my company tutor Dott. Giulio Vivo, as part of the Research & Factory Innovation department of the WCM R&I area, and with the advices and supervision of my academic supervisor Prof. Eugenio Brusa.

After three months of internship dedicated to the introduction to Human-Robot Collaboration concept, to some electronic basis and to the progression of the balance shafts insertion project, the activities target was to implement a Human-Machine Interaction within the cycle in order to control the gripping phase performed by the robot.

In the following pages is presented an introduction to HRC, focusing on the benefits deriving from the employment of collaborative robots in assembly lines, especially for industrial car production, by means of different models analyzed, considering that the concept is still in evolution. A summary of the ISO/TS 15066 also is included, to describe the technical specifications and safety standards introduced specifically for the collaboration.

Finally the implementation of the HMI is described, explaining the requirements on which is based and the content of the work, starting from the description of the workstations in the HRC Laboratory, passing through the progression steps, which are the signal analysis of the robot joints torque sensors and the developing of the algorithm for the interaction detection.

1 INTRODUCTION

1.1 HUMAN-ROBOT COLLABORATION

The HRC concept is based on the combination of robot strength, repeatability, precision, velocity and predictability together with human skills and intelligence in order to obtain a hybrid solution that guarantees safety and continuous cooperation between autonomous robotic systems and operators.

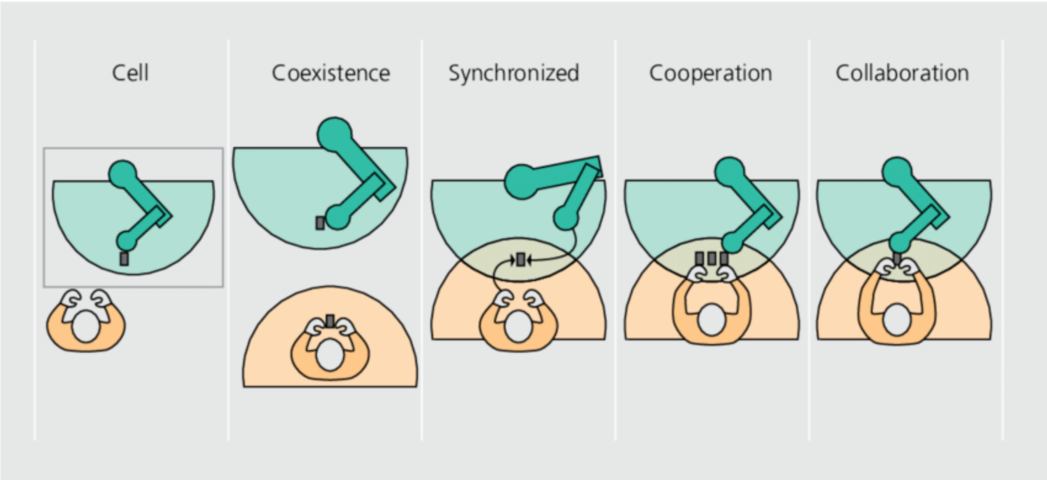

Fig. 1.1 - Levels of Human-Machine interactions (from ResearchGate)

The need to start or to increase the level of automation in assembly lines and stations, where the processes were mainly accomplished by human operators, was firstly aimed at obtaining an increase in terms of repeatability, precision, quality level, together with a strong reduction in operating time and human working stress, resulting so in an increase in manufacturing units productivity. Indeed industrial applications of human-robot collaborations were initially introduced in the automotive industry, where relevant enabling factors exist that justified and still justify the adoption of this paradigm, and are nowadays adopted also in other industrial sectors, such as the aeronautics, the white goods, the food, the medical and the service robotics.

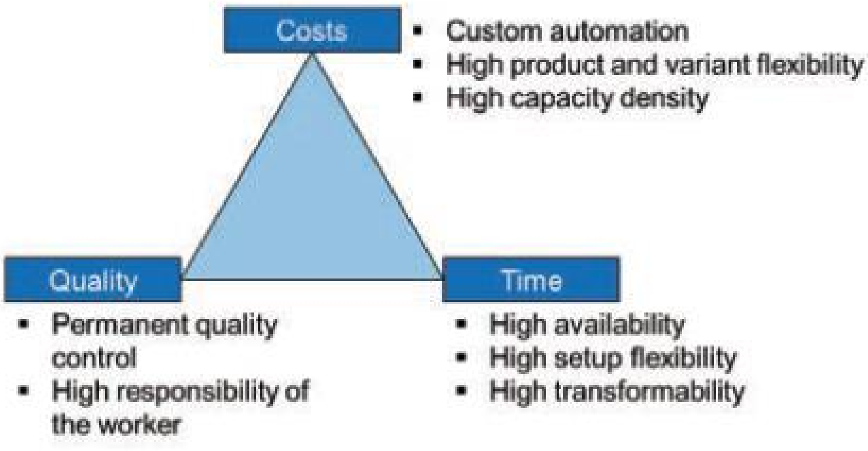

Fig. 1.2 - Potential for HRC (from [9])

By introducing HRC solutions in the assembly lines, it is possible to:

- Reduce the ergonomic stress using co-bots for:
  - Performing heavy operations
  - Carrying out high frequency repetitive tasks
  - Substituting operators in uneasy positions
- Improve quality production by:
  - Robot's repeatability
  - Directly using robot equipped with measurement tools or for specific assembly operations
  - Reducing possible human errors

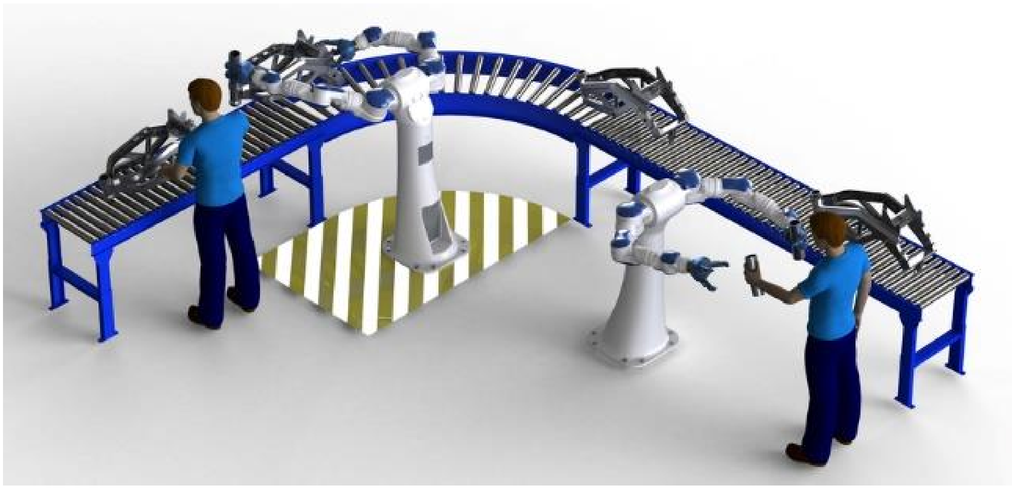

Fig. 1.3 - Assembly line example (from fierceelectronics.com)

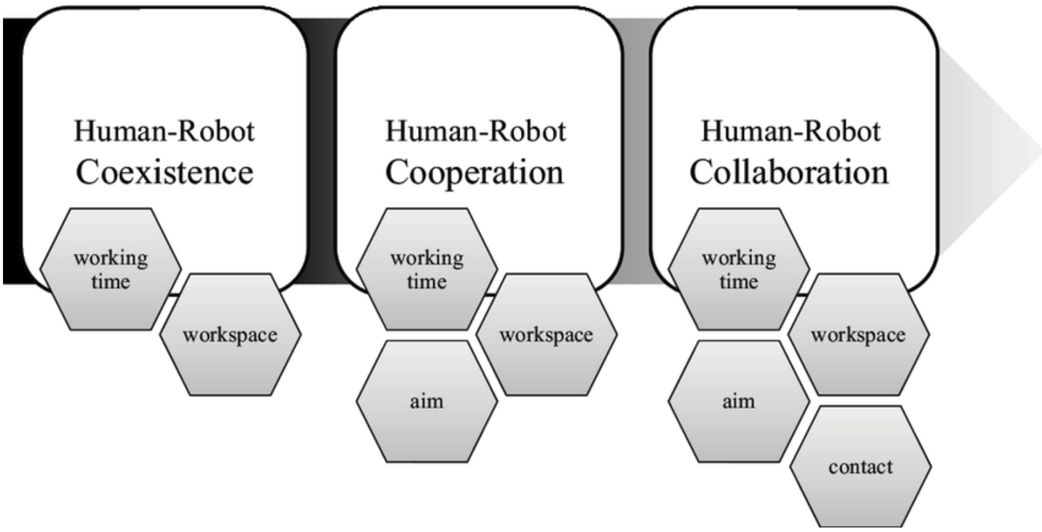

Fig. 1.4 - Classification of HMI aspects (from ResearchGate)

It's evident that the final desirable level is the execution, in direct physical interaction, of the same assembly task by the robotic system and the human operator. To fulfill this cooperation concept, control algorithms and special interfaces had to be implemented so to control the part's movement and operations done by both the robot and the human being.

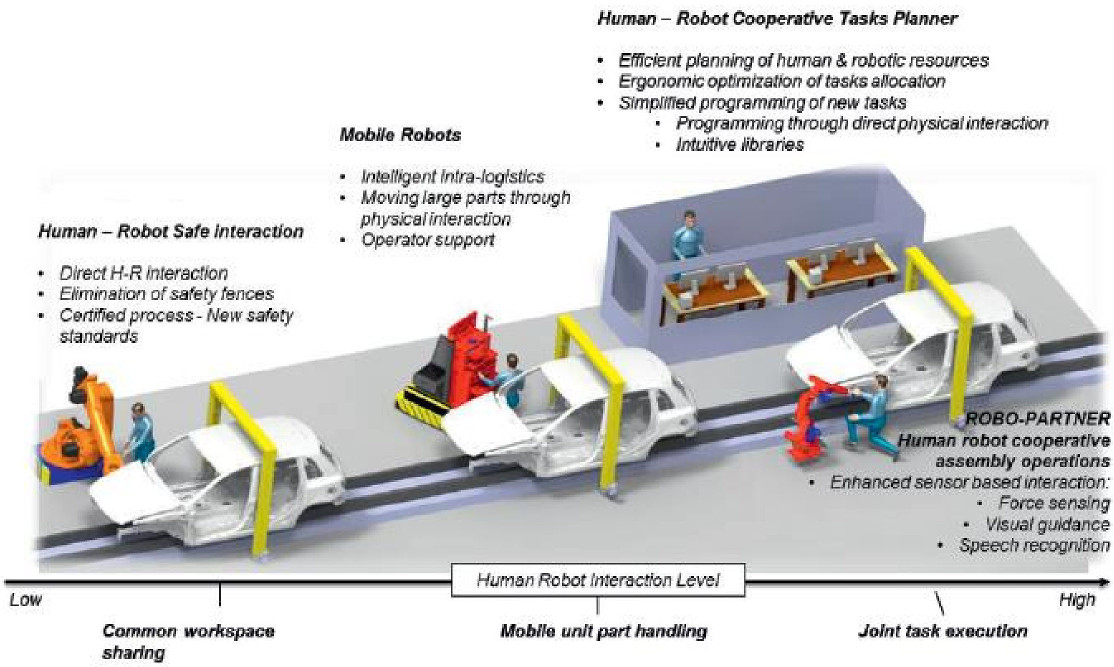

Fig. 1.5 - HRC assembly line example (from [5])

First of all, to obtain collaboration, people's safety in contact whit robots inside a working cell has to be ensured. Modern approaches involve the automatic adaption of the robot speed in case of detection of humans in proximity, as well as the possibility to re-adjust the trajectory in real time to avoid collisions, using different levels of proximity sensors (for example vision systems) and contact sensors (pressure/force detectors). Robots operate in an impedance mode and joints are controlled showing compliant behavior, combining motion sensors and Human Interaction during operations.

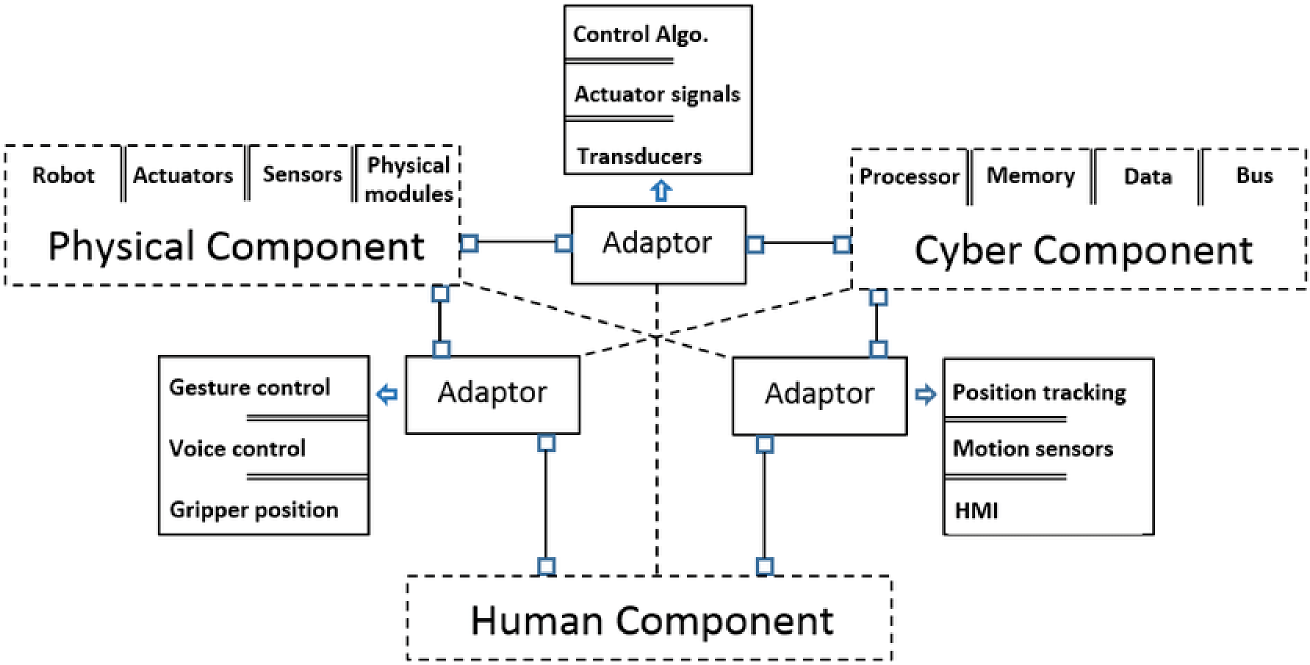

Fig. 1.6 - Diagram of a Human-Robot Collaboration framework (from ResearchGate)

The HRC concept has gained momentum in the last years through the market availability of a new generation of robots, qualified for the direct human robot interaction as previously described. There are different producers and competitive models on the market, such as Universal Robot with the e-Series products (different payloads available and integrations devoted for all the collaborative methods according to ISO 15066), Franka Emika with the Panda Powertool robot arm (equipped with torque sensors in all axes), Bosch with the APAS family (equipped with capacitive skin) as well as the LBR iiwa from KUKA (available with 2 different payloads and high sensitive force-torque sensors implemented in all axes), which is the Co-bot used in this thesis specific application.

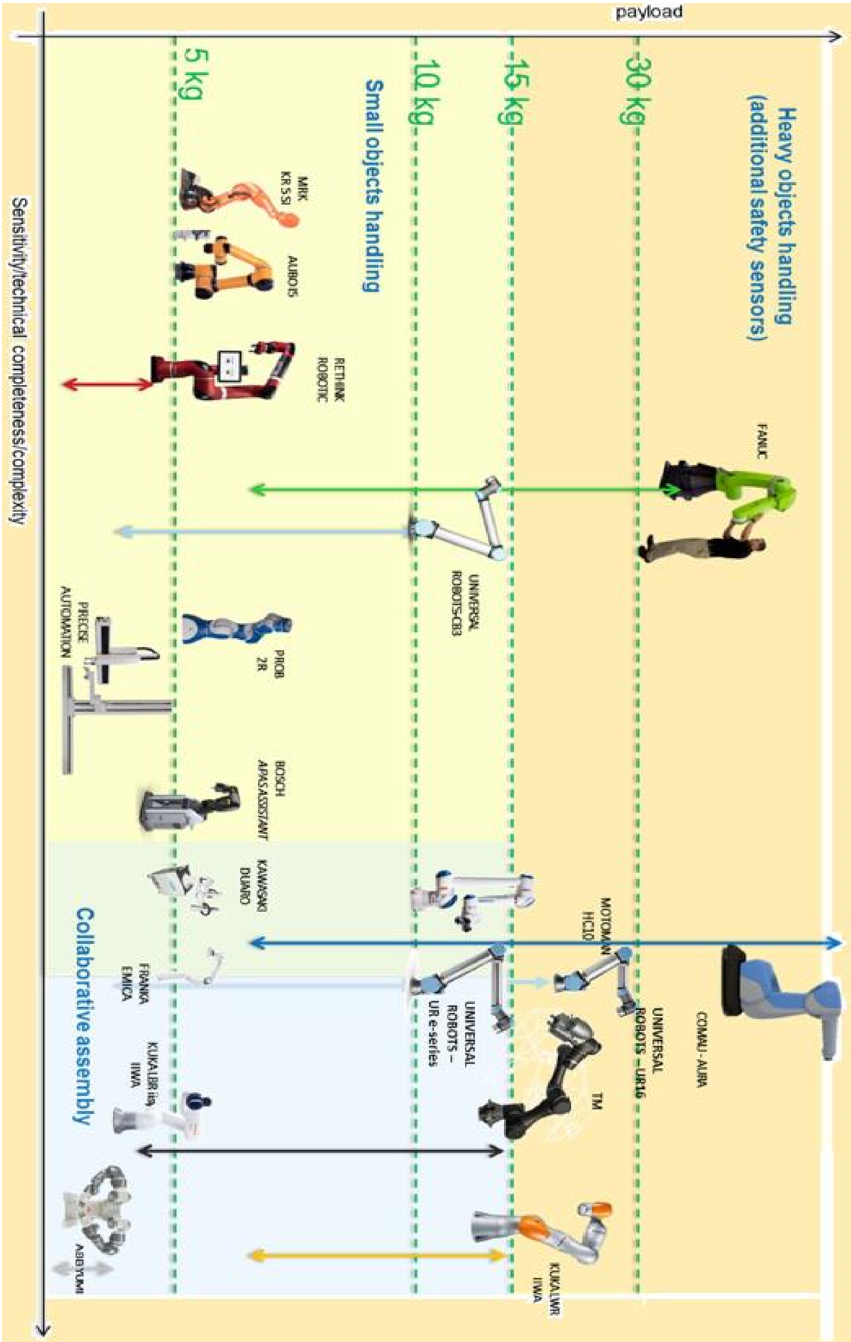

Fig. 1.7 - Cobots market scenario (2019) (from [16])

The current availability of new technologies and regulatory standards allow the implementation of the Human-Machine Collaboration in assembly lines. The design and use of HRC has to be motivated by means of proper analysis.

In the following pages different models for analyzing the feasibility and the type of interactions for a potential HRC workstations will be presented.

1.2 TASK PLANNING

A relevant topic, analyzed in the assembly line for the introduction of HRC, regards the implementation of methods for obtaining an efficient planning of the operations performed inside the working cell so to exploit at the maximum possible level the capabilities of both the operators and robots.

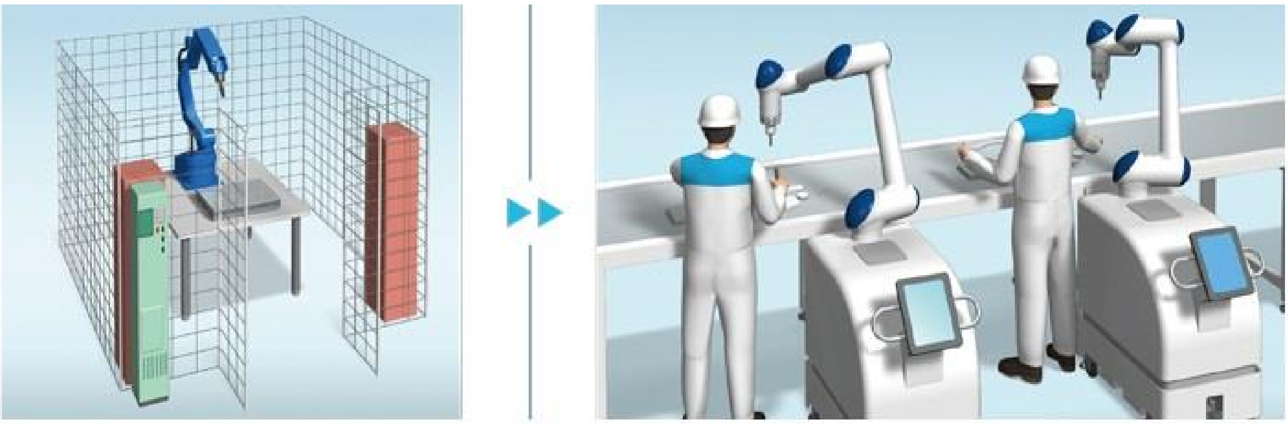

Fig. 1.8 - Collaborative operations example (from Yaskawa global)

There are different aspects have to be taken into account while organizing a collaborative workstation such as:

- Efficient analysis of the product structure and assembly aspects to attain assembly tasks and requirements.
- Planning of the assembly operations and tasks assignment to the most suitable entity (human/robot).
- Use of HRC simulations, in a semi/fully automated way, to evaluate the ergonomics and feasibility of the assignments.
- Use of proven decision making methods to evaluate the task assignments respect to user criteria, ensuring that the execution of the process is efficient and that each entity is exploited in the best way.
- Final exploitation of the results from the planning and simulations, integrating latest technologies to support the operators.

