Project ref. no.	**260159**
Project acronym	**ROBOFOOT**
Project full title	**Smart robotics for high added value footwear industry**
Type	Deliverable
Dissemination level	**PU**
Contractual date of delivery	31/03/2012
Actual Date of Delivery	18/05/2012
Title of document	D2.3 Safety issues for robot assisted shoe production
Version	V1.0
Number of pages	89
Partner Responsible	ITIA-CNR
Other Contributors	Consortium
Author	Nicola Pedrocchi
	Enrico Villagrossi, Lorenzo Molinari Tosatti, Federico Vicentini, Malosio Matteo
Keywords:	SAFETY, STANDARD, COLLISION, REDUNDANT
Abstract	The new standard ISO 10218-1:2006 and 10218-2:2011 and integration are paving the way for the implementation of hybrid production systems i.e. production systems characterized by a close linkage of human and robot in cooperative production tasks. Hybrid production systems can have a big economic benefit in small and medium sized production and especially in the so called traditional sector such as the footwear sector. Among other significant aspects that should be covered to empower hybrid production systems this task will focus on the adaptation of devices and algorithms enabling the safe space-sharing between robot and the human operator addressing the issue of the on-line re-planning of the trajectory in order to achieve a safe and cost-efficient execution of the task.

Document History			
Ver.	**Date**	**Changes**	**Author**
0.1	20.2.2011	Structure of deliverable	Nicola Pedrocchi
0.2	10.3.2011	Introduction and Standards description	Nicola Pedrocchi
0.3	18.3.2011	Shoe scenario description	Nicola Pedrocchi
0.4	2.4.2011	Proposed solution description	Nicola Pedrocchi, Malosio Matteo
0.5	3.4.2011	Internal Partial Review	Lorenzo Molinari Tosattti
0.6	5.4.2011	Collision Avoidance algorithms description	Enrico Villagrossi
0.7	6.4.2011	HW/SW description	Federico Vicentini
0.8	7.4.2011	Internal Review	Enrico Villagrossi
0.9	7.4.2011	Internal Review	Lorenzo Molinari Tosattti
0.10	16.4.2011	Review	Iñaki Maurtua
0.11	16.4.2011	Review	Nicola Pedrocchi
0.12	10.5.2011	Review	Nicola Pedrocchi
1.0	18.5.2011	Submitted version	Iñaki Maurtua

Content

List of figures and tables

Figures

Tables

No se encuentran elementos de tabla de ilustraciones.

Introduction

The new standard ISO 10218-1:2006 and 10218-2:2011 are paving the way for the implementation of hybrid production systems *i.e.* production systems characterized by a close linkage of human and robot in cooperative production tasks.

Hybrid production systems can have a big economic benefit in small and medium sized production and especially in the so called traditional sector such as the footwear sector.

Among other aspects that should be covered to empower hybrid production systems this deliverable will focus on the adaptation of devices and algorithms enabling the safe space-sharing between robot and the human operator addressing the issue of the on-line re-planning of the trajectory in order to achieve a safe and cost-efficient execution of the task.

The deliverable is organized as below:

- ***Section 1*** describes what the safety principles in industrial scenarios are. The most important harmonized standards are reported and detailed. State-of-the-art of the workgroup of ISO are reported
- ***Section 2*** tries to identify what is the procedure usually adopted in order to guarantee the safety in design and build of a robotic work-cell. Aspects that should be critical in footwear scenario are identified and deeply investigated
- ***Section 3*** reports the analysis of the footwear production plant and what are the safety requirements for the application chosen in ROBOFOOT project.
- ***Section 4*** describes the framework that has been identified that should guarantee the integration of sustainable safety work cells inside footwear industries
- ***Section 5*** reports the description of the algorithms that should be allow the collision avoidance among the robot and the human operators inside the **collaborative workspace** of the robots
- ***Section 6*** reports short experimental results of the set-up realized by CNR-ITIA to proof the (i) safety-framework described in Section 4 and the (ii) collision avoidance algorithms described in Section 5

1. Safety Standards

Safety concerning Industrial Robots (IRs hereafter) is covered by several international standards such as ISO 10218. As tasks for industrial robots have gotten more complex, *e.g.* cooperation with a worker, new standards are currently being developed. New standardization efforts have also been started on service robots in order to specify general safety requirements before serial products enter the market.