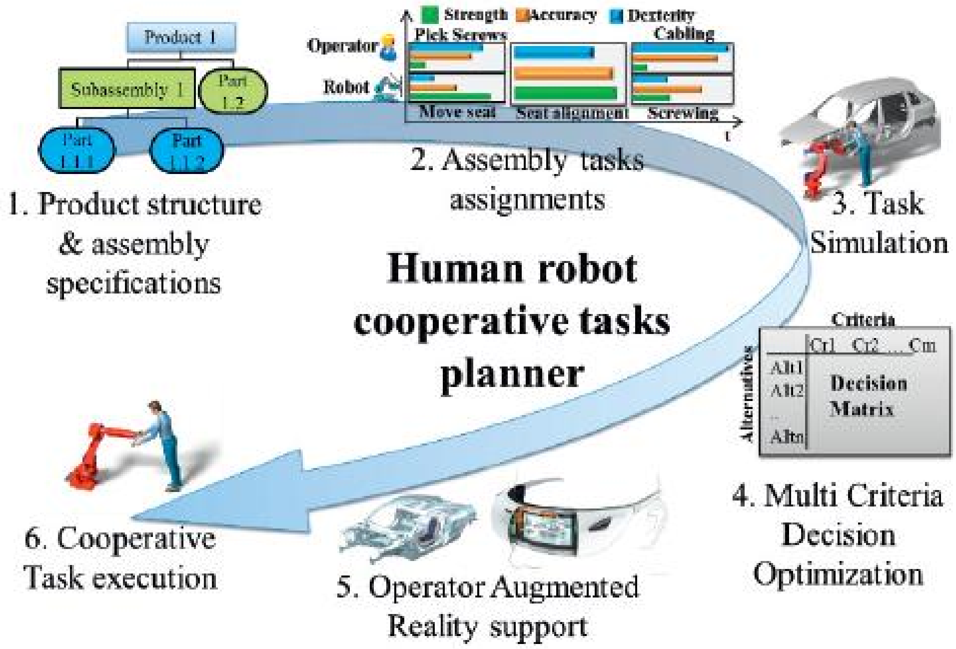

Fig. 1.9 - Task planning (from [5])

Starting from the task planning procedure described before, a diversification of the hybrid assembly systems can be presented by two typologies:

- Workplace sharing systems
- Workplace and time sharing systems

In the first one robot and operator both work in the same workplace, performing handling and assembly tasks in two different configurations:

- The operator is performing a handling task while the robot an assembly task
- Or the operator is performing an assembly task while the robot a handling one

In this type of system, the interaction between the robot and the operator is limited to avoiding collisions, so the robot will stop in case the distance between them is below a given security limit.

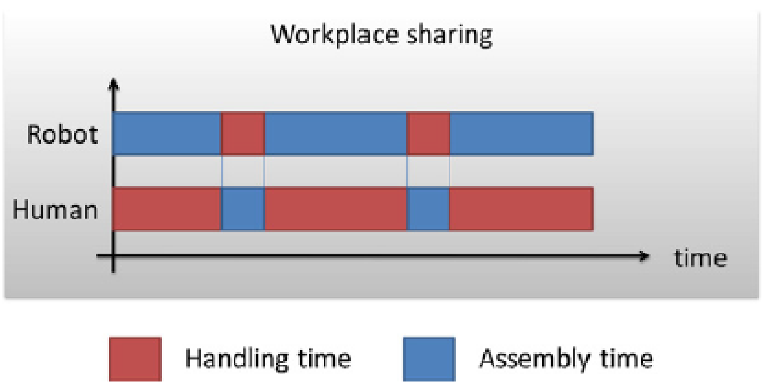

Fig. 1.10 - Example of division task in workplace sharing (from [8])

In the second case, workplace and time sharing systems, operator and robot are additionally able to perform handling or assembly tasks at the same time, in four different configurations:

- The robot is performing an assembly task while the operator a handling one
- The operator is executing an assembly task and the robot a handling one
- The robot and the operator are together performing an assembly task
- The operator and the robot are jointly executing a handling task

In order to perform these type of collaborations, the robot has to interact with the operator on a level that requires much more than just the avoidance of collisions. So the robot has to satisfy the requirements of a safe collaboration of this kind.

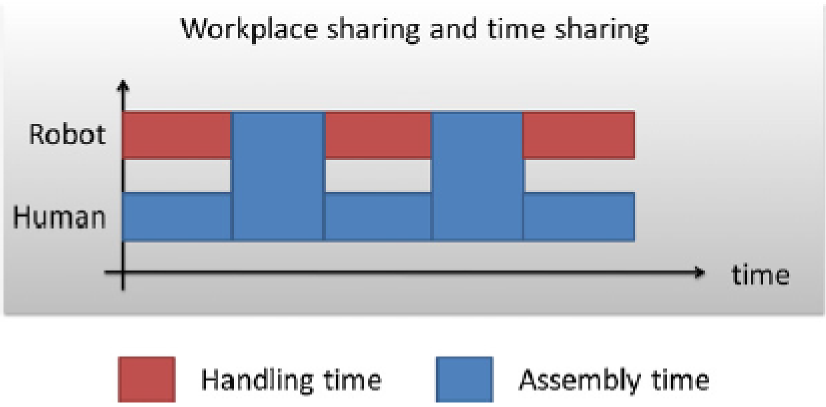

Fig. 1.11 - Example of division task in workplace sharing (from [8])

1.3 WORKPLACES COLLABORATIVE POTENTIAL

As introduced before in the task planning procedure, in manufacturing assembly lines not all the processes are equally suitable for the collaboration between human and robot. In order to implement the benefits from the interaction, strategies for designating the assistance system to the right workplace have been developed, an introduction to a model of this kind is presented below.

The following process can be used to evaluate the potential of a collaborative workplace, based on the MTM method (Methods Time Measurement) which is a tool commonly used in the automotive industry.

- The Methods Time Measurement is a tool used to predetermine the time needed for an operator to perform a task in an industrial environment. In this way, simple core tasks are used to synthesize complex works, defined with their fixed and measured time needed for the execution. A variant of this method is the MTM-UAS (Methods Time Measurement Universal Analysis System) which groups the original core tasks into a set of task-groups related to a specific field of production. Starting from the duration and the intensity of a single task, a time value is designated, as a multiple of the TMU (time Measurement Unit = 0.036s) which is the smallest time unit identified in MTM. In this way it is possible to combine movements of different groups and to evaluate the time needed for the operator to perform a specific task in the assembly line, in a standardized way.

In order to evaluate the automation potential by using HRC in a workplace, this approach is composed of 3 steps. First of all is estimated an automation potential for each single movement in the assembly line. Based on it and the frequency of occurrence, a score is evaluated, then all the outcomes are summarized to finally obtain the results for all the workplaces within the assembly line.

1.3.1 ESTIMATION OF AUTOMATION POTENTIAL FOR SINGLE MOVEMENTS

The database for this type of evaluation should originate from the logistic and the work description of the tasks performed in an assembly line of automotive production industries. The tasks conducted by an operator at a specific workplace, in one cycle, have to be described into single movements.

These movements are described by MTM-codes related to different categories. An automation potential score is assigned to each code having so an estimation of how much a single movement can be assisted or directly performed by a collaborative robot.

The following image presents an example of automation potential scores for different types of operations performed in a generic assembly line and the relative detailed descriptions:

Category (Color)	Movement Description	Potential [%]
Pick	Up to 1kg / Easy-to-pick form	100
	Up to 1kg / Hard-to-pick form	25
	Up to 1kg / Hand-full	0
	Up to 8kg / Average form	75
	Up to 22kg / Average form	50
Place	Approximate	100
	Loose	75
	Tight	25
Sequence	Special Movement	25
	Adjust / Align	0
	Replace	0
	Attach / Release	100
Move	Walk	25
	Bend / Raise	50
	Sit / Stand up	75
Handle	Approximate	75
	Loose	75
	Tight	25
Trigger	Easy	0
	Combined	0
Check	Visual	50
Process	Wait	50

Fig. 1.12 - Automation potential scores (from [6])

- PICK: The picking task is probably the most important movement for evaluating the suitability of a robotic assistance to a workplace. A process can consists of different parts or tools that a robot has to pick, so a detailed description should be considered, taking into account the form (considering that the picking is performed with a universal gripper) and the weight of the part (considering that the payload of the robot cannot be exceeded) to be picked. For example, the potential for picking a lightweight and easy shape object is assigned to be at 100%, while the potential for picking a hand full of small parts, which is not possible for a robot, is defined with no potential.
- PLACE: after the picking, the object has to be placed somewhere. The potential scores are based on the accuracy of the movement. For an approximate placing a 100% in assigned considering it an easy task for a robot to execute, while placing in a tight form could be difficult (blockage or breaking of the part) defining it with a 25% potential value.
- SEQUENCE: This category arranges all the movements that do not fit into any other one. Special movements refer to tasks performed in one movement but not possible to describe as a combination of basic ones, so the value assigned is 25%. Aligning and replacing, being corrective task referred to the human worker, are considered as zero potential, while attaching and releasing could be easily performed by a collaborative robot.

- MOVE: all the movements of the operator's body are described with changes of position and pose. The potential scores are assigned considering the maximum speed of the robot (according the safety requirements) and the feasibility of the movement.
- HANDLE: this category considers the handling of equipment. As the category "place", the automation potential scores depend on the type and tolerances of the task to be performed.
- TRIGGER: although triggering a button could be very easy for a robot, this type of tasks are related to the human perception and understanding of the process. Therefore a zero potential is assigned.
- CHECK: only one code is used to combine different possible situations. The score is 50% because it's easy to replace a visual inspection with a camera but a general statement is hard to make.
- PROCESS: When an operator has to wait for a process to end, maybe he should not be able to perform a task in parallel. So the potential score depends on the previous task and the value assigned is 50%.

1.3.2 SCORE CALCULATION

Analyzing the single station and operations performed in detail (using the previous description), then assigning potential values for each type of movement, a score can be eventually calculated for the whole assembly line.

The score of a movement can be defined as the product of different values:

$$S_m = p_m * t_m * f_m \quad (1)$$

Where:

- S_m is the score of the movement
- p_m is the potential value
- t_m the duration of the task
- f_m the relative frequency of occurrence per cycle

From this calculation the results are then summarized over all the movements to obtain the final score of the whole process, of the workplace or even more than one, so to compare the HRC potential of different workplaces in an assembly line by using the following equation:

$$S_W = \frac{1}{T * n_w * n_c} * \sum S_m \quad (2)$$

Where:

- S_W is the workplace potential
- T is the cycle time
- n_w is the number of operators working in that workplace
- n_c is the number of cycles the workplace is designed for
- $\sum S_m$ is the sum of the scores of the various movements executed during one cycle of the workplace

In this way is possible to evaluate the score of each workplace in an assembly line of a production car industry, so to consider the eventual integration of a Human-Robot Collaboration for a certain application. The higher the score, the higher the benefits from introducing robot assistant.

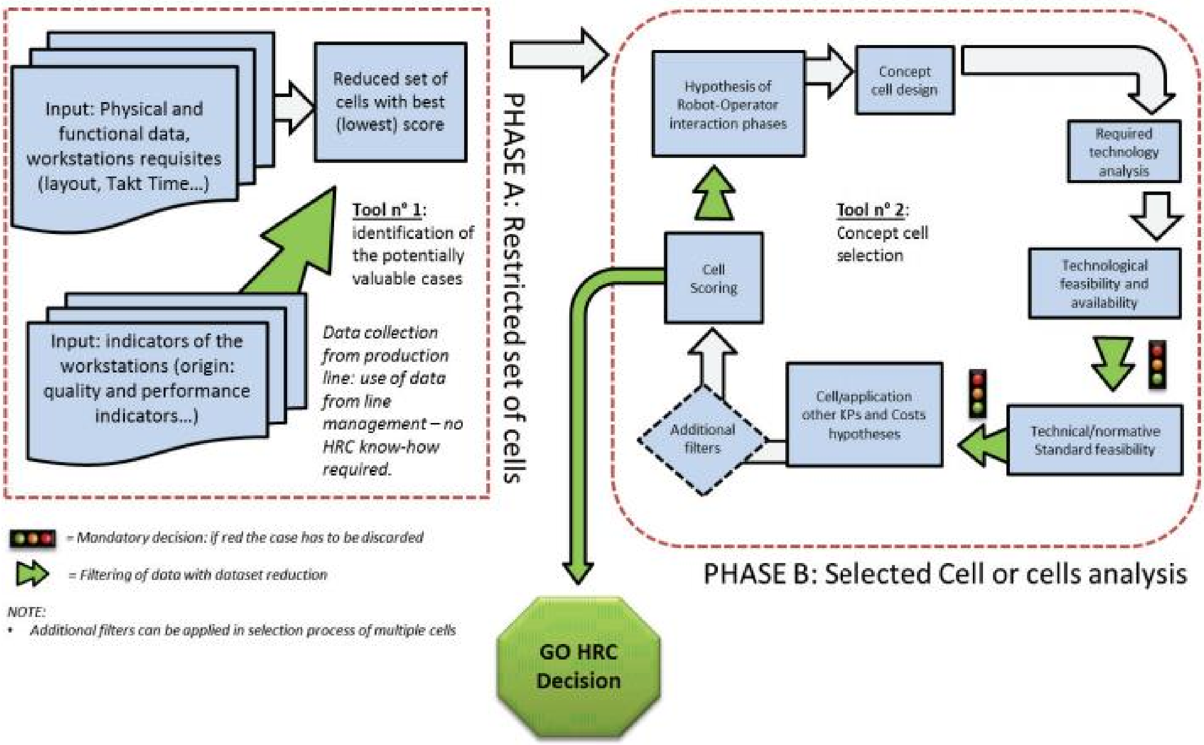

Fig. 1.13 - HRC decision methodology (from [22])

From this evaluation it can also be observed that by implementing assistant robots in assembly lines, not only costs and time, but also work quality and ergonomics can be positively affected. The quality of work of a specific workplace is higher the more value creating, primary tasks are conducted.

Primary tasks, value added activities are:

- Product transformation tasks (prick, place, handle, processing)
- Quality check
- Traceability or similar

Secondary tasks, non-value added activities are:

- Logistics
- Waiting
- Walking in general

So when considering the whole assembly line, to build one car, the total time spent is divided into 60% primary tasks (VAA) and 40% secondary tasks (NVAA). It's evident that the primary tasks are the ones with the higher automation potential, meaning that introducing HRC will increase the quality and production of that specific work. Nonetheless, there are still NVAA tasks where collaboration could be beneficial, for example logistic processes in proximity of the assembly tasks location.

It's also important to specify that dividing the activities in VAA and NVAA tasks is a common practice deriving from the adoption, in industrial manufacture, of the World Class Manufacturing (WCM) production method. It is a collection of concepts focused on operational efficiency by reducing waste, defects and producing cost efficient organization. The main principles on which is based are:

- Implementation of Just in Time (JIT) production
- Introduction of total quality management
- Implementation of total preventive maintenance

2 ISO/TS 15066:2016

Once an introduction to the HRC concept has been made, analyzing methods for task planning and Collaborative workstations potentials, an important focus has to be given to the ISO/TS 15066:2016 which is the normative specifically dedicated to the Human-Robot Collaboration. In the following pages a summary of the concepts introduced are presented.

Human Robot Collaboration is described in the ISO/TS 15066:2016 in terms of risk assessment both for the robot system and the workplace, supporting the industrial robot safety standards explained in the ISO 10218-1 and 10218-2 as it is possible to see in the following image.

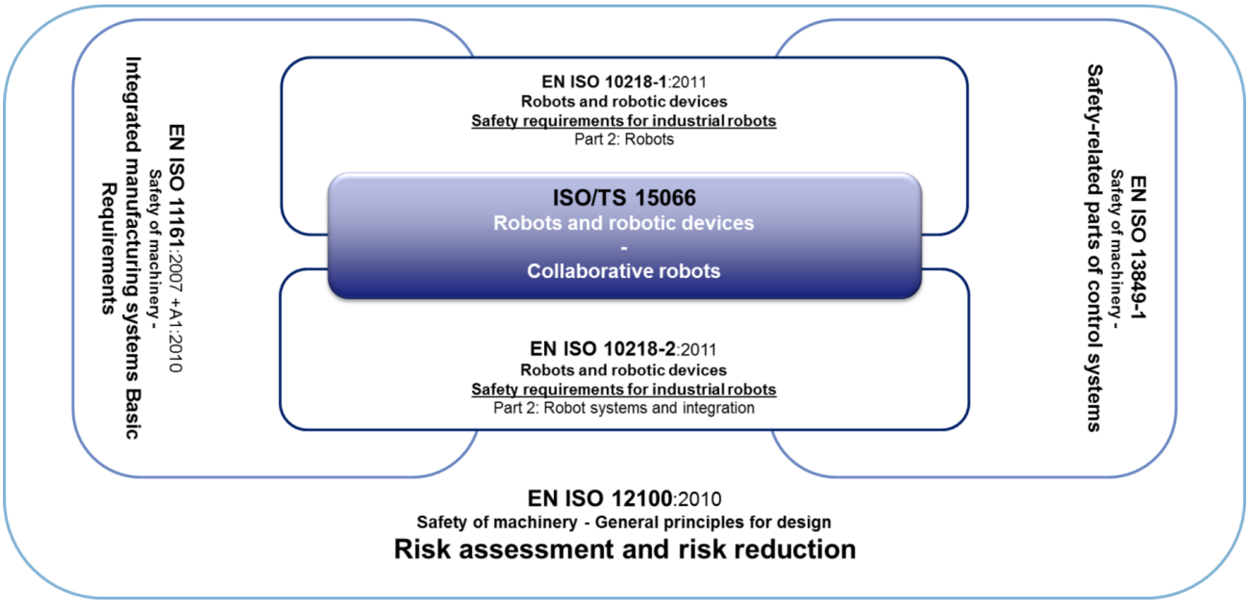

Fig. 2.1 - Main ISO standards related to HRC (from CoLLaboratE D2.1)

This Technical Specification provides some important definitions for the Human-Robot Collaboration such as:

- Collaborative robot: a robot that can be used in a collaborative operation.
- Collaborative operation: an operation in which specifically designed robots work, inside a defined workspace, in direct cooperation with an operator.
- Maximum space: the space in which a robot system (robot + end effector and workpiece) can move.
- Restricted space: the restricted portion of the maximum space limited by using devices that set limits not to exceed.
- Operating space: a part of the restricted space in which all motions are performed during the task program.
- Safeguarded space: a portion of space established by perimeter safeguarding.

An example of the different spaces defined in the ISO is reported in the image below:

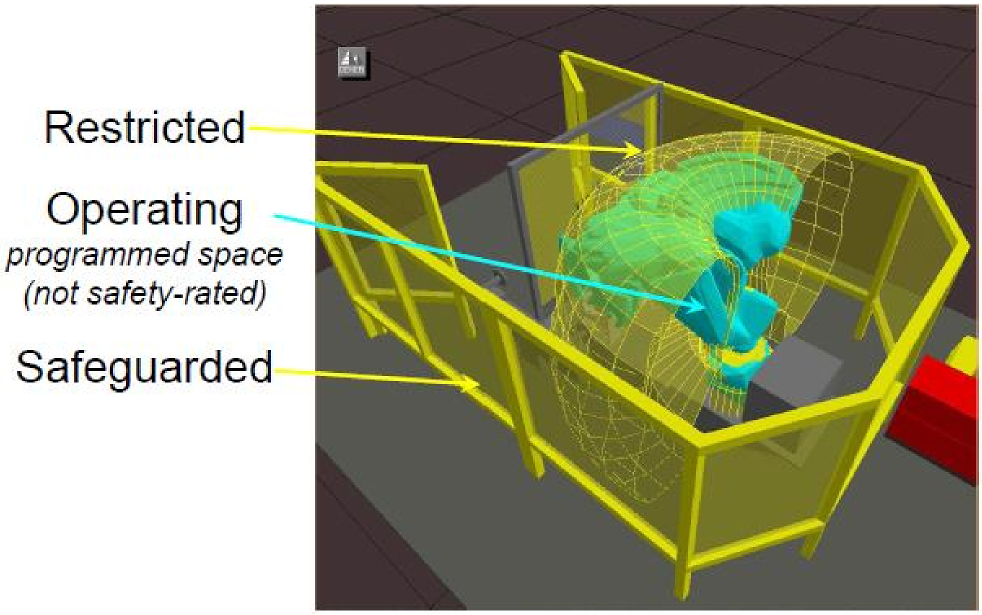

Fig. 2.2 - Spaces specification (from [7])

Starting from the previous definitions, it's important to define the Collaborative Workspace that can be identified as the portion of space inside the operating one where the robot system and an operator can accomplish tasks simultaneously during production operations by means of a specific designing, from the analysis of the tasks and actions to be executed.

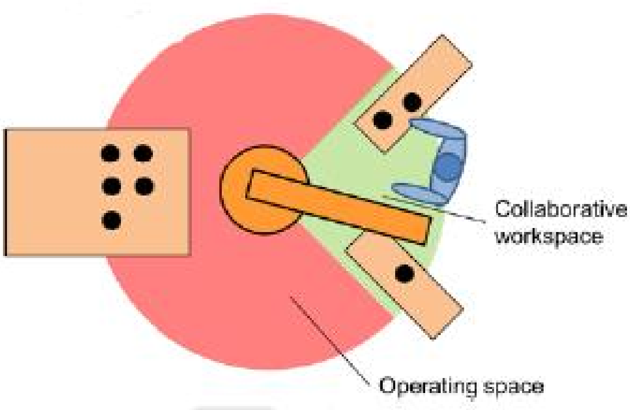

Fig. 2.3 - Collaborative Workspace (from [3])

Indeed, the designing of a collaborative workspace shall permit the operator to perform all the expected tasks, as well as all the risks introduced, due to the presence of machines or equipment, need to be reduced to an acceptable level by means of the precautions adopted through the risk assessment (that will be introduced after).

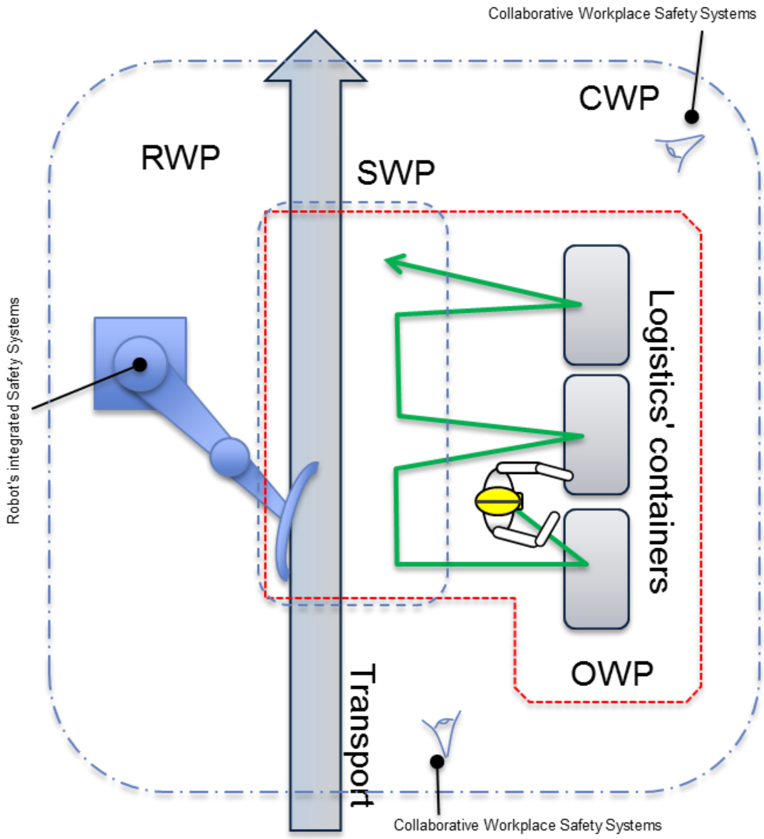

Fig. 2.4 - Detailed collaborative workspace (from [15])

Also, according to the previous image, a more detailed description of a collaborative workspace can be defined by:

- OWP: Operator's Work Place (zone in which the operator moves)
- RWP: Robot Work Place (robot + end effector + component envelope)
- CWP : Collaborative Work Place (according to ISO 15066)
- SWP: Shared Work Place (zone where both the operator and robot work)

Which are the different zones that may be present in a possible assembly line station, so it's important, in developing an HRC layout, to foresee every movement of the operator and the robot in any moment of the application, considering them as dynamic actors, in order to design the cell in such a way to not influence the collaboration and the production of the line.

2.1 RISK ASSESSMENT

As written before, in designing and introducing Human-Robot Collaboration, it's also important to analyze and satisfy all the safety requirements that may derive by the implementation of the equipment and the operations to perform in the station. To do this type of analysis the ISO/TS 15066:2016 introduces new specifications for the evaluation of the specific risk assessment.

In general, a Risk Assessment is a process used to identify risks and to apply reduction measures, performed iteratively in order to assess that the measures chosen are able to achieve the desired effects.

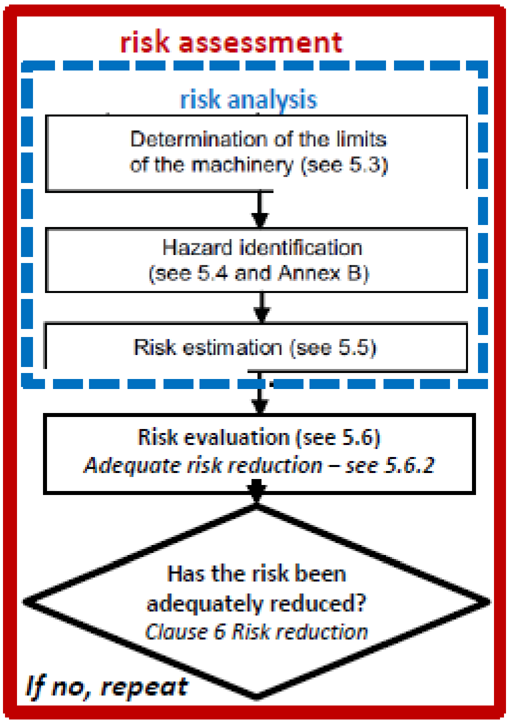

Fig. 2.5 - Example of Risk assessment (from [7])

So, the Collaborative Risk Assessment presented in the ISO adds some strong conditions respect to "standard" robotic applications in the collaborative tasks, in order to analyze, evaluate and document about:

- Intended and reasonably predictable contacts between operators and parts of the robot system.
- Type of contact to be determined for each affected body parts.
- Duration and frequency of the contacts.
- Task with more than one operator involved.
- Transition between collaborative and non-collaborative operations.

In this way it is possible to foresee any hazardous interaction with the robotic system, introducing specific safety implementations and studying the typology of collaboration.

Basically the differences between a traditional and a collaborative risk assessment, according to ISO, are summarized in the following image:

"Traditional" Applications	Collaborative Applications
Inherently Safe Design Measures	
Process design, limiting access, layout	Process modifications, reduced energy, compliant (soft) materials
Safeguards and SRP/CS	
Fixed & interlocked guards	Safety-rated speed, position
Sensitive protective equipment	Safety-rated soft axis and space limits
Hard axis limits or safety-rated soft axis and space limits	Safety-rated torque sensing (impact)
	More...
Safety functions for protective devices and reducing risks	
Information for Use	
SAME or SIMILAR	

Fig. 2.6 - Differences between traditional and collaborative applications (from [7])

It's evident that the risk assessment requires the operation to be specified with all of its components and constraints, resulting in an assessment for the overall application. In case the risks are too great, measures have to be applied in order to reduce them to an acceptable level.

2.2 OPERATION'S TYPOLOGIES

Furthermore, the most important implementation of the ISO/TS 15066:2016 is the classification of the Collaborative typologies.

According to ISO there are 4 typologies of operation for a collaborative application as it is possible to see in the following image.

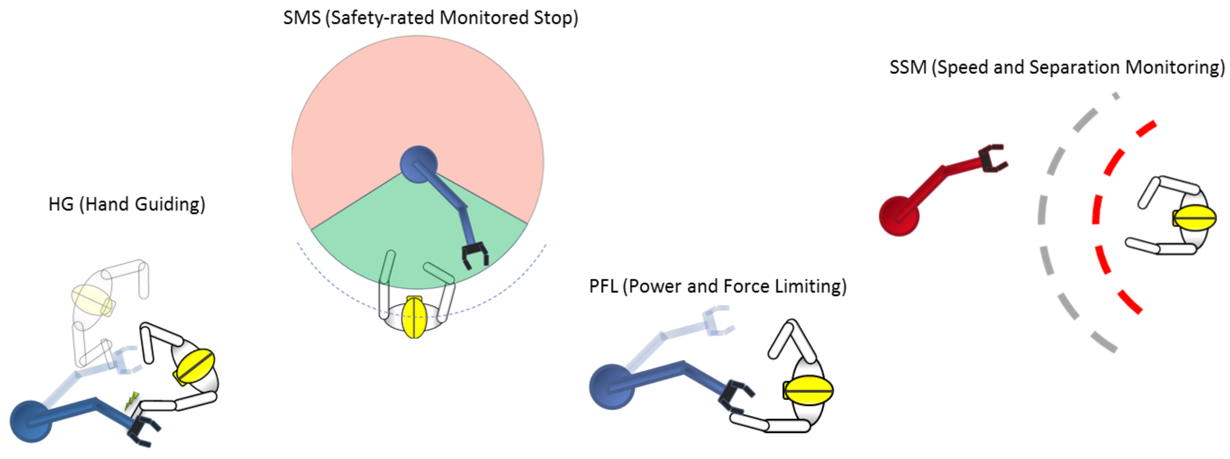

Fig. 2.7 - Collaborative operating methods (from [15])

Any HRC application is defined by one or more collaborative modes, having that the change point between collaborative and autonomous operations should be designed in order to avoid any possible damage from the robot to the operator.

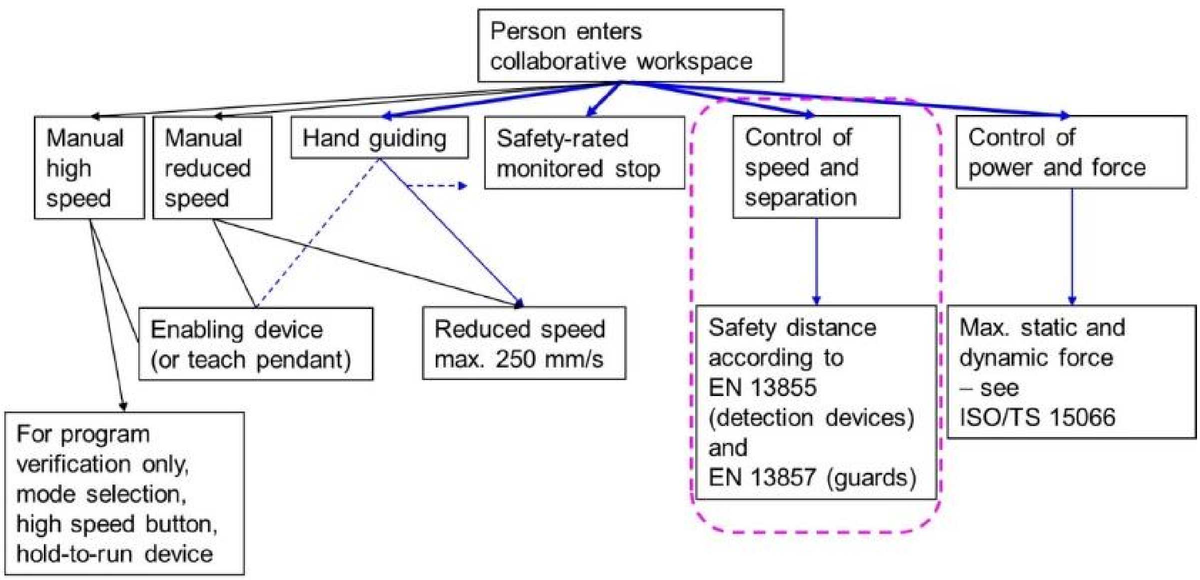

Fig.2.8 - Collaboration typologies diagram (from degruyter.com)

A general description of the different typologies is presented below, defining the specifics and the applications for each of them.

2.2.1 SAFETY-RATED MONITORED STOP (SMS)

This first type of collaboration permits direct interaction between the operator and the robotic system under certain circumstances:

- Safety-rated stop condition before operator gets inside the cell
- Robot drive power rests on
- Robot motion resumes once the operator has left the workspace
- Protective stop issued in case of stop condition violated

Indeed this interaction is suitable for applications such as: direct loading/unloading of part to end effector, inspections during operation, in combination with other techniques.

Robot <system> motion or stop function		Operator's proximity to collaborative workspace	
		Outside	Inside
Robot's <system> proximity to collaborative workspace	Outside	Continue	Continue
	Inside and moving	Continue	Protective stop
	Inside, at Safety-Rated Monitored Stop	Continue	Continue

Fig. 2.9 - Robot system requirements (from [3])

2.2.2 HAND-GUIDING OPERATION

In this specific modality, the operator uses a specific hand-operated device to directly move the robot, the requirements are:

- Safety-rated stop condition before operator gets inside the cell
- Drive power rests on
- Operator holds hand-guiding device and stars operation/motion
- Once the operator has left the cell, the non-collaborative operation resumes

So, some of the applications where hand-guiding operations are used are: assist for robotic lift, like a manually “tool”, Small or limited batch production.

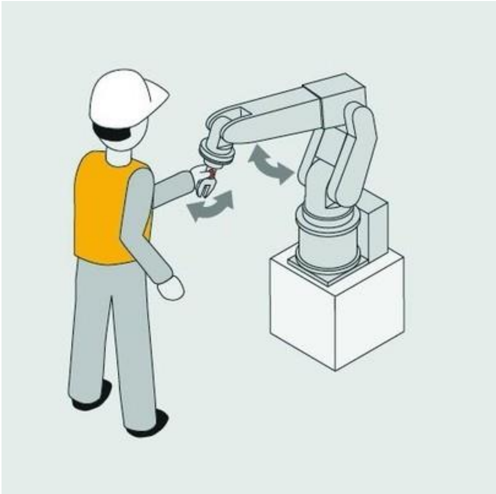

Fig. 2.10 - Hand guiding example (from Rossini-project)

2.2.3 SPEED & SEPARATION MONITORING

This modality allows concurrent motion between the operator and the robotic system inside the collaborative workspace under these circumstances:

- Need of minimum protective distance to separate the operator from the robot, during all the operation
- Protective devices used to regulate approach
- Lowered speed
- Safety-rated protective stop in case of violation of the minimum protective distance

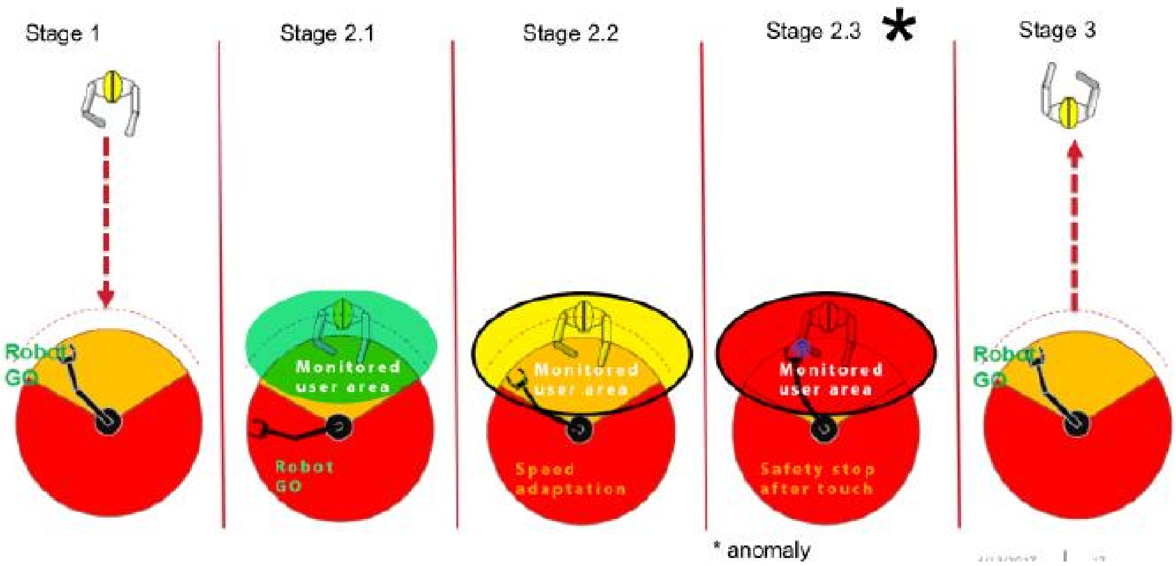

Fig. 2.11 - Stages of SSM interaction (from [17])

Many applications are suited for this type of collaboration such as: Contemporary tasks, Direct interface.

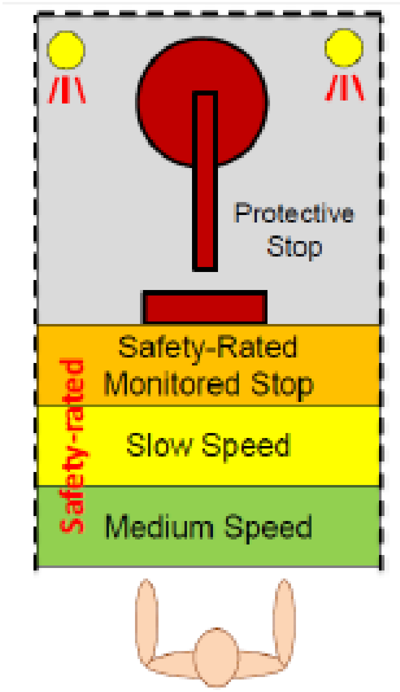

Fig. 2.12 - Protective distance (from [3])

To guarantee operator's safety during operation, a formulation has been implemented to correlate the Protective separation distance with all the possible variables inside the workspace:

$$S_p(t_0) = S_h + S_r + S_s + C + Z_d + Z_r \tag{1}$$

Where:

- $S_p(t_0)$ is the Protective separation distance
- S_h is the operator's change in location
- S_r = the robot's change in location
- S_s = the robot's stopping distance
- C = the intrusion distance that an operator's body part can move in the hazard zone before the activation of the safeguard
- $Z_d + Z_r$ = Position uncertainty of the robot and the operator

Obtaining so the following graphs, representing the trend of the speed and separation distance in function of time in case of a robot system stop during operation, considering its T_r reaction time and T_s stopping time:

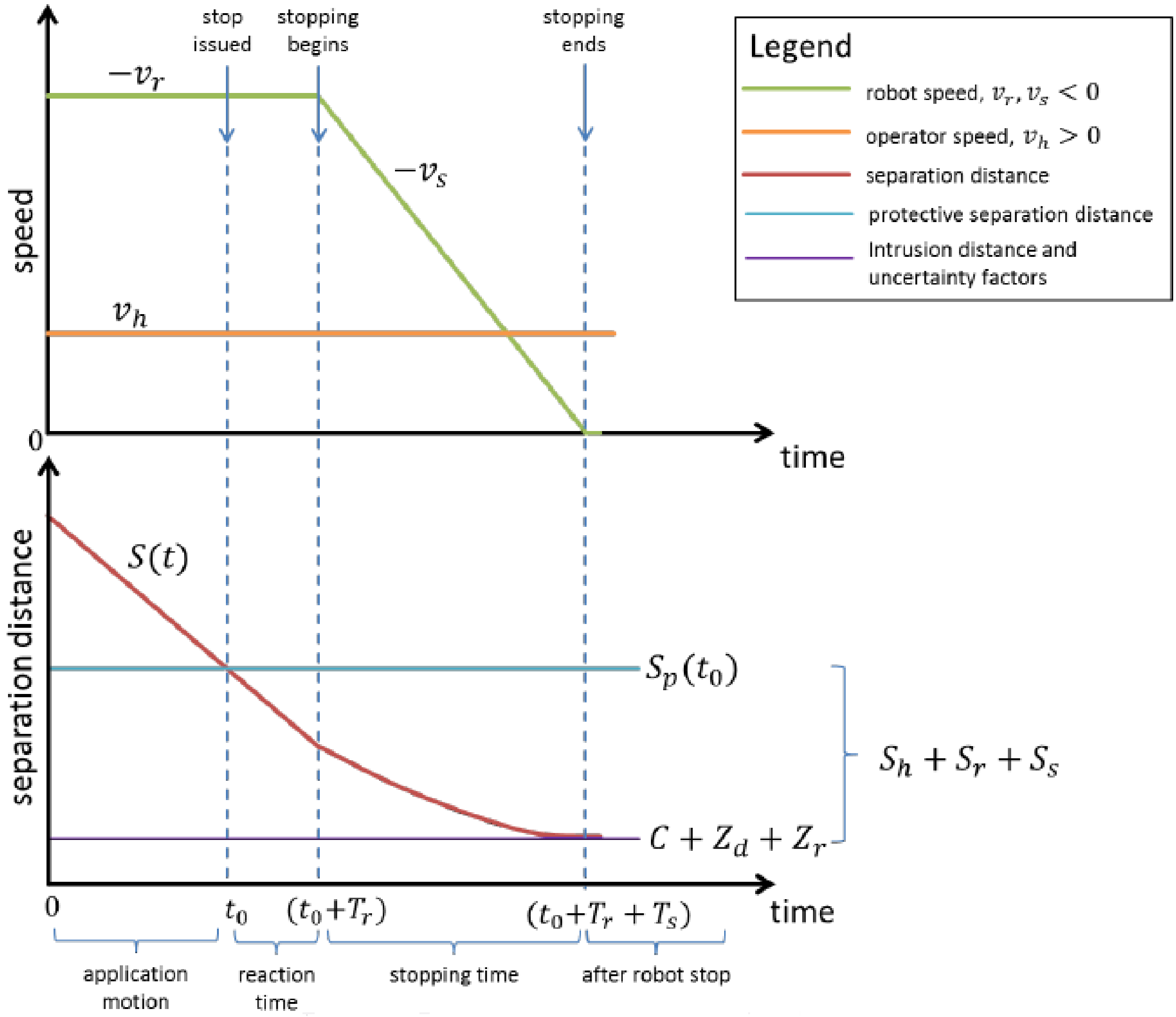

Fig. 2.13 - Trend of speed and distance (from [3])

2.2.4 POWER & FORCE LIMITING

In this last interaction modality, physical contact between the operator and the robot system can occur, intended or not, the requirements are:

- Robot system has to be designed intentionally for power and force limiting.
- Forces that can be applied are needed to be limited.
- Robot system has to react if contact occurs, contact could be transient (dynamic) or quasi-static (pressure).

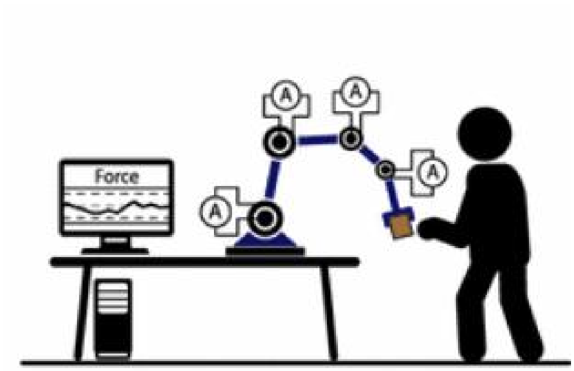

Fig. 2.14 - PFL Collaboration (from link.springer.com)

Applications for this type of collaboration can be variable and with frequent operator presence by means of compliant behavior of the robot manipulator, one of this example is presented as the case study in the following pages.