The ROBOFOOT project aims to promote standardization efforts on robot safety.

1.1 Introduction

The actual approach to safety on machinery is based on ***Council Resolution of 7 May 1985***, where a first roadmap for technical harmonization and standards was delivered. The resolution fixed four basic principles in order to impose safety on machinery design:

1. legislation is limited to directives fixing “essential safety requirements” for products to have free movement throughout the Community
2. developing technical specifications is entrusted to competent organizations
3. technical specifications are not mandatory but voluntary standards
4. National authorities are obliged to recognize that products manufactured in conformity with harmonized standards are presumed to conform to the "essential requirements" established by the Directive.

This approach is extremely simple and efficient, since it allows a double approach to safety: all machinery providers have to design and build safe machines, that is, they must provide a certification that they have faced all the aspect concerning the safety of the device. This conformity can be achieved or (i) by a self-certification that all the due has been done, (ii) by the adoption of the harmonized standards. Hence, general approach to define a machine and/or a work-cell law-conformant is

a. at first a risk assessment must be performed;
b. the machine is designed and manufactured to avoid risks;
c. if risks are still present, protective devices should be foresaw (fitted on machinery or use by people);
d. introduce complementary protective measures (e.g. interlocked guards, light curtains, safety mats, hold-to-run controls, two-hand controls and enabling switches);
e. if residual risks are still present: information and education should be foresaw taking into account the capacities of users;
f. Safety must be taken into account for all the life-cycle of machine (construction, transportation, installation, put into service, use, maintenance, dismount, dispose of, recycling)
g. Safety must be documented (“technical file” or “relevant technical documentation”).

Beyond the resolution, Council delivered a “machine directive” containing the basic technical requirements needed to guarantee safety. It has been first issued as 89/392/EEC (dir.num. 392, year 1989), and it has been amended more times. The current version is the ***Directive 2006/42/EC of the European Parliament and of the Council of 17 May 2006*** on machinery, and it came into force December 29th, 2009. All machine builders have to certify and demonstrate that their products are conformant to this directive in order to obtain the CE mark. In addition, it introduces the basic principle that is forbidden to build, sell, rent, and use machinery which is not safe.

The directive aims to ensure free circulation of machinery in the European Union (no national barriers) while assuring an appropriate safety level, that is, machinery should be safe for persons, domestic animals and property.

Scope of directive is not only the machinery, but also interchangeable equipment, safety components, partly completed machinery (quasi-machines), etc. It is worth to note that usually, ***interpretation of the directive is that a general purpose robot is not a machine but the robotic cell is a machine***. That is equivalent to consider a general purpose robot as partially completed machinery.

Machinery Directive indicates what the duties in charge of the manufacturer are that guarantee the certification for the safety (and, as a consequence, obtain the CE-mark). Before placing machinery on the market and/or putting it into service, the builder must

a) Ensure that it satisfies the relevant essential health and safety requirements set out in Annex I;
b) Ensure that the technical file referred to in Annex VII, part A is available;
c) Provide, the necessary information, instructions;
d) Carry out the appropriate procedures for assessing conformity
e) Draw up the EC declaration of conformity
f) Affix the CE marking.

Another, and fundamental, instrument introduced by the directive is the use of Harmonized standards to help machine builder to be conform to the Machine Directive. Harmonized standards can be delivered by CEN and CENLEC (*Comité Européen de Normalisation, Comité Européen de Normalisation Électrotechnique*) and they are officially accepted by the European Union. It is worth to underline that CEN/CENLEC work in cooperation with national standardization bodies (UNI/CEI for Italy); members of committees are technician or teachers from appropriate areas. CEN/CENLEC works in cooperation with ISO for worldwide standards definition.