Regarding the risks of potential contact, there has to be a study in order to reduce as much as possible harm to the operator:

- Identify conditions for which the contact can occur
- Evaluate risk potential
- Design collaborative workspace and robot system in order to have avoidable and infrequent contact
- Account for type, frequency, origin and region of operator's body, force and speed of the contact that can occur, considering that regions like head, throat and neck have to necessarily be avoided

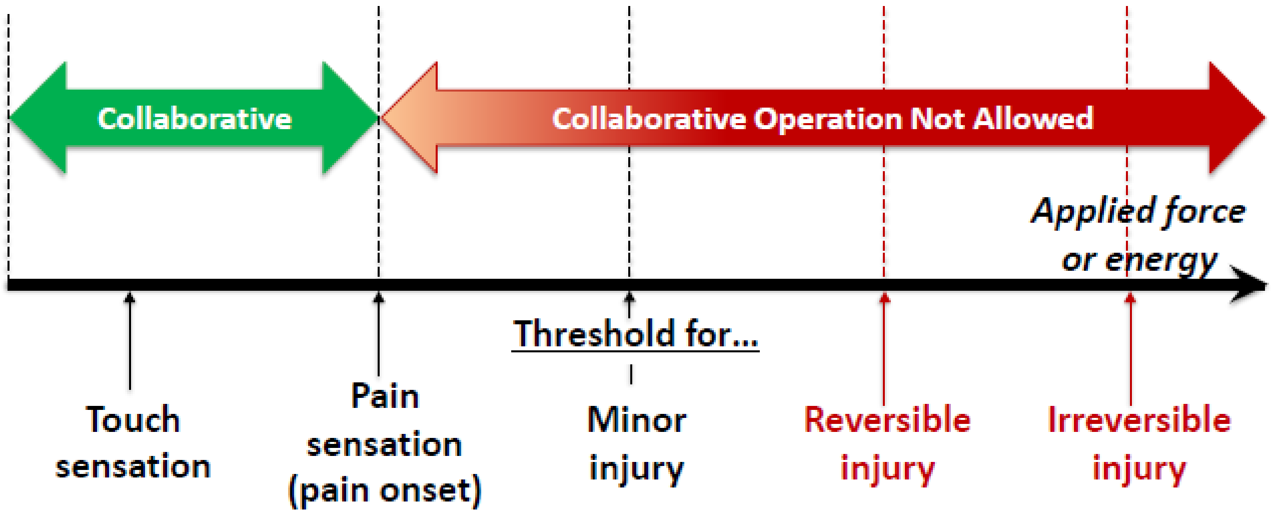

Fig. 2.15 - Threshold injury (from [7])

It’s evident that higher forces can be applied only for a minimum amount of time, avoiding so possibility of a severe injury, thus it can be considered a maximum value for a transient force/pressure applied for an instant, as well as a maximum value for a quasi-static force applied for a longer time in order to have an acceptable region of force/pressure exerted by a contact between operator and robot system as presented in the following graph.

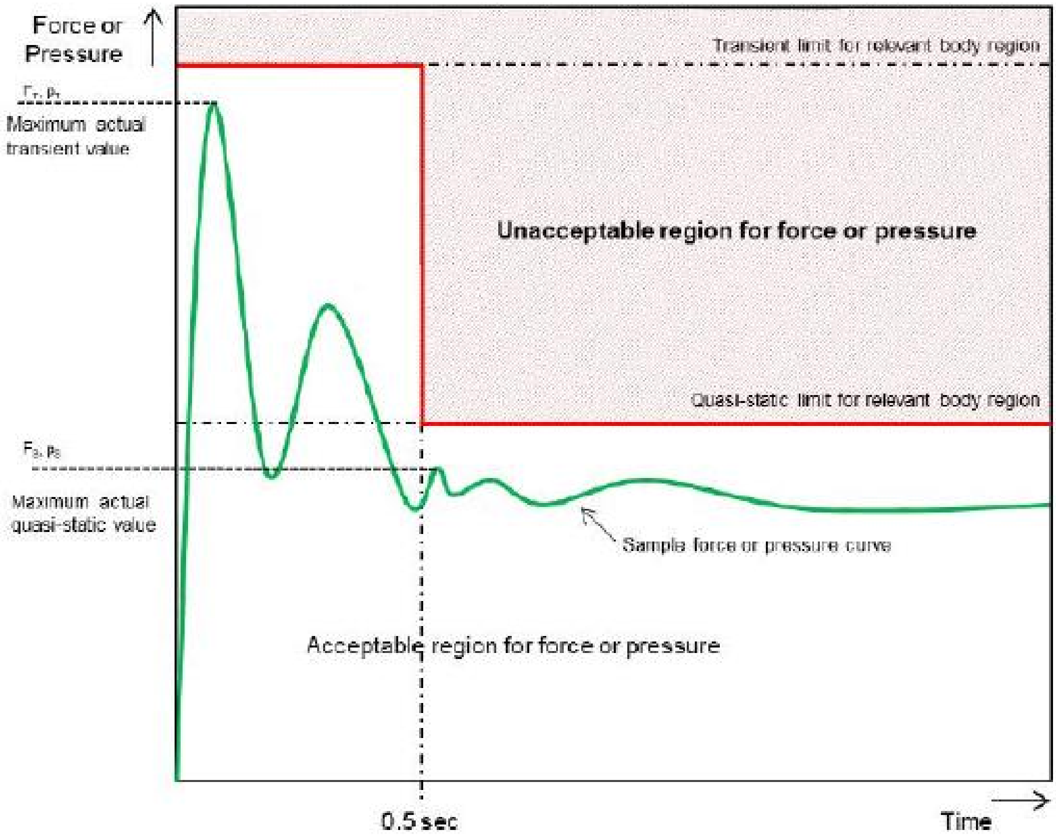

Fig. 2.16 - Regions of acceptable/unacceptable forces in the PFL modality (from [3])

Furthermore, according to a research by Haddidin et. Al. the energy transferred between a robot system and a human being can be affected by changes in speed, having that it can be considered as an inelastic collision.

Starting from this consideration, it’s obtained a graph that relates the effective robot mass converted from peak pressure limit values versus speed limit, for any different body parts.

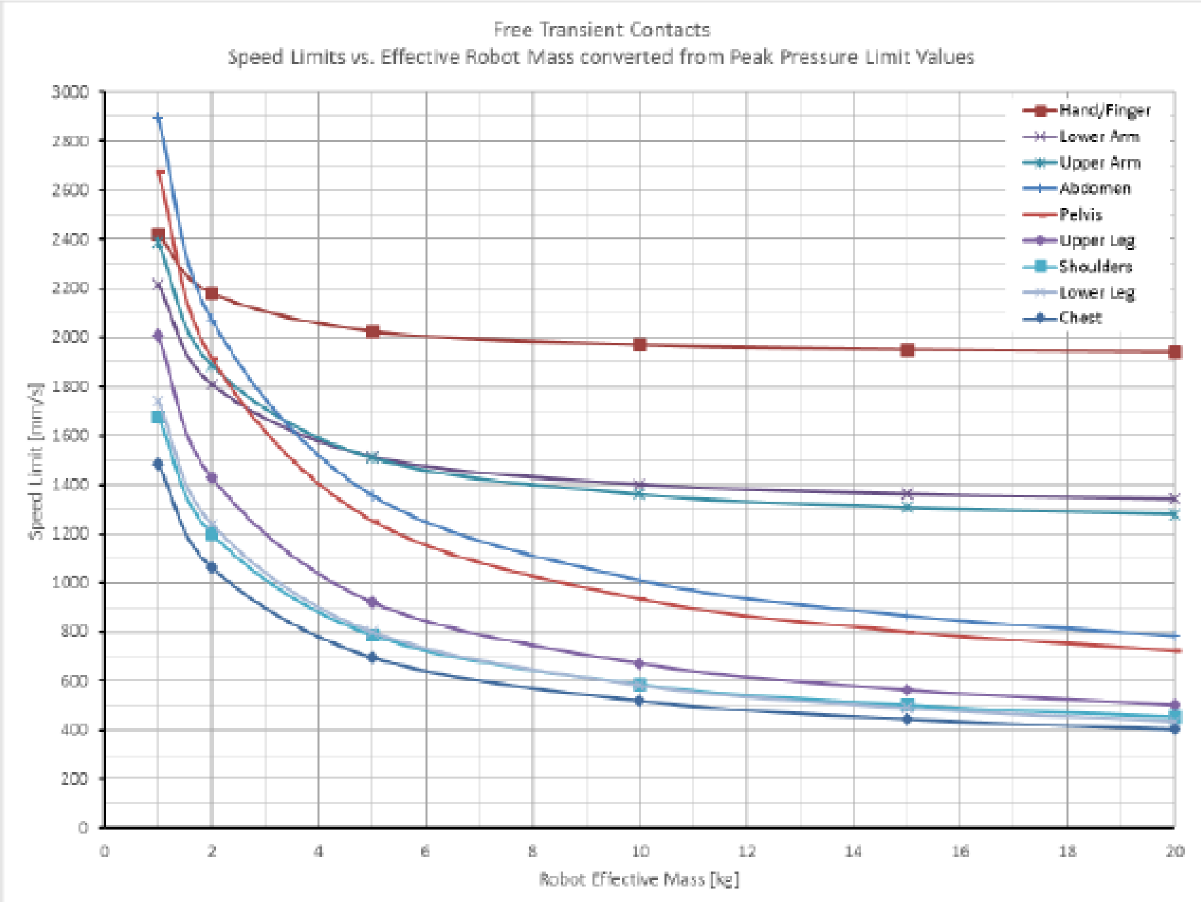

Fig. 2.17 - Mass (Pressure) VS Speed limit (from [3])

It can be observed that the more sensitive the body part, the lower the speed limit for which the same mass can be moved, this because the injury would occur faster and would be more dangerous.

Those limits can be modified or affected by several precautions, for example:

- Eliminating crush and pinch points
- Reducing inertia of the robot system
- Reducing transfer energy by reducing robot speed
- Increasing contact surface area by changing the robot posture
- Avoiding as much as possible sensitive body areas

3 HUMAN MACHINE INTERACTION FOR WORKSTATION DESIGN

A general description of the Human-Robot Collaboration has been presented, introducing benefits, potential workstations and the implementation of the ISO/TS 15066:2016. Since the following case study is based on the developing of a physical Human-Machine Interaction for a specific cycle and work-cell, in the following pages will be presented a model that analyzes the design of a HMI inside a workstation from the point of view of all the operator's behavioral loops and involvement in a manufacturing cell, starting from the Equipment Design and Production System Design.

A manufacturing system can be described by ED (Equipment Design), that consists in the design of hardware components of the process (machines, cell layouts, fixtures), and PSD (Production System Design) that refers to the interactions and connections between manufacturing resources. Modelling a system means modelling human performances and functions and how they interact with robots. Regarding the ED, designers try to satisfy the Functional Requirements FRs, understanding the sources and the process to convert system-level FRs into equipment-level FRs.

A manufacturing cell can have different levels of control modes, such as manual, hybrid and automatic control. Human operators add flexibility in terms of design and dexterity during manufacturing processes, also monitoring the entire cell which can be considered as a semi-automated subsystem. Human-Machine Interactions in manufacturing cells can be modelled with the supervisory control paradigm describing the operator's work routine. Using this model, it is possible to describe the mechanism and the functions by which the operators get process-state feedbacks and react according to these.

The operator's involvement as a supervisor can be described using five functions:

- Plan
- Teach
- Monitor
- Intervene
- Learn

These functions' feedbacks correspond to operator's behavioral role in controlling the system: skill-based, rule-based and knowledge-based, also referred as Behavioral Loops (BLs).

As the following figure shows, these BLs stand for the iterative nature of the functions performed by operators also incorporating process control and continuous improvement activities.

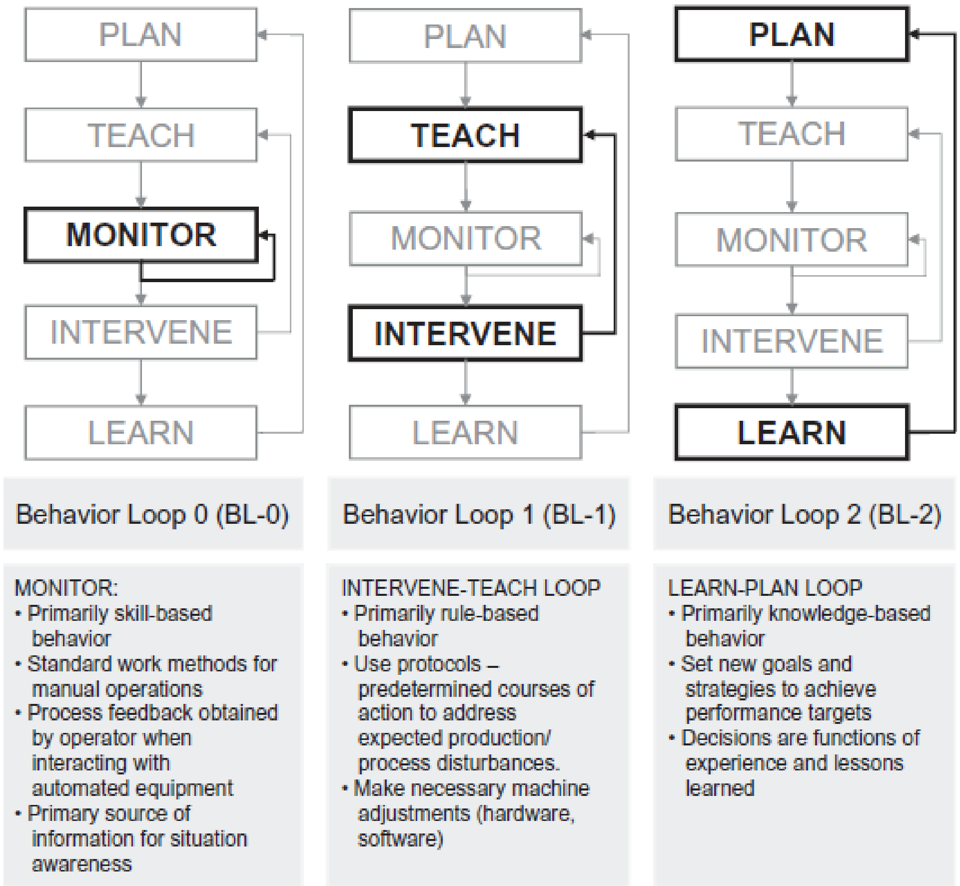

Fig. 3.1 - Sheridan's supervisory control paradigm (from [4])

Monitoring is designated as a skill-based function at BL-0. In order to achieve successful system control, it's needed for the operator to acquire necessary skills through experience and training programs. Standard work routines consists of operator's familiarity and functional capability to handle specific types of equipment.

The Intervene-Teach Loop as BL-1, focuses on rule-based behavior. For instance, when a process disturbance occurs, the operator's controlling action is classified as BL-1 and follows a specific procedure for fixing the problem condition. In such a way, the operators develop written procedures knowing the best corrective actions to perform in order to solve specific problems.

The Learn-Plan Loop, BL-2, exploits the operator's cognitive capability. The operator collects data based on observations through the learning functions. Understood the information, is capable of making decisions during the Plan stage. Indeed, planning is the action by which an operator can perform change in the system goals and strategy resulting from the experience gained during system operation.

3.1 MODEL

This model determines and maps FRs and DPs (Design Parameters) into categories used for the modelling of the Behavioral Loops.

Regarding the skill-based HMI (BL-0), an example of FRs and DPs pairs are shown in the following list, these sets are derived from the Manufacturing System Design Decomposition (MSDD Cochran et al. 2002) and relate to human, information and machine elements that interact when skill-based tasks are performed by the operator.

BL-0 Human Work Content	
FRs	DPs
Eliminate operator waiting on machines	Human-machine separation
Ensure predictable equipment output	Maintenance of equipment reliability
Ensure availability of workers	Perfect attendance program
Operator has knowledge of required tasks	Training program
BL-0 Machine Design	
Eliminate wasted motion of operators	Design of workstations/work-loops to facilitate operator tasks
Minimize wasted motion in operators' work tasks	Ergonomic interface between the worker, machine and fixture
ensure operator human errors don't translate to defects	Mistake proof operations
BL-0 Information	
Reduce variability of tasks completion time	Standard work methods to provide stabilized processing times
Operator consistently performs tasks correctly	Standard work methods
Determine capability of process	Measure current process

It can be observed that workers must receive specific training for the responsibilities related to physical interaction and information transfer while the machine should possess specific characteristics so to help operators in receiving process state information and in physical interactions, as summarized in the following image.

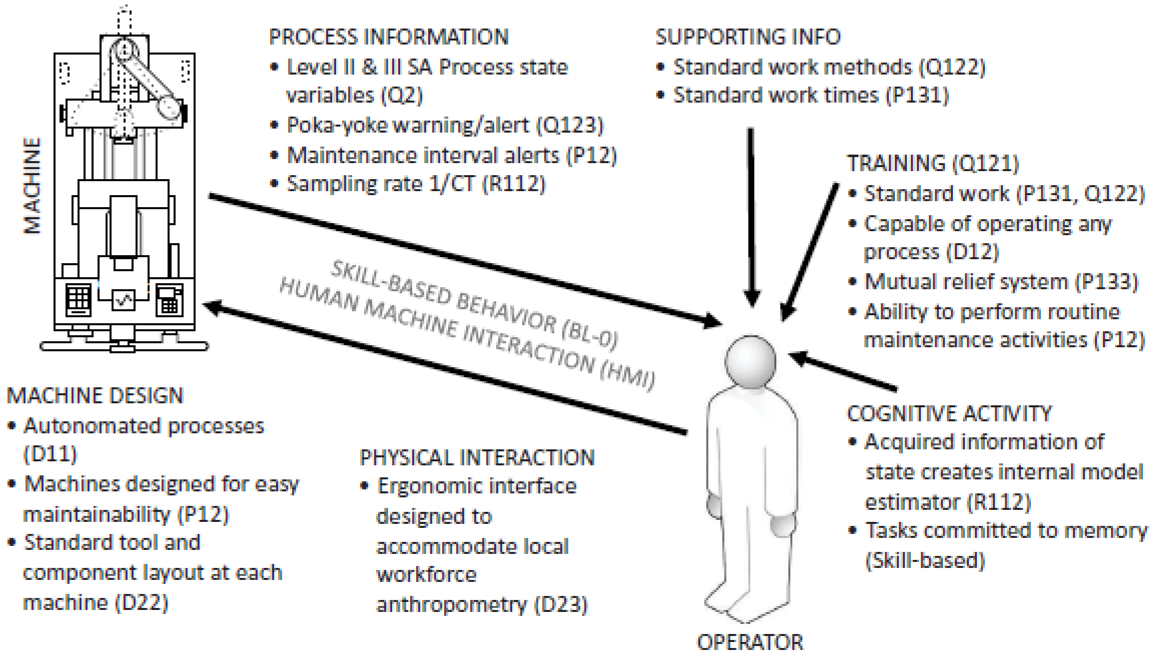

Fig. 3.2 - HMI Model for skill-based behaviour BL-0 (from [4])

Instead, for the rule-based BL-1 HMI, the FRs and DPs are associated with rule-based tasks and affect how disturbances are recognized by the operator and how he acts and resolves the problem, interacting with the affected subsystem. An important feature of the rule-based behavior model is that an important amount of supporting information is required for the operator to perform the intervention and teaching functions.

Analyses and failure mode, containing information on root cause of a process disruption and documentation of cause and effect relations, should be maintained in order to enable reaction and to document the corrective actions. Indeed machines should be designed and equipped with user-friendly interfaces so to complete with minimal effort any adjustment or logic reprogramming. A machine that is easy to maintain, reduces the amount of time in the teaching function.

The following image presents the characteristics of the BL-1 HMI described before:

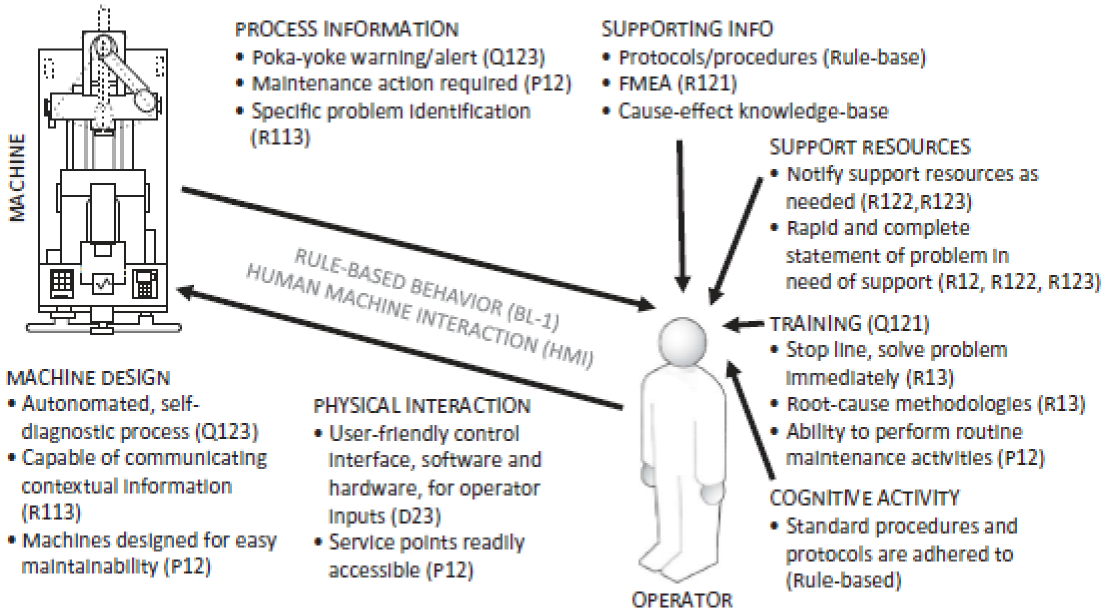

Fig. 3.3 - HMI Model for Rule-based behaviour BL-1 (from [4])

Finally, regarding the Knowledge-based tasks, BL-2, DPs are chosen to address work content, machine design or information-based FRs that influence how operators approach their role in the system. It has been said that BL-1 describes how an operator withdraws from standard operating procedure to compensate for a problem that may have been accounted for or predicted through good system design, while BL-2 explains how an operator solves problem not predicted or anticipated. It facilitates discovery of the reasons why the problem manifested, learns/records and sets new goals, methods and strategies to avoid possible recurrence.

All these specifics are summarized and displayed in the image below:

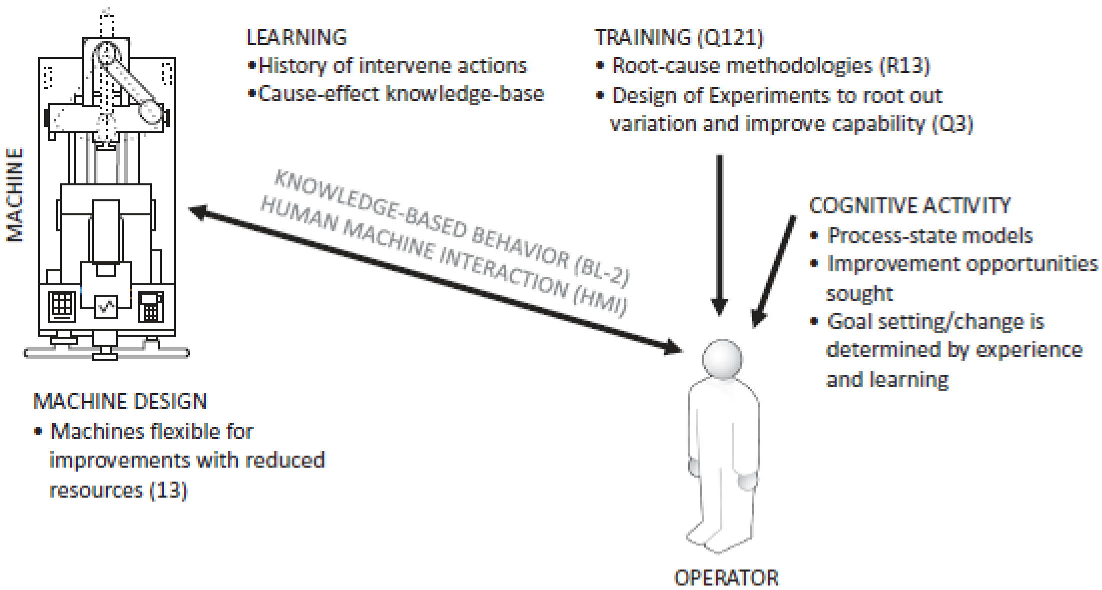

Fig. 3.4 - HMI Model for Knowledge-based behaviour BL-2 (from [4])

Once the correlations are made, the last part of the model consists in the integration of the three behavior models in a complex structure of loops connected with the five supervisory functions of plan, learn, monitor, intervene and teach. This implementation can be expressed as:

Operators performs manual tasks and monitors automated production processes during standard operation of a manufacturing cell. During BL-0, performs value-adding tasks while gaining process-state awareness. If a device or the operator detects a disruption, BL-1 is activated. The operator leaves normal operations to Intervene and start a procedure to deal with the disturbance as a Teach function. In case there is no available procedure to solve the problem (BL-2 paradigm), the operator documents the occurrence to identify the root cause through Learn. The workers then formulate a corrective action as Plan, implementing it by means of adjusting system settings with Teach. At this point the cell/subsystem has once again resumed standard operation and the operator returns performing tasks and monitoring it.

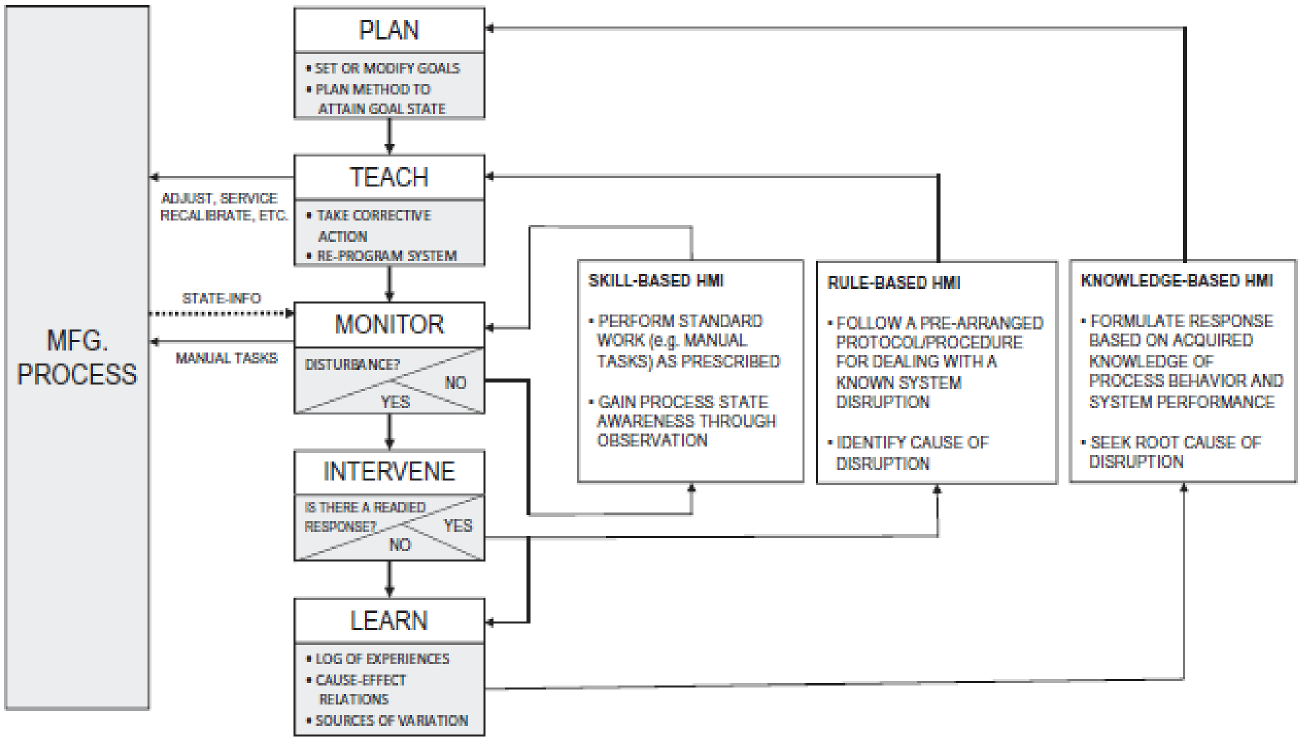

Fig. 3.5 - Nested supervisory control for three behavioural loops (from [4])

So, in the end, FRs are generated during ED, Product Design and Production System Design. The integration of HMI models during Equipment Design should improve the utility of the HMI approach for designers obtaining so successfully interactions and workstation design. A system designer should use these models to generate FRs and related DPs for ED for a specific manufacturing cell, considering so the levels of HMI, if needed, during the progression.

4 CASE STUDY: BALANCE SHAFTS INSERTION

In the previous pages has been presented an introduction and a description to the Human-Machine Collaboration concept, through models and aspects analysis. This section will be dedicated to the case study at the HRC Laboratory of Centro Ricerche Fiat in which a specific physical Human-Machine Interaction has been developed. In the following pages the workstation and the cycle are described, focusing on the single components in detail, then the implementation of the HMI will be fully explained.

This operation consists in the insertion of two balancing shafts inside an engine basement. The cycle is entirely performed by means of a collaborative robot in continuous moving, through specific programming, with the operator assistance.

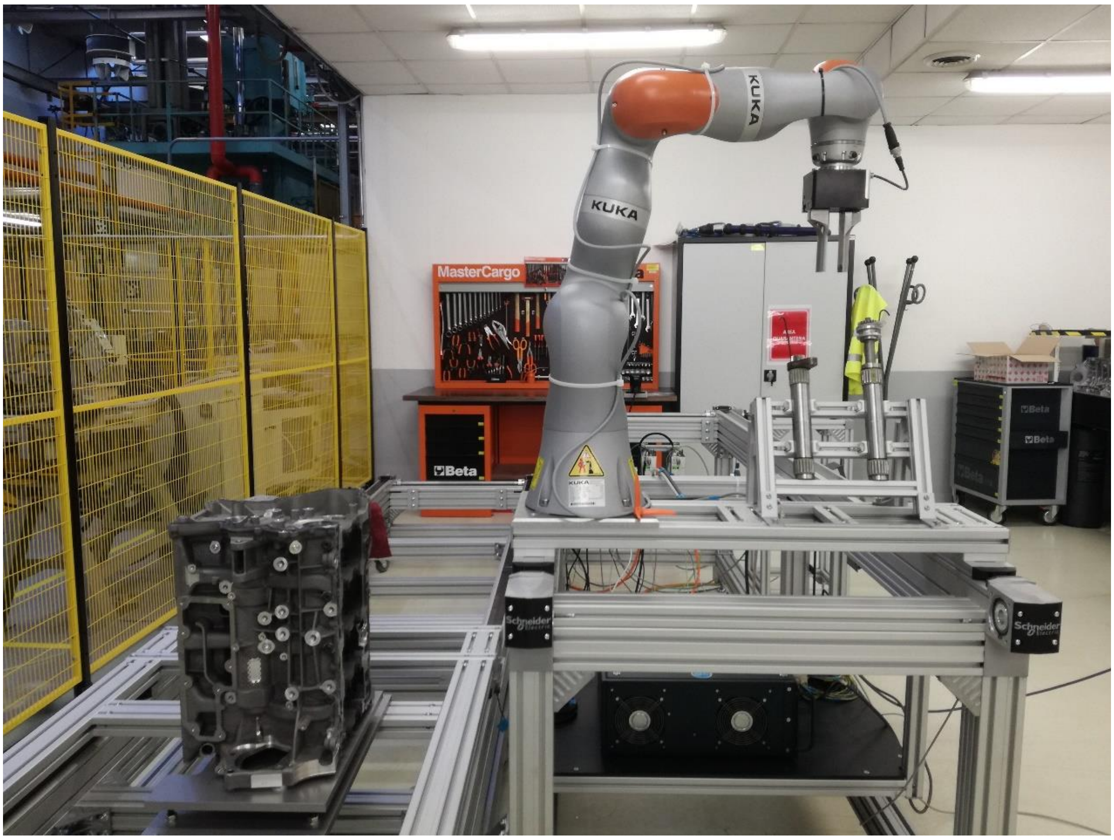

Fig. 4.1 – Workstation (from HRC LAB CRF)

4.1 CELL LAYOUT AND COMPONENTS

Fig. 4.2 - Cell layout (from HRC LAB CRF)

The components of the working cell are:

- Collaborative robot: KUKA LBR iiwa 14
- Zimmer Gripper integrated with specialized claws
- Two different motorized linear axes manufactured by Schneider Electric
- Engine basement
- Two balancing shafts
- Two support structures specifically designed
- Cables and electronic components

A detailed description of the elements is presented below.

4.1.1 COLLABORATIVE ROBOT

The KUKA LBR iiwa 14 (LBR stands for the German version of LWR LightWeightRobot and iiwa for Intelligent Industrial Work Assistant) is the co-bot used in this application. It's a 7-axes robot arm with all rotational joints each one equipped with a torque sensor, resulting in a very sensitive component able to detect contact all over and to work with humans in a safety collaboration through programmable compliance. The payload capacity is 14 kg with a weight of 29.9 kg and the total reachable distance is 820 mm. The housing is entirely made of aluminum, reducing weight and increasing safety. The design is made in such a way to eliminate edges, minimizing crushing and shearing hazards when working together with the operator. The repeatability is ± 0.1 mm and the axis-specific torque accuracy is ± 2%, making it very suitable for complex assembly tasks.

The robot can be programmed using the software KUKA Sunrise Workbench, in the KUKA Sunrise programming environment (Java technology). Indeed, to fully exploit the capability of the robot, it was necessary to read and understand the manual provided by KUKA.

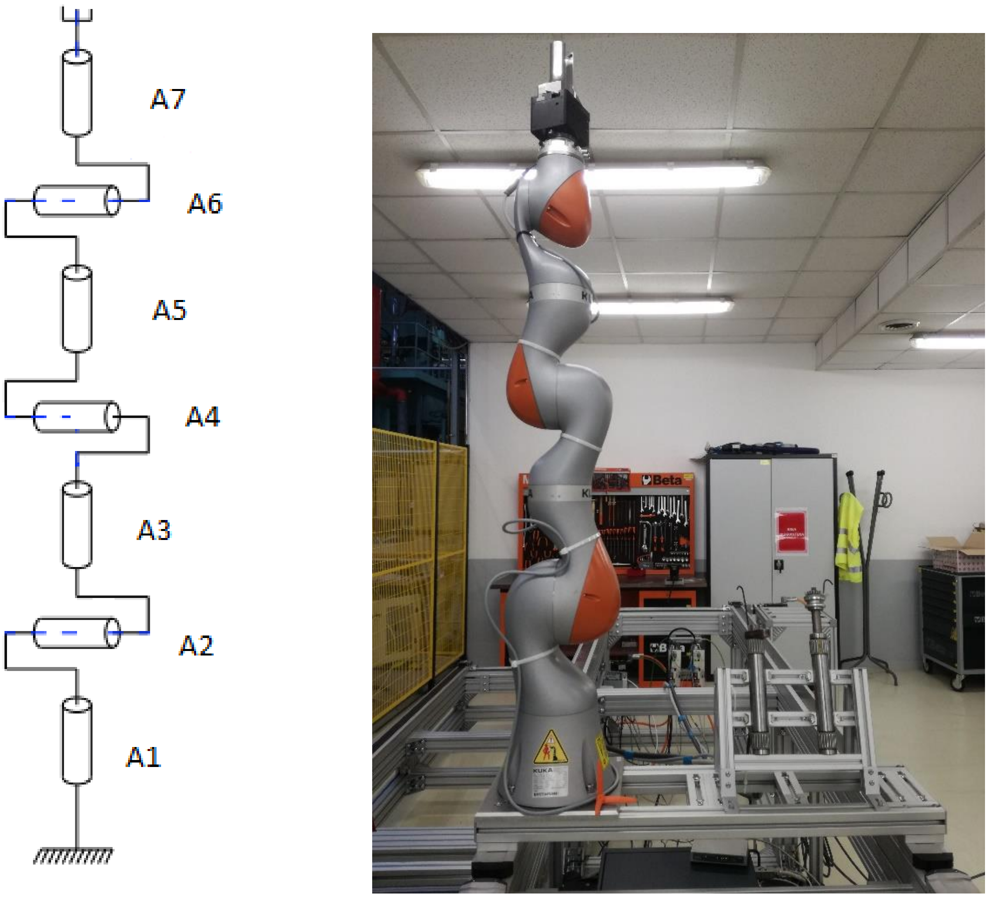

Fig. 4.3 - : Kinematic representation of the KUKA LBR IIWA 14 industrial robot and detail of the work-cell (from HRC LAB CRF)

The human-machine interface of the LBR iiwa 14 consists of a smartPAD control panel that has all the operator control and displays function required for the operation. It has a touch screen (smart-HMI) and several buttons to control the motion of the robot arm (to teach points and to simulate working cycles).

Fig. 4.4 - KUKA LBR iiwa smartPAD (from [20])

The following picture is a generic representation of the mechatronic system on which the KUKA LBR iiwa is based, enhancing the advancing technology based on position and torque sensors integration.

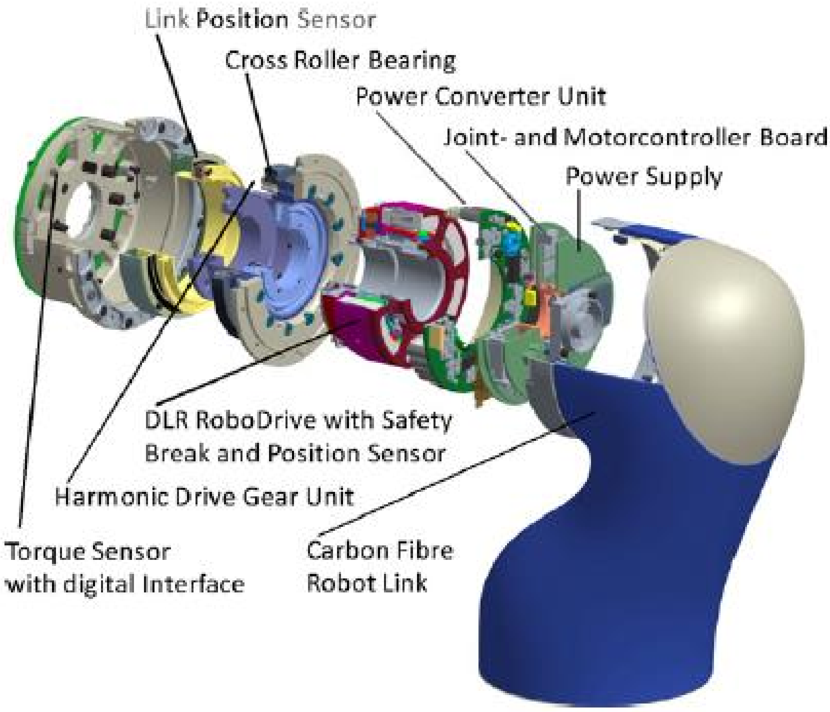

Fig. 4.5 - Mechatronic system of a Light Weight robot arm (from [8])

4.1.2 BALANCING SHAFTS

The insertion of the shafts is a critical operation due to the sub-millimeter alignment and force requirements during the operation.

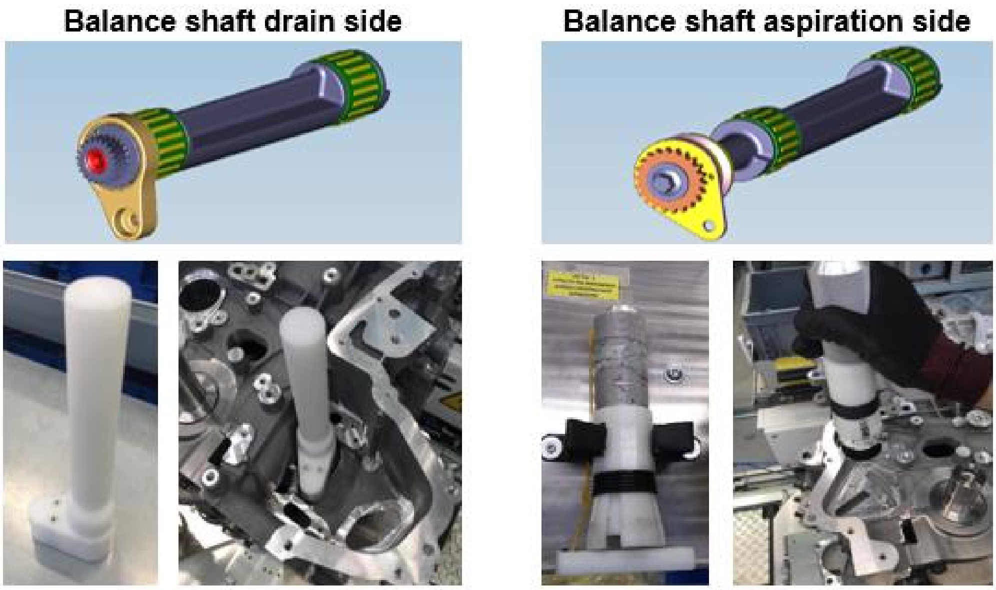

Fig. 4.6 - Balance Shafts insertion (operator charts from Termoli T3 plant)

The two shafts are respectively at Drain side and Aspiration side of the engine basement.

Fig. 4.7 - Details of the shafts and the handling support (from HRC LAB CRF)

4.1.3 GRIPPER

The gripper used is manufactured by Zimmer, mounted on the robot and implemented with two specifically designed claws.

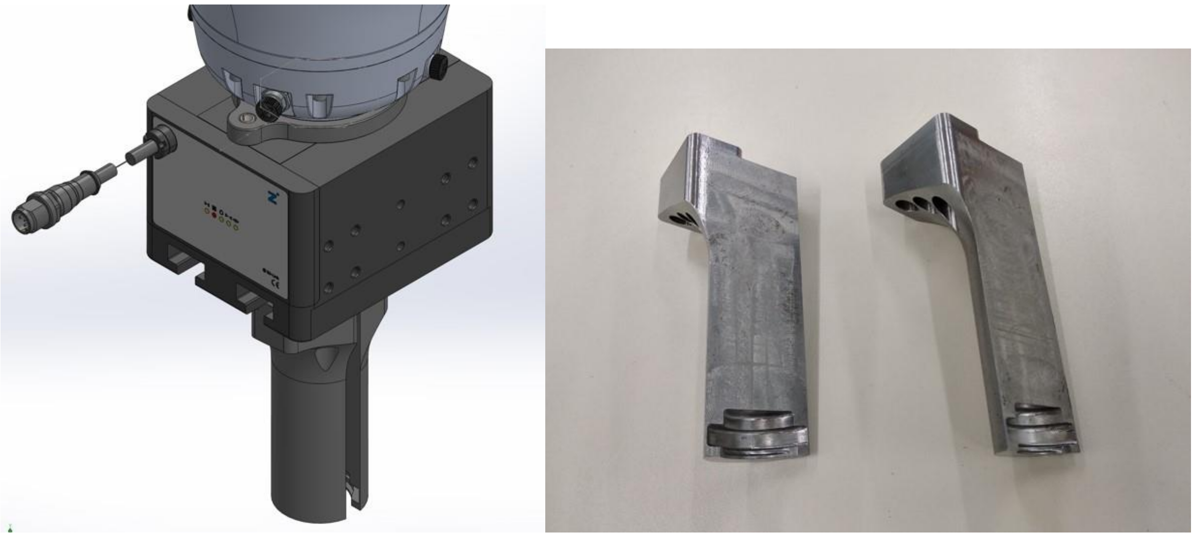

Fig. 4.8 - Gripper CAD and Claws details (from HRC LAB CRF)

The gripping procedure consists in grabbing the gears at the top of the two shafts, achieving a strong and stable grasp, for the whole duration of the cycle, thanks to the designed profiles of the two claws. The multifunctional tooling specification and construction of the claws allow the geometric grip of the two different shafts without adopting complex, expensive and time-consuming operations (based on tool changers, for instance).