Importance of the harmonized standards is due to the fact that Machinery manufactured in conformity with a harmonized standard, shall be presumed to comply with the essential health and safety requirements covered by such a harmonized standard.

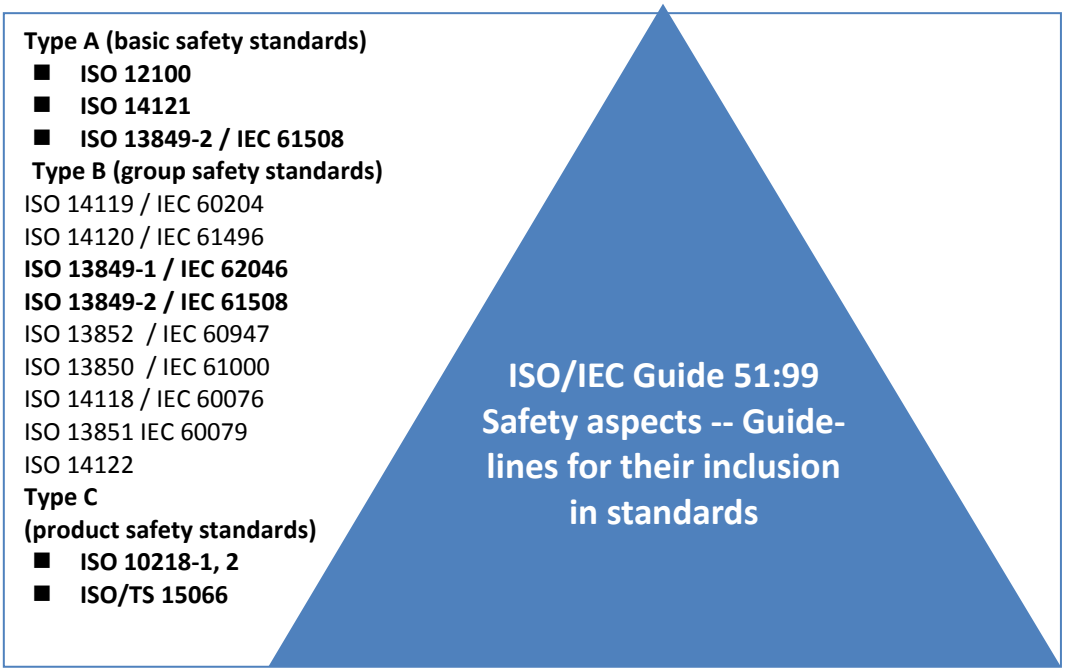


Fig. 1 Classification from [18] **ISO/IEC Guide 51**

Among the tons of standards concerning the safety, the ***ISO/IEC Guide 51*** try to identify the roadmap and a way to classify them. Standards that match with ***Type A*** of ISO/IEC Guide 51:99, mainly two are important for safety design of robot application:

1. ***ISO 12100-1:2003(E) Safety of machinery – Basic concepts, general principles for design –*** Part 1: Basic terminology, methodology and ISO 12100-2:2003(E) Safety of machinery – Basic concepts, general principles for design Part 2: Technical principles
2. ***ISO 14121-1:2007 Safety of machinery. Risk assessment. Principles.***

Another fundamental instrument in order to design safety machinery, is the adoption of Technical Specification (TS), that is a document published by ISO or IEC for which there is the future possibility of agreement on an International Standard, but for which at present the required support for approval as an International Standard cannot be obtained, there is doubt on whether consensus has been achieved, the subject matter is still under technical development, or there is another reason precluding immediate publication as an International Standard. The content of a Technical Specification, including its annexes, may include requirements. A Technical Specification is not allowed to conflict with an existing International Standard. Competing Technical Specifications on the same subject are permitted.