Fig. 4.9 - Gripping details (from HRC LAB CRF)

4.1.4 LINEAR AXES

The two linear axes used are manufactured by Schneider Electric. They are products of the family Lexium Max S, consisting in two parallel carriages with toothed belt on both axis and roller guide type. The maximum speed achievable is 8 m/min, and the maximum stroke is respectively 4377 mm and 3014 mm for the one sustaining the engine and the other for the robot and the shafts.

Fig. 4.10 - CAD Linear axis from Schneider Electric (from [19])

4.1.5 CYCLE

As the components of the station are presented, it's also important to describe the cycle, so to introduce the concept of the physical Human-Machine Interaction that it's been developed.

The overall cycle is composed of different parts, performed by the operator and the KUKA robot arm. The cycle, from the positioning of the shaft on the handling support by the operator, has the whole duration of 1 minute, completing both the assembly operations with the linear axis performing 2500 mm of displacement. The sequence of operations is presented as follow.

OPERATOR

- Unpacks the first shaft (drain side shaft) and places it on the handling support
- Unpacks the second shaft (aspiration side shaft) and places it on the handling support

KUKA

- From the start position moves to the position for gripping the first shaft (drain side shaft)
- Completes the gripping procedure

HMI

- Before the robot moves from the gripping position (holding the shaft) it waits for the touch from the operator to continue the cycle

KUKA

- Moves to the insertion position above the engine
- Performs the insertion procedure, assembling the shaft
- Moves back to the position for gripping the second shaft (aspiration side shaft)
- Completes the gripping procedure

HMI

- Before the robot moves from the gripping position (holding the shaft) it waits for the touch from the operator to continue the cycle

KUKA

- Moves to the insertion position above the engine
- Performs the insertion procedure, assembling the shaft
- Moves back to the start position, end of the cycle

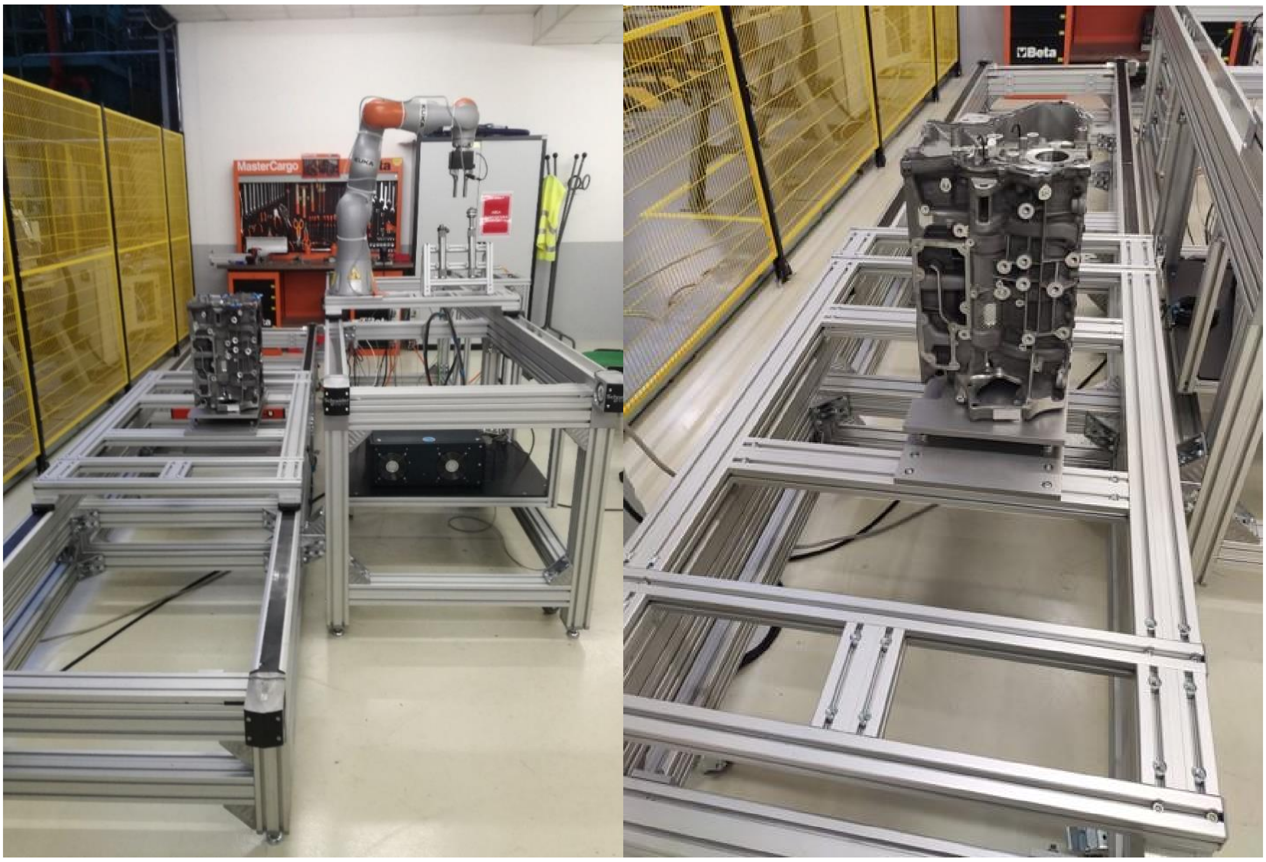

Fig. 4.11 - Workstation and Engine block (from HRC LAB CRF)

4.2 HUMAN-MACHINE INTERACTION

As described before, a physical interaction is part of the overall cycle, and has been developed specifically for this case. This section is entirely dedicated to the description and the execution of the work, starting from the definition of the HMI, through the functional requirements, arriving to the algorithm for the programming implementation.

Basic types of interaction of a robot manipulator with the environment are shown in the following figure:

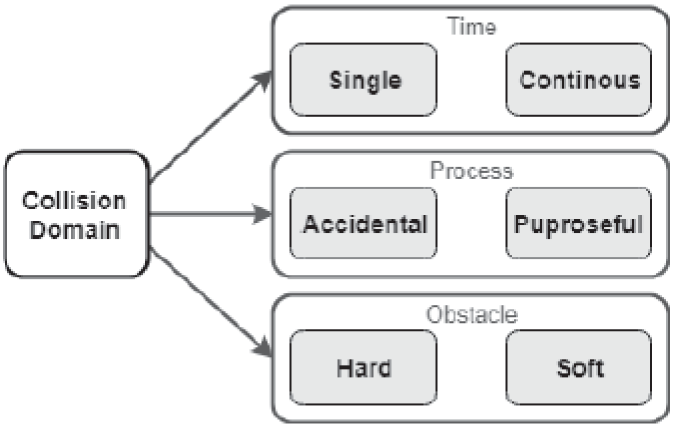

Fig. 4.12 - Types of interaction (from [21])

The classification of collisions starts with the detection of an external force applied to the manipulator. Then, measuring the duration of the contact, it's possible to distinguish single and continuous event, which in turn can be divided into purposeful or accidental, according to the specific current task. For accidental, it's referred as physical interaction not expected from the robot during the operation, while purposeful stands for possible contact. In general, collisions expectations have to be predefined by the operator through control programming. From the analysis of the collision is possible to distinguish between soft or hard objects (soft expect smaller change of torque rate).

From this collision classifications, possible reactions can be defined:

- Touch reaction: stop/continue execution from operator's touch on the robot. Classified as single short purposeful collision with a soft object. This can be used to control the robot operation
- Wait: stop reaction in case of unexpected collision, resuming motion later. Used in both soft and hard collisions to avoid any possible damage. The motion can resume when external contact is no more present. For instance, if the obstacle is the operator, that quickly moves away from the robot's trajectory, it's not necessary to stop the cycle

- Elbow reaction: this one uses kinematic redundancy, in case of contact whit robot elbow, to overcome the obstacle
- End-effector reaction: changing the trajectory to get over the obstacle. This reaction occurs when the collision is located at a point near the end-effector. In this case, if it's acceptable, trajectory should be changed
- Compliant mode: this reaction consists in changing robot configuration manually. Used in case of complex environment to move the end-effector in the desired position

In this particular application, the HMI consists in the touch control on the body of the KUKA robot from the operator, consisting so in a "Touch reaction" collision type.

The interaction is required, due to the complexity of the cycle, considering that it has the role of allowing the robot to continue the application after the gripping procedure is completed. So the operator has the task of supervising that the procedure has been done safely and the grasp of the robot is stable, in order to continue the cycle and perform the insertion without possible failures caused by shaft's oscillation or misalignment. This Interaction combines the repeatability and the precision of the robot with the dexterity and the observation role of the operator, allowing a total safe procedure which should not proceed in case of error.

Below are presented The Functional Requirements, starting from what has been described before, on which was based the HMI development.

FUNCTIONAL REQUIREMENTS:

- Safe collaboration
    - Needed to guarantee operator's safety
    - Evaluable, according to ISO 15066, through the duration and frequency of contact and the zone of the operator's body in which the contact occurs
- High sensibility robot arm
    - In order to easily detect the interaction
    - Evaluable through the sensibility of the torque sensors integrated with the robot
- Possibility to stop the cycle in case of error
    - So to avoid any hazardous phenomena during the operation
    - Achievable in case non-acceptable interaction is performed

Indeed, in order to satisfy these FRs, it has been studied a way to interact with the robot that consists in the operator's touch on the 6^{th} axis after that the robot has completed the gripping phase. This interaction can be done without risks, considering that the position of the robot arm in easily reachable from the operator without colliding or interfering with the rest of the workstation during the cycle. The robot is not in motion and the contact occurs through a soft touch performed by the operator with his hand which is not a sensitive body area.

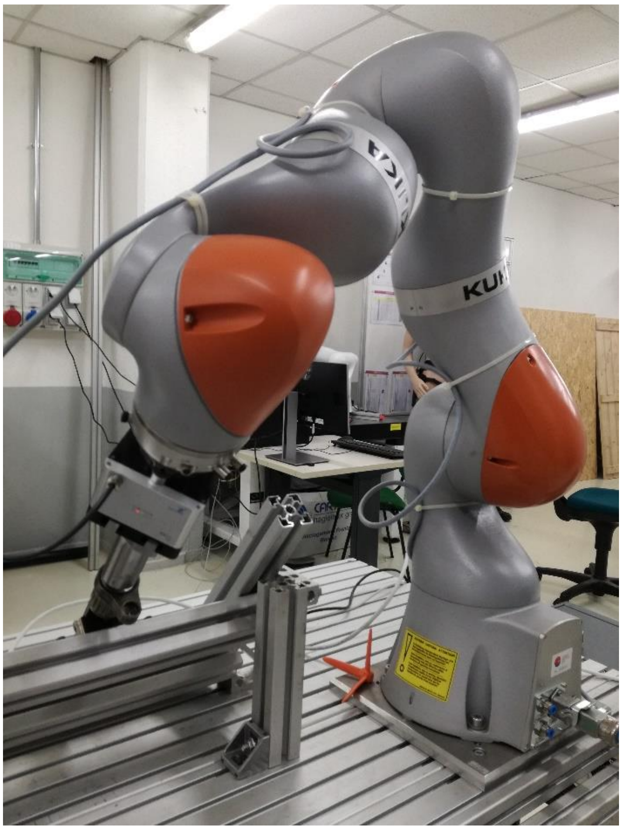

Fig. 4.13 - Detail of the 6th axis of the robot (from HRC LAB CRF)

So, it was decided to develop this interaction by means of variation of the joints torque value of the KUKA robot arm. Indeed, an important feature of this procedure is the detection of the operator's touch. Due to the high sensibility of the KUKA LBR iiwa 14 (torque sensors integrated on each joint), an algorithm has been developed in order to detect even soft touches from the operator, while filtering any external noise (i.e. vibrations).

In the following pages, all the steps of the procedure will be presented.

4.2.1 SIGNAL ACQUISITION

As first step, to study in details the characteristics of this Human-Machine Interaction, a specific signal acquisition activity from the KUKA robot controller has been carried out.

The procedure consisted in the acquisition of the signal's trend from all the torque sensors across the instant of the operator's touch, as expected from the application cycle. The acquisition has been done by means of the function getMeasuredTorque that returns the torque value of each axis in real time, implemented in the interface ITorqueSensitiveRobot inside the Java programming environment. This operation resulted in 7 different signals, sampled at 10 milliseconds intervals, for a total duration of 5 seconds, obtaining so the torque value's trend before, during and after the operator's touch. Different examples of this acquisition are presented below.

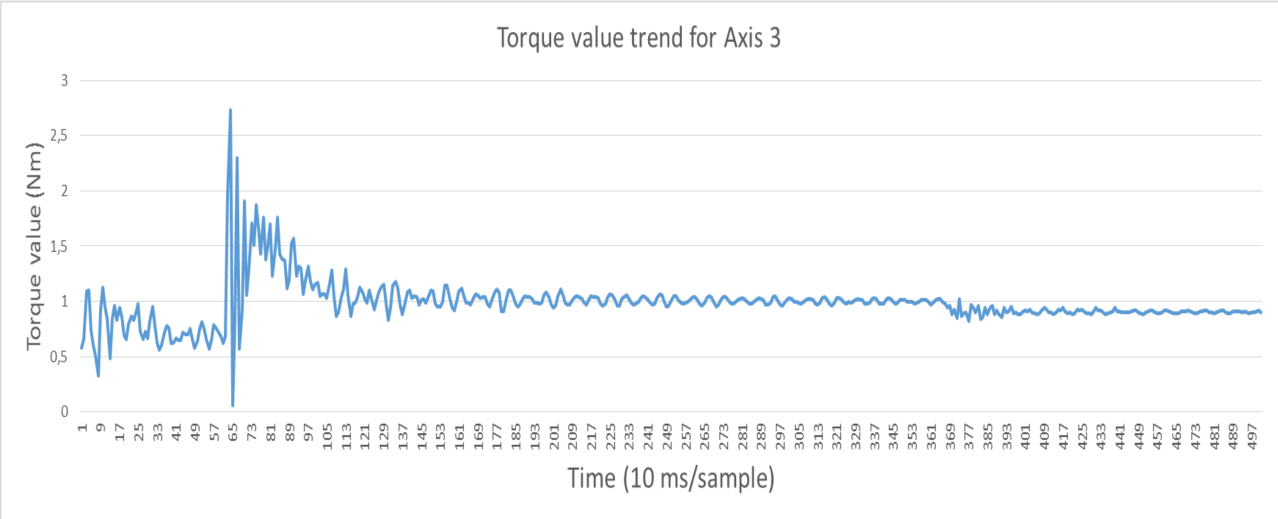

Fig. 4.14 - Torque signal from Axis 3 (drain side shaft)

Then, starting from this procedure, several experimental trials and signal acquisition cycles were performed, changing the energy of the impact (soft and strong), in order to obtain the response of the sensors in all of the possible situations, as it can be seen in the following graphs.

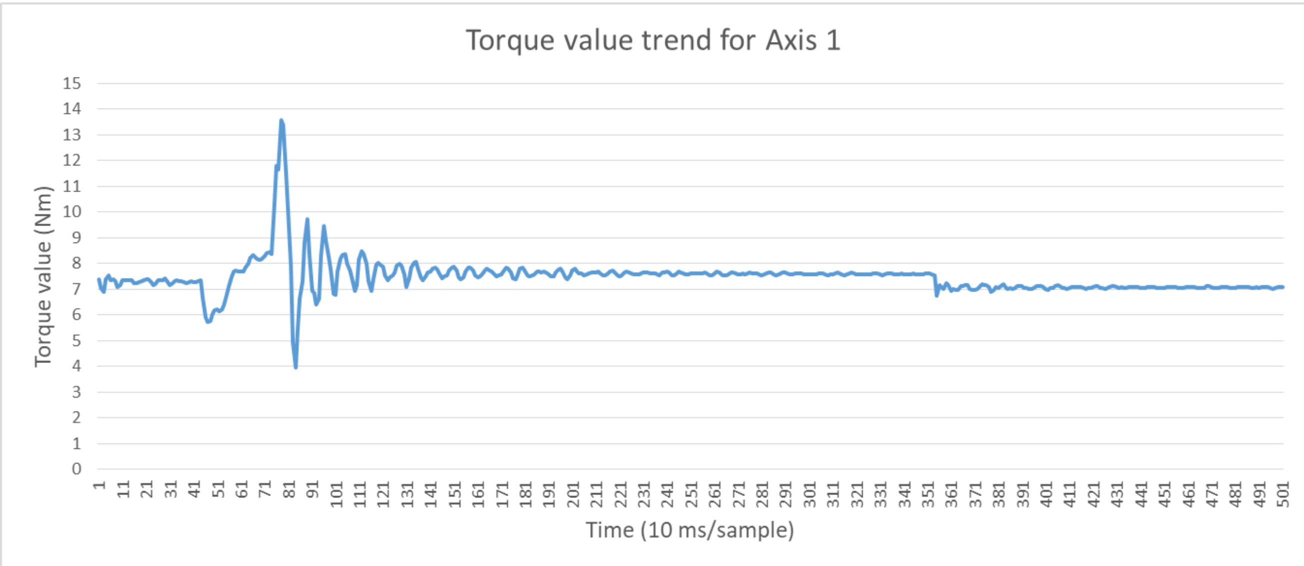

Fig. 4.15 - Torque signal from Axis 1 (aspiration side shaft, strong impact)

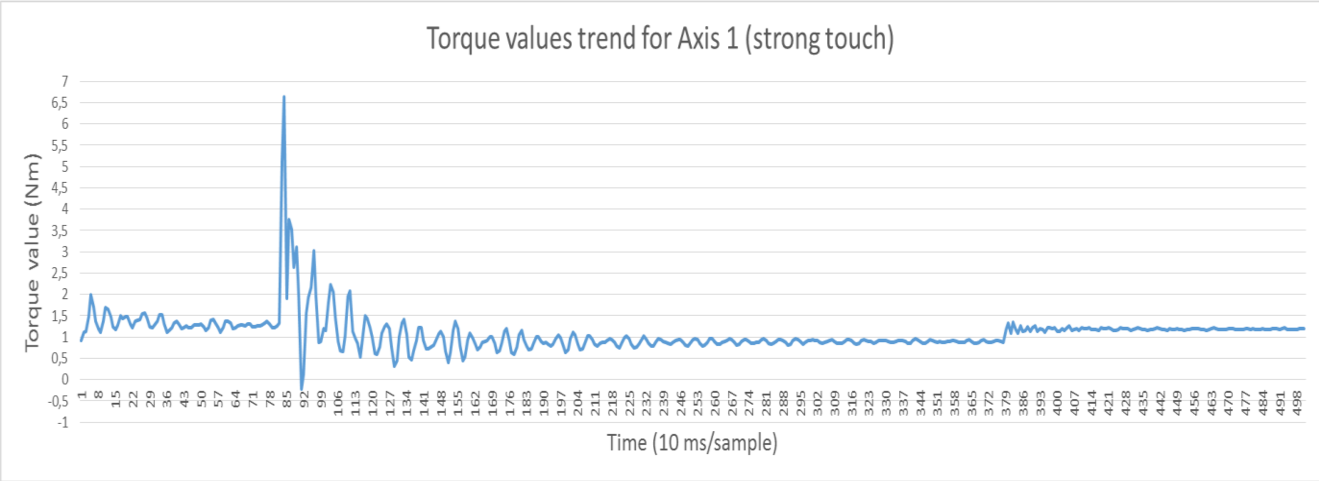

Fig. 4.16 - Torque signal from Axis 1 (drain side shaft, strong impact)

Furthermore, this procedure was done for both the gripping positions - drain side shaft and aspiration side shaft - observing that the differences in the signals (overall trend obtained for the two positions) were negligible, for this type of analysis. So no more distinction between the two positions has been made during the developing (and also in the following graphs).