1.2 Functional Safety, Safety of the "partial completed machinery"

Since safety functions in modern systems are more and more frequently implemented by electronic (programmable) systems, the fundamental challenge for safety involves guaranteeing proper functionality. The functional requirements are listed in:

1. ***EN ISO13849***, Safety of machinery – Safety-related parts of control systems
2. ***EN IEC62061***, Safety of machinery – Functional safety of safety-related electrical, electronic and programmable electronic control systems
3. ***EN IEC 61508***, Functional safety of electrical, electronic and programmable electronic safety-related control systems
4. ***EN IEC61511***, Functional safety - Safety instrumented systems for the process industry sector

The importance of these standards consists on that they are intended to be a basic functional safety standards applicable to all kinds of industry. Furthermore, EN IEC 61508 and EN ISO 13849 should be considered two "equivalent" standards. Despite the differences, both aim to identify the instrument and the assessment criteria to evaluate the safety. IEC 61508 defines functional safety as:

> *"Part of the safety relating to the EUC (Equipment Under Control) and the EUC control system which depends on the correct functioning of the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities."*

IEC 62061 is the machinery specific implementation of IEC 61508. It provides requirements that are applicable to the system level design of all types of machinery safety-related electrical control systems and also for the design of non-complex subsystems or devices. In order to measure the functional safety level, the standard introduces the SIL, safety integrity. Such level is determined primarily from the assessment of three factors. Higher level safety integrity levels require greater compliance in all three areas, and so:

1. Improved reliability.
2. Failure to safety.
3. Management, systematic techniques, verification and validation.

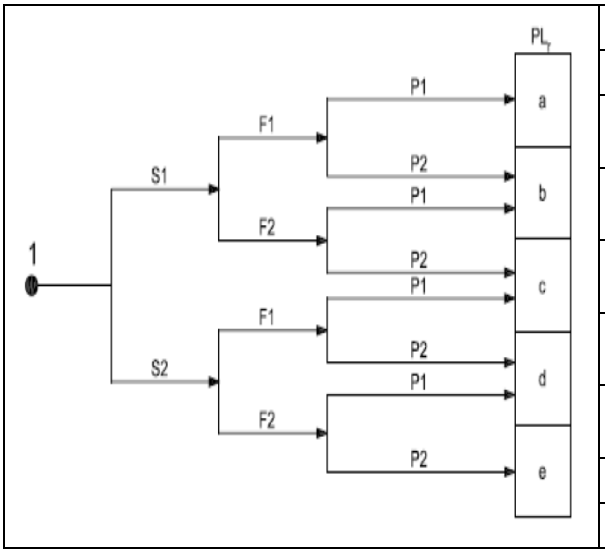
For systems that operate continuously and systems that operate more than once per year (high demand), the allowable frequency of failure must be determined. For systems that operate intermittently (less than once a year / low demand) the probability of failure is specified as the probability that the system will fail to respond on demand as:

Average probability of failure on demand	High demand or continuous mode	Probability of dangerous failure per hour
1	≥ 10−2 to < 10−1	≥ 10−6 to < 10−5
2	≥ 10−3 to < 10−2	≥ 10−7 to < 10−6
3	≥ 10−4 to < 10−3	≥ 10−8 to < 10−7
4	≥ 10−5 to < 10−4	≥ 10−9 to < 10−8

Tab. 1 SIL (safety integrity level)

Similarly, **EN ISO13849**, introduces the Performance Level scale, **PL**, with values from **a**-level to **e**-level, calculated as depicted in Tab .2

Considering **EN IEC61508**, this standard covers the complete safety life cycle, and may need interpretation to develop sector specific standards. It has its origins in the process control industry sector. The safety life cycle has 16 phases which roughly can be divided into three groups as follows: Phases 1-5 address analysis, Phases 6-13 address realization, Phases 14-16 address operation.