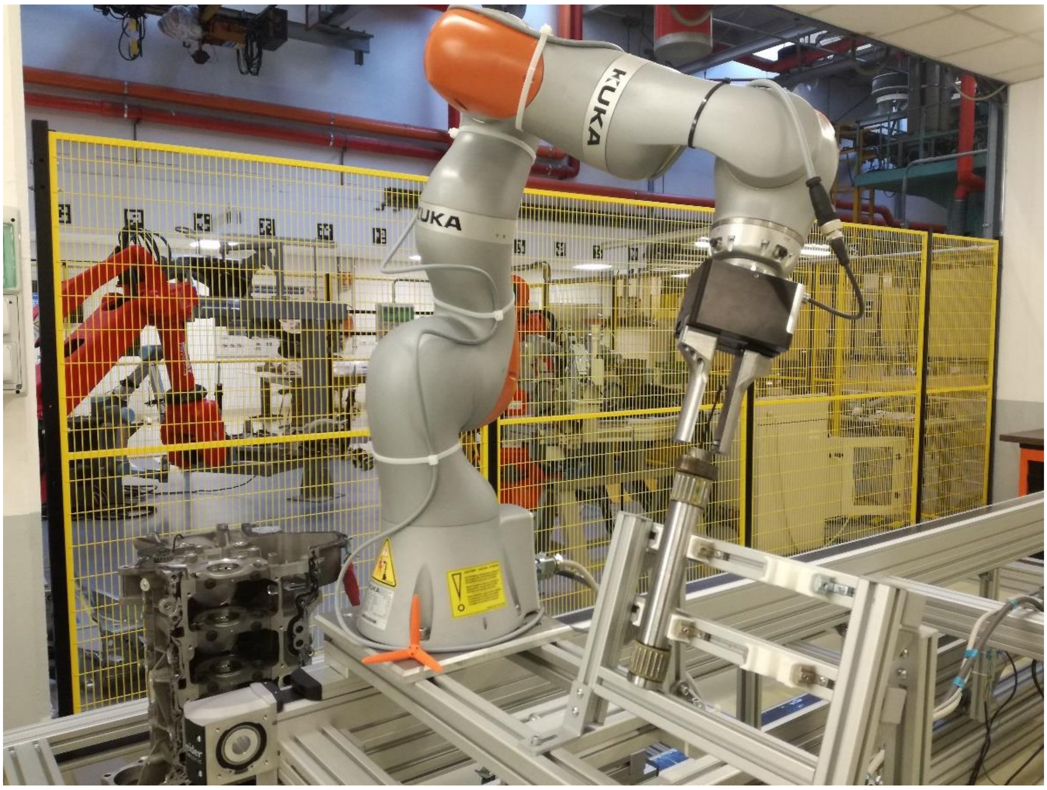

Fig. 4.17 - Gripping position for drain side shaft (from HRC LAB CRF)

As second step, starting from the activities described before, it has been decided to do further analyses on the signals acquired, such as:

- Evaluation of the maximum span (difference between maximum and minimum values of the signal) for each axis, to understand their behavior during the interaction.
- First derivative (approximated at the first order by the differences between consecutive values), so to obtain a signal trend across the zero torque value.

From the results obtained it was observed that, in any case, the 1st axis was the one with the higher span detected, so it was considered as the most sensitive to operator's touch. This conclusion led to the development of the detection algorithm based on its torque sensor's signal, acquired using the function getSingleTorqueValue for the Joint 1, as presented in the graphs below.

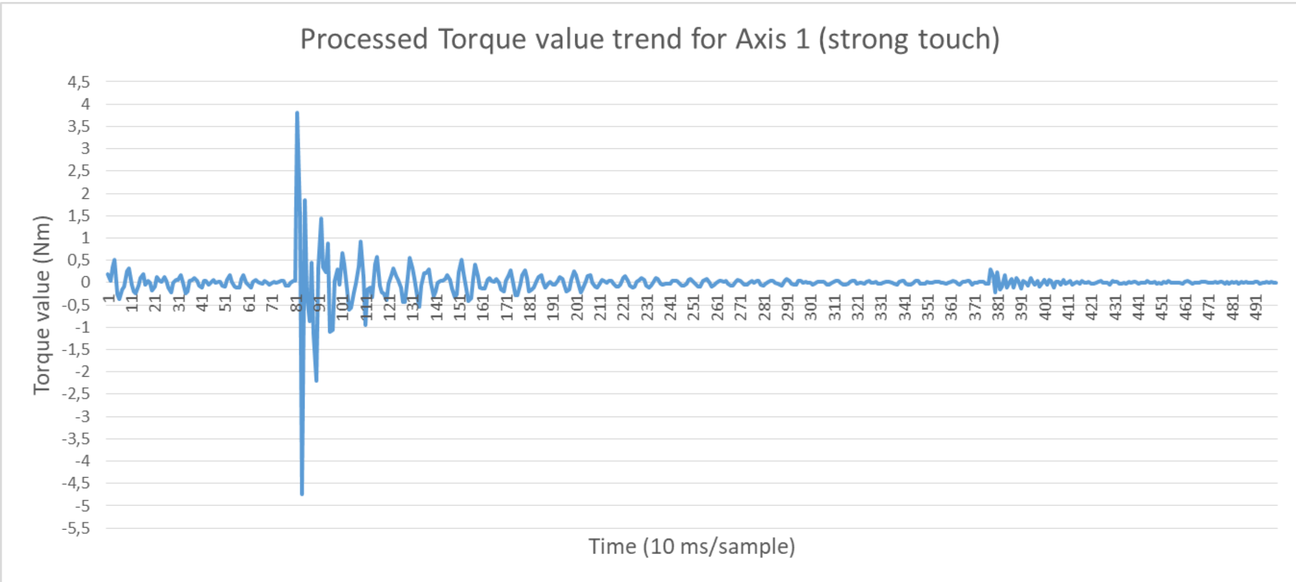

Fig. 4.18 - Processed Torque signal from Axis 1 of Fig.4.16 (strong impact)

Eventually, from the analyses described before on the signals of the 1st axis, for any of the possible cases, it was evident that all the signals shared the same general trend: a first part in which the torque value settles for the new position reached by the robot arm, the central part in which the interaction takes place and the final part in which the signal tends to come back at the value before the perturbation.

In order to remove the continuous component of the signals, first derivative were produced (as described before); in the processed signals, all the significant events are represented by zero crossings. So, the final idea was to develop an algorithm able to detect a large enough zero crossing corresponding to the perturbation of the signal from the HMI, while filtering all the smaller ones that stood for the signal evolution or any external noise. An example of these analyses is presented in the two following graphs, in which it's evident the signal evolution and the processed trend behavior explained before.

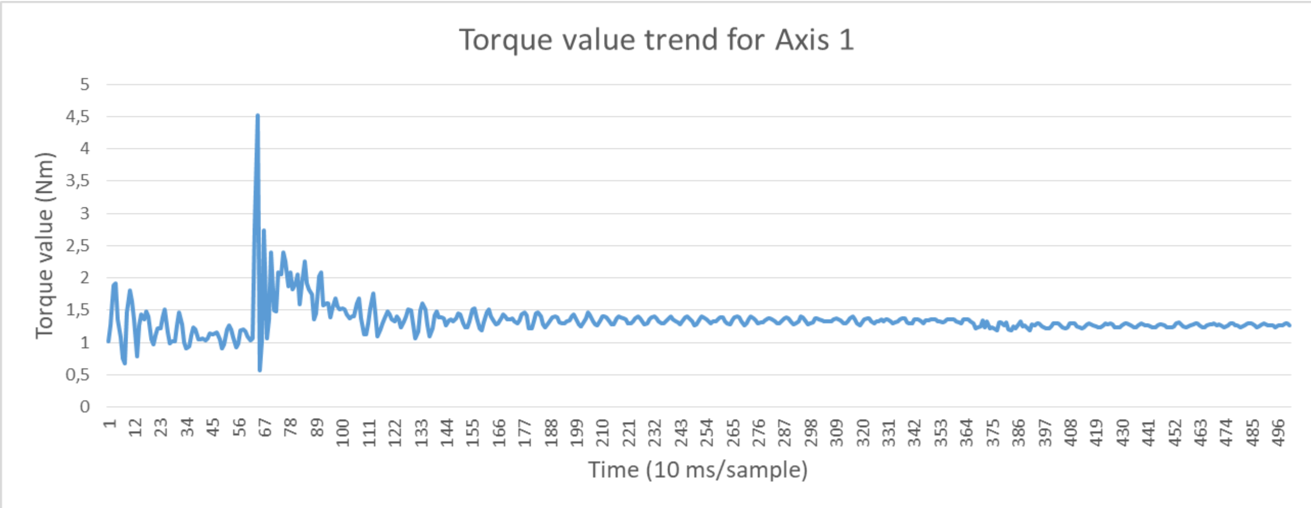

Fig. 4.19 - Torque signal from Axis 1

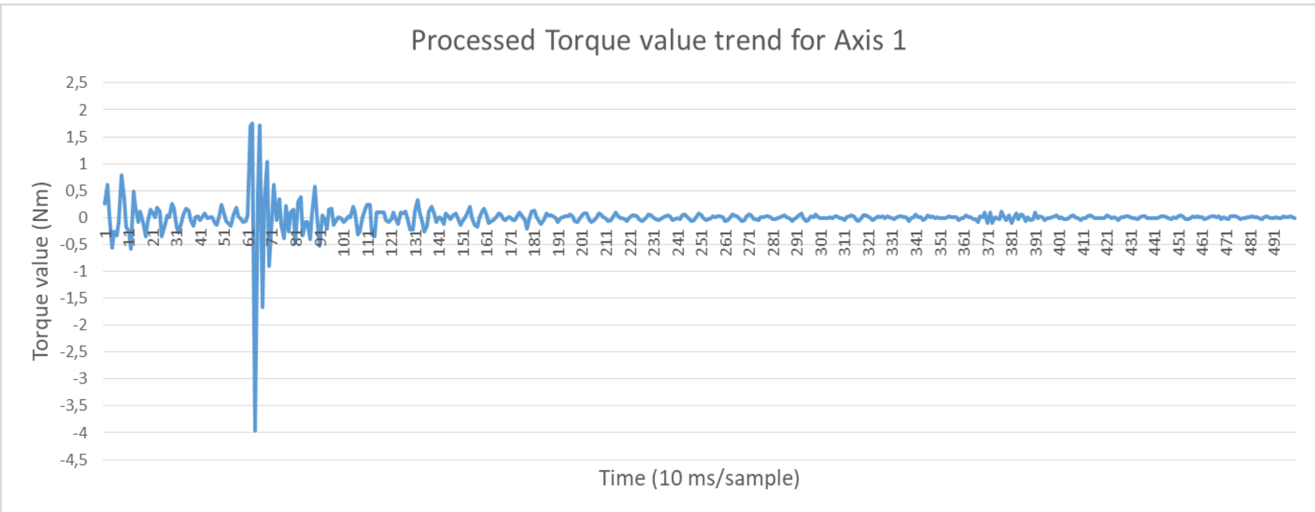

Fig. 4.20 - Processed Torque signal from Axis 1 of Fig. 4.19

4.2.2 IMPLEMENTATION AND STATE MACHINE

As third and final step, from the processed signal's trend, it was observed that the perturbation from the HMI was always represented by a zero crossing with the same descent behavior, indeed the algorithm to develop had to be able to intercept this kind of evolution. So that part of the signal could be identified with a descent front, starting from a local maximum followed by a zero crossing and a local minimum, as clearly presented in the zoomed evolution graph below.

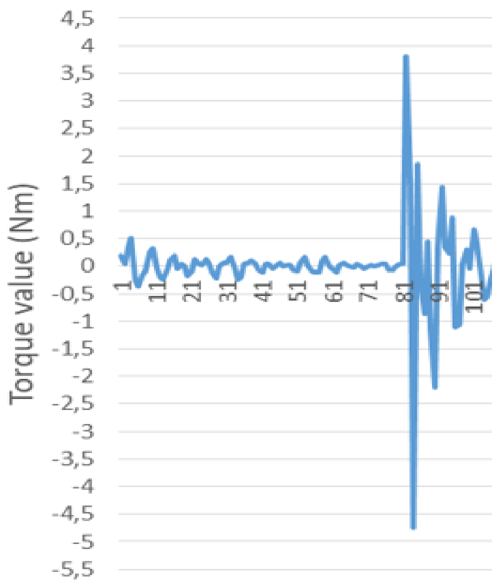

Fig. 4.21 - Detail of the descent front of Fig.4.18

In order to intercept only this zero crossing, the algorithm's behavior had to be fine-tuned by setting some characteristic parameters, such as:

- Threshold positive value for maximum (crossUpThr)
- Threshold negative value for minimum (crossLowThr)
- Limit value for the span, as the difference between the two points, in order to filter all the smaller fronts (TorqueLimit)

So, it was necessary to identify local points able to satisfy these requirements, and to analyze the signal in real time and develop the algorithm an implementation by means of a State Machine was conducted. In the following pages a general description and the progression of the work is presented.

A State Machine is a mathematical computational model, it's an abstract machine that in any given time can be in a specific state of the finite ones implemented. A transition from a state to another can be caused by an external input or if a specific condition is satisfied. A State Machine is defined by the list of its states and the conditions for the transition between them.

In this application the State Machine is used to scan in real time the torque values signal from the 1st axis during the HMI, in order to find a succession of local maximum and minimum that satisfies the requirements from the parameterization, such as:

- Local positive maximum higher than crossUpThr
- Local negative minimum lower than crossLowThr
- Difference between the two values higher than TorqueLimit

So, to conduct this type of analysis the State Machine is composed of 4 different States:

1. State 1 (S1) for the local minimum search
2. State 2 (S2) for the local maximum search
3. State 3 (S3) in case of successive torque values equal to local minimum
4. State 4 (S4) in case of successive torque values equal to local maximum

From the definition of the States, the entering parameters of the machine are:

- Current torque value from the signal: Data (as the difference of consecutive values)
- crossUpThr, crossLowThr, TorqueLimit
- Local maximum and minimum: Max, Min (set at the beginning equal to Data)
- State = S1 (first state is State 1)
- Count, that is a counter used in case of successive Data equal
- CountThr, which stands for a threshold for the counter

To fully understand the behavior of the algorithm in the following image a diagram of the State Machine is presented. The arrows stand for the transitions between different states while the conditions related and the corresponding actions taken, are respectively colored in red and blue. Then follows an explanation of each State, to describe in detail what has been introduced until now.

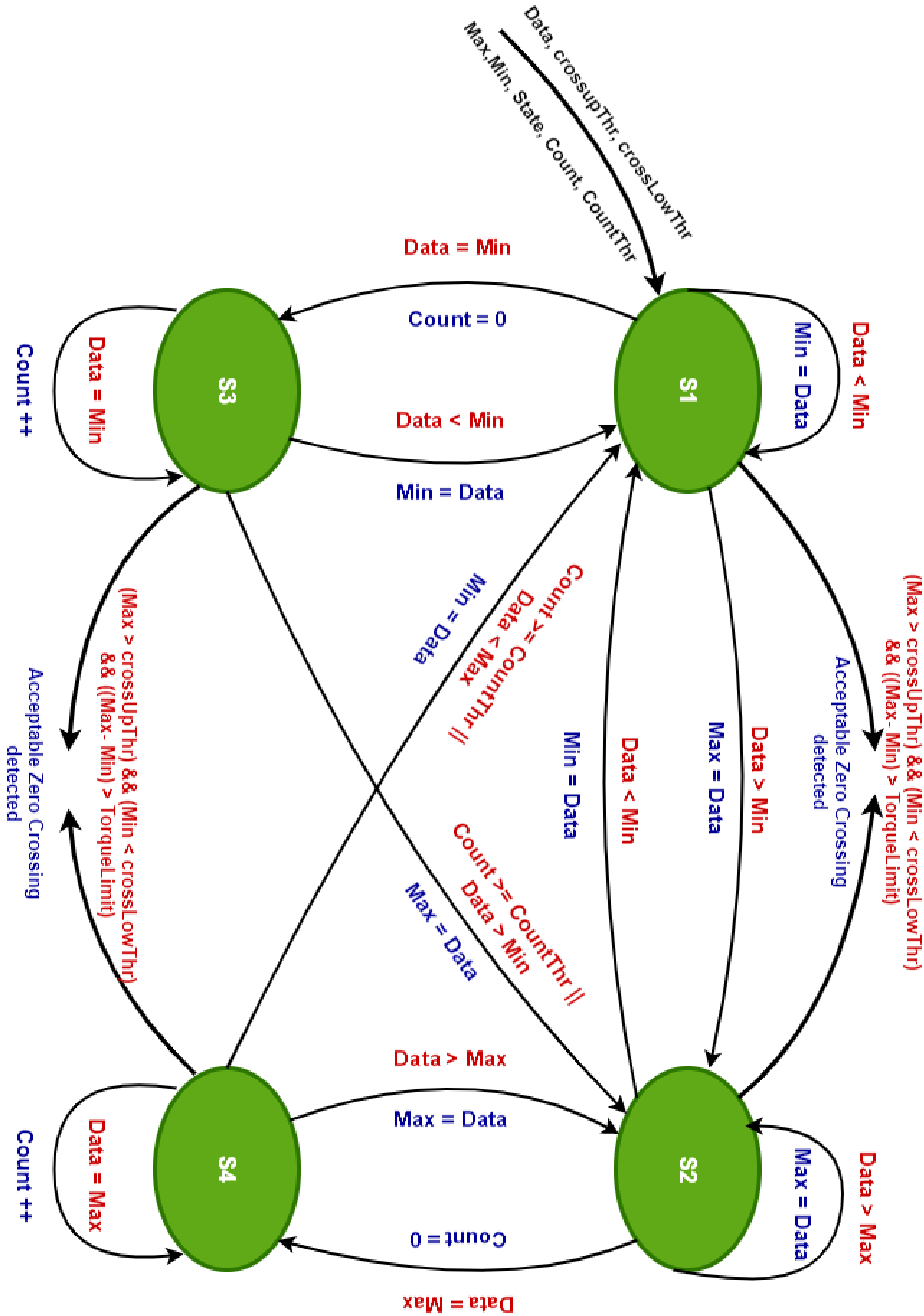

Fig. 4.22 - State Machine Diagram for robust touch detection

4.2.2.1 STATE 1:

This state is the first entered and is dedicated to the local minimum search.

Starting from the current torque value Data, the conditions analyzed and the corresponding actions taken are:

1.

$$if\ (Data\ < Min) \rightarrow Min = Data \quad (1)$$

$$\Lsh [if\ (Min < crossLowThr)\ \&\&\ (Max > crossUpThr)\ \&\&\ \big((Max - Min) > TorqueLimit\big) \rightarrow Acceptable\ Zero\ Crossing\ detected] \quad (2)$$

In this case the current value is lower than the previous local minimum found, so its new value has to be updated. Since a new minimum has been detected, a possible acceptable descent front may have been intercepted. If the conditions are satisfied, the research is completed, the interaction is performed and the cycle can continue.

2.

$$if\ (Data > Min) \rightarrow Max = Data \quad (3)$$

$$\rightarrow State = S2 \quad (4)$$

In this possibility, the current value is higher than the previous minimum, this means that a new possible maximum has been founded, its value updated and the state changed, going in the local maximum search.

3.

$$if\ (Data = Min) \rightarrow Count = 0 \quad (5)$$

$$\rightarrow State = S3 \quad (6)$$

In this particular occurrence, the current Data is equal to the local minimum previously found, so the counter is initialized and the state is updated, for the case of successive equal values.

4.2.2.2 STATE 2:

This state is dedicated to the local maximum search.

Starting from the current torque value Data, the conditions analyzed and the corresponding actions taken are:

1.

$$if\ (Data > Max) \rightarrow Max = Data \tag{7}$$

In this case the current value is higher than the previous local maximum found, so its new value has to be updated.

2.

$$if\ (Data\ < Max) \rightarrow Min = Data \tag{8}$$

$$\Lsh\ [if\ (Min < crossLowThr)\ \&\&\ (Max > crossUpThr)\ \&\&\ \big((Max - Min) > TorqueLimit\big) \rightarrow Acceptable\ Zero\ Crossing\ detected] \tag{9}$$

$$\rightarrow State = S1 \tag{10}$$

In this possibility, the current value is lower than the previous maximum, this means that a new possible minimum has been founded, its value updated and the state changed, going in the local minimum search. Since a new minimum has been detected, a possible acceptable descent front may have been intercepted. If the conditions are satisfied, the research is completed, the interaction is performed and the cycle can continue.

3.

$$if\ (Data = Max) \rightarrow Count = 0 \tag{11}$$

$$\rightarrow State = S4 \tag{12}$$

In this particular occurrence, the current Data is equal to the local maximum previously found, so the counter is initialized and the state is updated, for the case of successive equal values.

4.2.2.3 STATE 3:

This state is entered only in case of successive Data equal while searching the local minimum. It means that the signal's trend is horizontal. A threshold for the number of following equal points has been arbitrarily set, in order to stop the minimum search if the signal has gone too far from the local maximum already detected.

Starting from the current torque value Data, the conditions analyzed and the corresponding actions taken are:

1.

$$if\ (Data = Min) \rightarrow Count++ \tag{13}$$

$$\Lsh\ [if\ (Count \geq CountThr) \rightarrow Max = Data, State = S2] \tag{14}$$

In this case the current value is equal to the previous, so the signal is horizontal and the counter is incremented. If the counter has reached the threshold value, the previous maximum is lost, so its value has to be updated and state changed.

2.

$$if\ (Data\ < Min) \rightarrow Min = Data \tag{15}$$

$$\Lsh\ [if\ (Min < crossLowThr)\ \&\&\ (Max > crossUpThr)\ \&\&\ \big((Max - Min) > TorqueLimit\big) \rightarrow Acceptable\ Zero\ Crossing\ detected] \tag{16}$$

$$\rightarrow State = S1 \tag{17}$$

In this possibility, the current value is lower than the previous minimum, this means that a new possible minimum has been founded, its value updated and the state changed, going in the local minimum search. Since a new minimum has been detected, a possible acceptable descent front may have been intercepted. If the conditions are satisfied, the research is completed, the interaction is performed and the cycle can continue.

3.

$$if\ (Data > Min) \rightarrow Max = Data \tag{18}$$

$$\rightarrow State = S2 \tag{19}$$

In this case the current value is higher than the previous local minimum found, so the new maximum value has to be updated and the state changed.

4.2.2.4 STATE 4:

This state is entered only in case of successive Data equal while searching the local maximum. It means that the signal's trend is horizontal. A threshold for the number of following equal points has been arbitrarily set, in order to stop the maximum search if the signal has gone too far from the local minimum already detected.

Starting from the current torque value Data, the conditions analyzed and the corresponding actions taken are:

1.

$$if\ (Data = Max) \rightarrow Count++ \tag{20}$$

$$\Lsh\ [if\ (Count \geq CountThr) \rightarrow Min = Data, State = S1] \tag{21}$$

In this case the current value is equal to the previous, so the signal is horizontal and the counter is incremented. If the counter has reached the threshold value, the previous minimum is lost, so its value has to be updated and state changed.

2.

$$if\ (Data > Max) \rightarrow Max = Data \tag{22}$$

$$\rightarrow State = S2 \tag{23}$$

In this case the current value is higher than the previous local maximum found, so its new value has to be updated and the state changed.

3.

$$if\ (Data\ < Max) \rightarrow Min = Data \tag{24}$$

$$\Lsh [if\ (Min < crossLowThr)\ \&\&\ (Max > crossUpThr)\ \&\&\ \big((Max - Min) > TorqueLimit\big) \rightarrow Acceptable\ Zero\ Crossing\ detected] \tag{25}$$

$$\rightarrow State = S1 \tag{26}$$

In this possibility, the current value is lower than the previous maximum, this means that a new possible minimum has been founded, its value updated and the state changed, going in the local minimum search. Since a new minimum has been detected, a possible acceptable descent front may have been intercepted. If the conditions are satisfied, the research is completed, the interaction is performed and the cycle can continue.

4.3 CONCLUSIONS

So, in the end, this algorithm of the State Machine has been converted into a Switch-Case code (see Appendix), where each case deals with a specific state, and implemented into the whole cycle program.
Several trials were performed, testing the quickness in real time of the procedure, obtaining satisfying results. The following graphs represent some of the tests and acquisitions made, so to evidence the behavior of the algorithm by setting specific parameters (the evolutions are already been processed, because the real time analyses are made on this type of torque value trend).

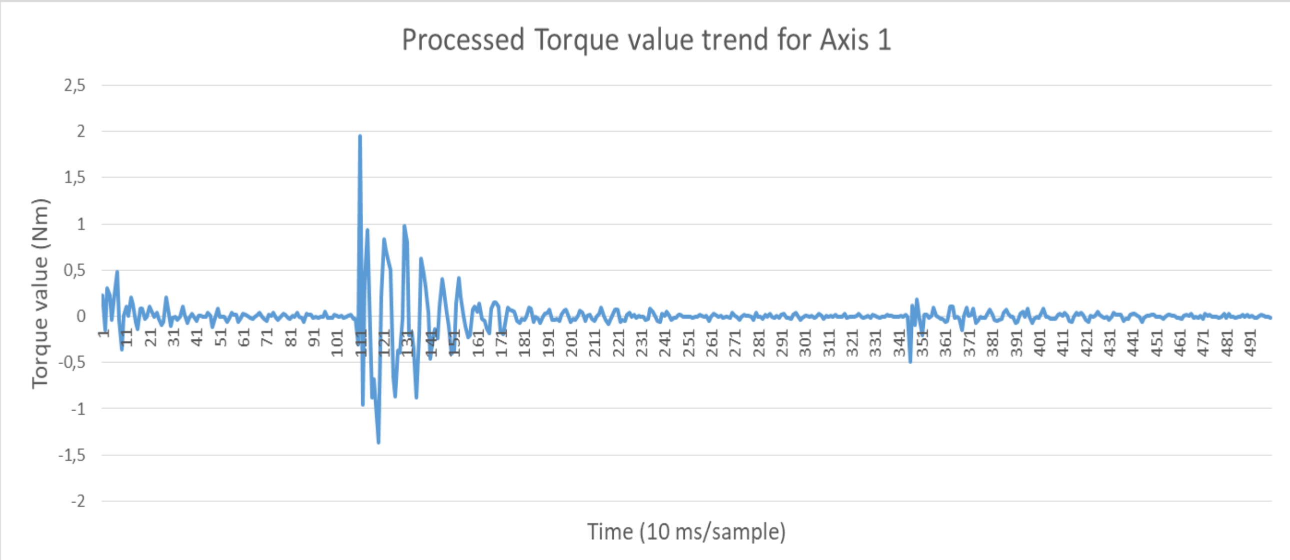

Fig. 4.23 - Processed Torque signal from Axis 1 (soft impact)

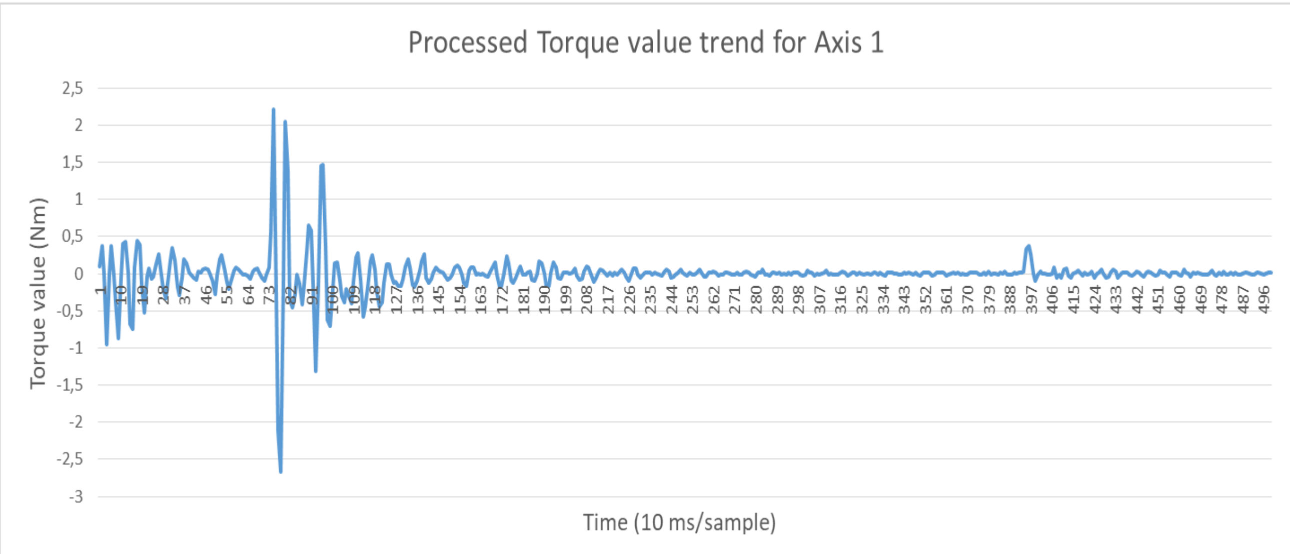

Fig. 4.24 - Processed Torque signal from Axis 1 (soft impact)

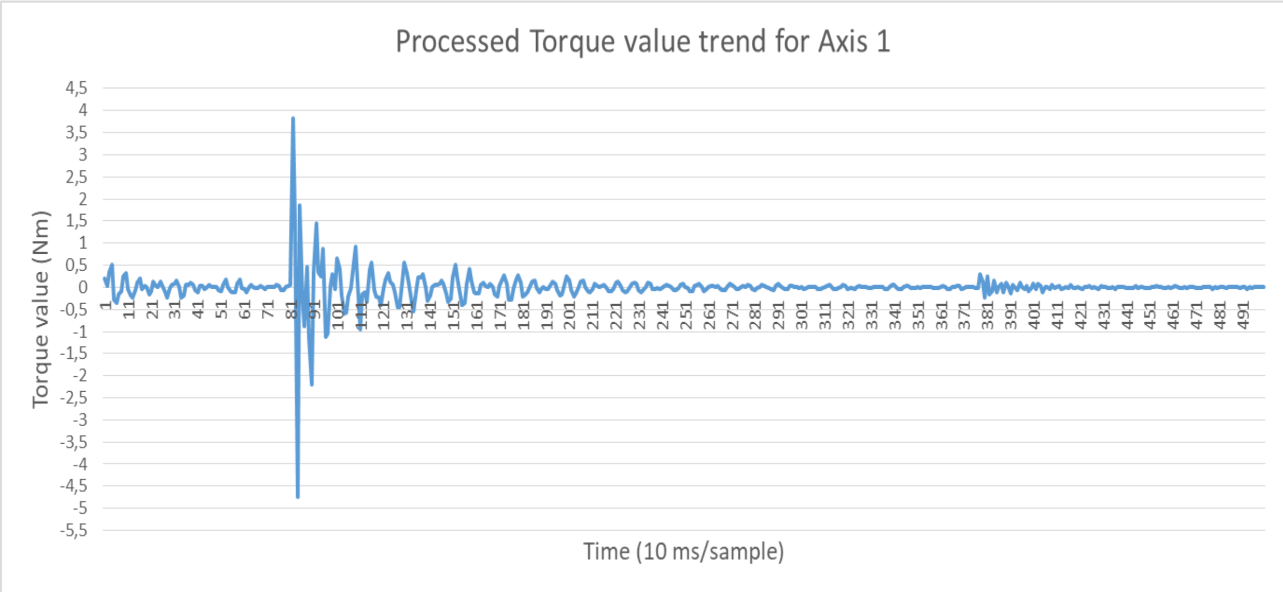

Fig. 4.25 - Processed Torque signal from Axis 1 (strong impact)

Indeed, for example, by setting this parameters:

- 0.5 Nm as cross threshold value both for maximum and minimum
- 2 Nm as the torque limit (difference between successive maximum and minimum)
- 3 as threshold value for the constant value counter

It's evident that in all the previous cases presented (soft and strong touch) the algorithm is able to filter all the vibrations and signal evolutions, detecting only the perturbations caused by the operator's touch as soon as it occurs, resulting so in a very fast interaction within the whole cycle. Also, it was observed, during the different tests, that the worst possible case was the one in which the operator had to perform 2 touches on the robot arm to continue the cycle, probably because the first one resulted too soft respect to the parameters set in the function, but never occurred a case in which the robot moved on its own without the operator permission. So this underlines the functionality of this implementation, both in terms of efficiency and of safety for the cycle.

Finally, with this HMI implementation, the operator can monitor the gripping phase, stopping the cycle in case of error just by not touching the KUKA arm, which will stop the execution in case of acceptable zero crossing not detected within a time limit set arbitrarily in the program. While just by performing a soft impact, the cycle will continue as fast as possible, not slowing down the operation and eventually the whole assembly line.

As explored in the previous pages, the Human-Machine Collaboration is a concept continuously evolving, due to the implementation of new technologies and features on robot arms, allowing safe interactions and workspaces sharing by means of smart detection systems and high sensitivity. Several benefits can be obtained by introducing robot assistance, such as increase in production volume and quality work together with reduction in ergonomic and working stress for human workers.

But, while designing and arranging a collaborative workstation, it’s important to consider which level of collaboration is necessary for the application, for example:

- If contact or proximity are not required, it’s possible to use Safety-Rated Monitored Stop or Speed & Separation monitoring, whichever is the most suitable (costs, speed of execution)
- If proximity and contact are necessary, it’s needed to study the nature and the intensity of the interaction and which collaborative robot is the most appropriate for the application

And also, other important aspects to be considered are:

- Evaluation of which operations are attributed to the robot and which to the operator, by means of task planning models
- Execution time of the cycle
- Available space and security constrains related to the ISO standards

In this work has been presented just one of the possible exploitation of a HRC in an automotive assembly line, describing in detail the development of a specific Human-Machine Interaction, but several applications and cycles can be performed, and in order to do that, analyses and models implementations can be applied to fulfill the requirements and obtain benefits of different kinds.

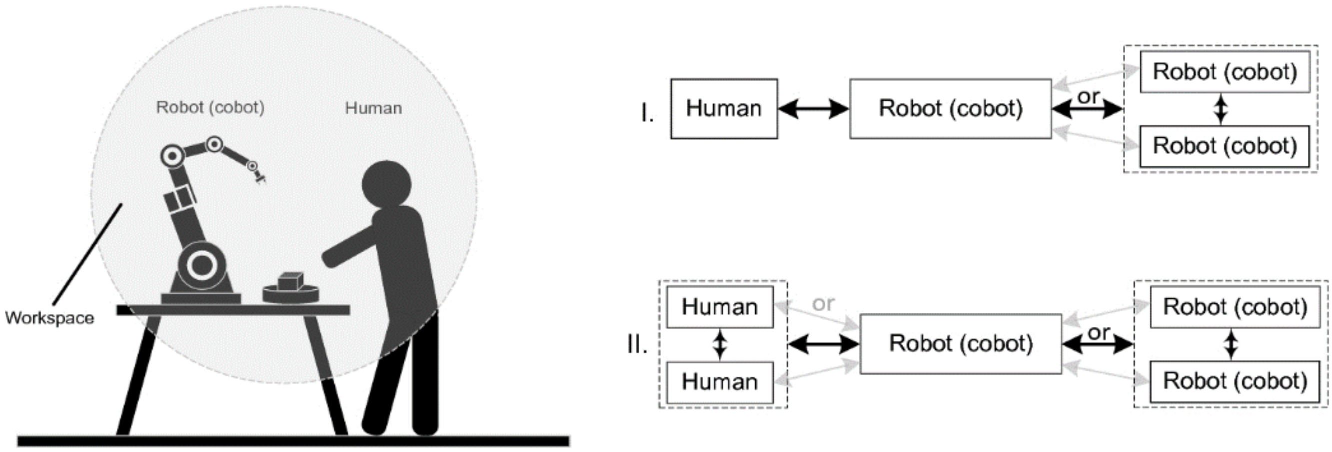

Fig. 4.26 - Human-Robot Collaboration sharing workspace (from link.springer.com)

5 REFERENCES

1. EN ISO 10218-1:2011, *Safety requirements for industrial robots. Part 1: Robots.*
2. EN ISO 10218-2:2011, *Safety requirements for industrial robots. Part 2: Robot Systems and integration.*
3. ISO/TS 10566:2016, *Robots and robotic devices - Collaborative Robots.*
4. David S. Cochran, Jorge F. Arinez, Micah T. Collins & Zhuming Bi (2016): *Modelling of human–machine interaction in equipment design of manufacturing cells, Enterprise Information Systems.*
5. George Michalos, Sotiris Makris, Jason Spiliotopoulos, Ioannis Misios, Panagiota Tsarouchi, George Chryssolouris (2014): *ROBO-PARTNER: Seamless Human-Robot Cooperation for Intelligent, Flexible and Safe Operations in the Assembly Factories of the Future.*
6. Teiwes Johannes, Banziger Timo, Kunz Andreas, Wegener Konrad (2016): *Identifying the Potential of Human-Robot Collaboration in Automotive Assembly Lines using a Standardised Work Description.*
7. Roberta Nelson Shea, Rockwell Automation (2016): *Collaborative Robot Technical Specification ISO/TS 15066 Update.*
8. J. Kruger, T.K. Lien, A. Verl (2009): *Cooperation of human and machines in assembly lines.*
9. Rainer Muller, Matthias Vette, Ortwin Mailahn (2016): *Process-oriented task assignment for assembly processes with human-robot interaction.*
10. TechVision Group of Frost & Sullivan (2018): *Cobots, Cognitive Manufacturing, Metal 3D printing, Hybrid Manufacturing and Nanofabrication Drive growth in Manufacturing.*
11. TechVision Group of Frost & Sullivan (2016): *3DP, AR, VR, Advanced Robotics, and IIoT Drive Opportunities in Smart Manufacturing.*
12. Sten Grahn, Bjorn Langbeck: *Benefits of collaborative robots in assembly-an evaluation scheme.*
13. Tsarouchi, P., Michalos, G., Makris, S., Athanasatos, T., Dimoulas, K., & Chryssolouris, G. (2017): *On a human–robot workplace design and task allocation system*. *International Journal of Computer Integrated Manufacturing*.
14. Centro Ricerche Fiat S.C.p.A: *Linee guida per l'individuazione e la validazione dei requisiti HMI per la collaborazione uomo-robot.*
15. Zanella A. (2019): *HRC industrial workapplications an industrial point of view, Presentation at ERF Workshop: Collaborative robots for industrial and professional service tasks.*
16. Zanella A. (2019): *Interactive robotics in industrial environment, Presentation at I-RIM, Roma.*
17. Zanella A. (2018): *Human-Robot Collaboration, criteria for implementation in the shop floor.*
18. Papakostas N, Michalos G, Makris S, Zouzias D, Chryssolouris G (2011): *Integrating Digital Manufacturing and Simulation Tools in the Assembly Design Process: A Cooperating Robots Cell Case.*
19. Schneider Electric: *Lexium MAX:MAXS3BR- Data Sheet.*
20. www.kuka.com
21. Stanislav Mikhel, Dmitry Popov, Shamil Mamedov, Alexandr Klimchik: *Advancement of Robots With Double Encoders for Industrial and Collaborative Applications.*
22. Alessandro Zanella, Alessandro Cisi, Marco Constantino, Massimo di Pardo, Giorgio Pasquettaz, Giulio Vivo (2017): *Criteria definition for the identification of HRC use cases in automotive manufacturing.*

6 ACKNOWLODGMENTS

Many people deserve to be thanked for their contribution along my path in these last 5 years.

First of all, my supervisor Prof. Eugenio Brusa for his support since the Bachelor Degree Thesis, but especially for this project, helping me in the elaboration of this work and in general for the internship.

My company tutor Dott. Giulio Vivo, for teaching me so many things and treating me so well that I will never forget this experience, for all the duration of the internship and thesis period. I will make good use of what I learned from you, thank you. But I would also like to thank all the colleagues of the Centro Ricerche Fiat for making me feel like a member of the group since the beginning and part of something very interesting and challenging, hoping that in future I will collaborate again with you.

Of course I want to thank my family, my parents Angela and Massimo and my sister Valentina, and I will never stop doing it, for their support in everything during this years, in bad and good times, never making me feel alone or forced in anything. Without you I could not have achieved all this results. Thank you, really.

But also all my friends, old and new, from my birthplace, my teen years and my PoliTo courses, for their presence and their support along the path. We’ve already enjoyed one celebration, but I know that I want to celebrate this one too, whit all of you, and I’m sure you too.

In the end, a special thank is for myself, for never giving up and achieving this result, which is just the beginning of a long path, but still not an everyday thing. GG.

7 APPENDIX

As explained before, this appendix presents the conversion of the algorithm for the touch detection into the code for the implementation in the programming cycle. The first rows are dedicated to the import of the various JAVA interfaces for the use of the different functions provided by the KUKA programming environment, then the effective function code begins (it was named waitUntilTouch for the effective role in the cycle). The first part is for the declaration of the variable used (named as it was described in the algorithm development before), then there is the acquisition of the torque values by using the function getSingleTorqueValue(JointEnum.J1), the difference for obtaining the processed values and in the end the Switch-Case code, in which each case deals with a specific state, as presented before. (Conversion of the State-Machine diagram).

Follows the code implemented.

```
package application;

import com.kuka.common.ThreadUtil;
import com.kuka.roboticsAPI.deviceModel.JointEnum;
import com.kuka.roboticsAPI.deviceModel.LBR;
import com.kuka.roboticsAPI.sensorModel.TorqueSensorData;


   public int waitUntilTouch(long timeout, double torqueLimit, int constStepLimit, double
zcrossUpThr, double zCrossLowThr)
   {
          int i, j, state;
          double max, min;
          double dat, dat1, dif;
          TorqueSensorData measuredData;

          measuredData = _lbr.getMeasuredTorque();
          dat1 = measuredData.getSingleTorqueValue(JointEnum.J1);
          ThreadUtil.milliSleep(10);

          measuredData = _lbr.getMeasuredTorque();
          dat = measuredData.getSingleTorqueValue(JointEnum.J1);
          ThreadUtil.milliSleep(10);

          dif = dat - dat1;
      min = max = dif;
      dat1 = dat;
      state = 1;

      for (i=3, j=0;;i++)
```

```
        {
            measuredData = _lbr.getMeasuredTorque();
            dat = measuredData.getSingleTorqueValue(JointEnum.J1);
            ThreadUtil.milliSleep(10);

           dif = dat - dat1;
           dat1 = dat;

           switch (state)
           {
           case 1: // State 1 Minimum search
              if (dif < min)
              {
                 min = dif;
                 if ((max > zcrossUpThr) && (min < zCrossLowThr) && ((max - min) > torqueLimit))
                 {
                    System.out.printf("Acceptable z.c. detected at i --> %d delta --> %f state --> %d", i,
(max-min), state);
                    return 0;
                 }
              }
              else if (dif > min)
              {
                 max = dif;
                 state = 2;
              }
              else
              {
                 j = 0;
                 state = 3;
              }
              break;

           case 2: // State 2 Maximum search
              if (dif > max)
              {
                 max = dif;
              }
              else if (dif < max)
              {
                 min = dif;
                 if ((dif < max) && ((max > zcrossUpThr) && (min < zCrossLowThr) && ((max - min) >
torqueLimit)))
                 {
                    System.out.printf("Acceptable z.c. detected at i --> %d delta --> %f state --> %d", i,
(max-min), state);
                    return 0;
                 }
```

```
        state = 1;
      }
      else
      {
        j = 0;
        state = 4;
      }
      break;

    case 3: // State 3 Successive equal minimum
      if (min == dif)
      {
        j++;
        if (j >= constStepLimit)
        {
          max = dif;
          state = 2;
        }
      }
      else if (dif < min)
      {
           min = dif;
           if ((max > zcrossUpThr) && (min < zCrossLowThr) && ((max - min) > torqueLimit))
           {
                  System.out.printf("Acceptable z.c. detected at i --> %d delta --> %f state -->
%d", i, (max-min), state);
                  return 0;
           }
        state = 1;
      }
      else
      {
        max = dif;
        state = 2;
      }
      break;

    case 4: // State 4 Successive equal maximum
      if (max == dif)
      {
        j++;
        if (j >= constStepLimit)
        {
          min = dif;
          state = 1;
        }
      }
      else if (dif > max)
```

```
            {
               max = dif;
               state = 2;
            }
            else
            {
               min = dif;
                    if ((max > zcrossUpThr) && (min < zCrossLowThr) && ((max - min) > torqueLimit))
                    {
                              System.out.printf("Acceptable z.c. detected at i --> %d delta --> %f state -->
%d", i, (max-min), state);
                              return 0;
                    }
               state = 1;
            }
            break;
         }
         if (i*10 > timeout)
         {
            System.out.printf("Timeout elapsed, not acceptable z.c. detected after %d ms.", i*10);
          return -1;
         }
         ThreadUtil.milliSleep(10);
      }
   }
}
```