S severity of injury	
S1	slight (normally reversible injury)
S2	serious (normally irreversible injury or death)
F frequency and/or exposure to hazard	
F1	seldom-to-less-often and/or exposure time is short
F2	frequent-to-continuous and/or exposure time is long
P possibility of avoiding hazard or limiting harm	
P1	possible under specific conditions
P2	scarcely possible

Tab. 2 Performance Level (PL, ISO)

	EN ISO 13849-1	EN IEC 62061
Non-electronic (e.g. hydraulic)	Included	Not included
Electrical engineering (e.g. relay and/or electronics)	All architectures and up to PL e	All architectures and up to SIL 3
Complex electronics (e.g. programmable)	All architectures and up to PL e	Up to SIL 3 for according to EN IEC 61508
Application software	Up to PL e	Up to SIL 3

Tab. 3 Comparison of standards for machine manufacturing

All phases are concerned with the safety function of the system. The standard has seven parts: Parts 1-3 contain the requirements of the standard (normative) Parts 4-7 are guidelines and examples for development and thus informative.

Central to the standard are the concepts of ***risk*** and ***safety function***. The risk is a function of ***frequency*** (or likelihood) of the ***hazardous*** event and the event consequence severity. The risk is reduced to a tolerable level by applying safety functions which may consist of E/E/PES and/or other technologies. While other technologies may be employed in reducing the risk, only those safety functions relying on E/E/PES are covered by the detailed requirements of **IEC61508**.

IEC61508 has the following views on risks:

- ***Zero risk can never be reached***
- ***Safety must be considered from the beginning***
- ***Non-tolerable risks must be reduced (ALARP)***

SIL refers to a single method of reducing injury (as determined through risk analysis), not an entire system, nor an individual component.

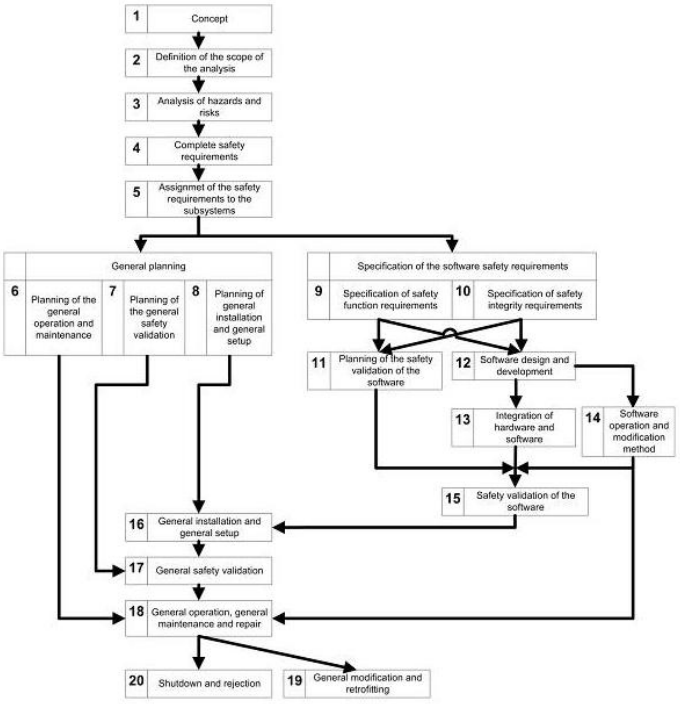


Fig. 2 Safety life cycle according to IEC 61508.

1.3 Safety of machinery

In addition to all the consideration for the partial machineries (still valid also for machines), standard **EN ISO14121-1:2007** "Safety of machinery. Risk Assessment. Principles" is the more general and important standards for the identification of the safe design guidelines for machinery and robotic cells. The manufacturer must implement a process to identify all hazards generated by the device and to estimate the according risks for each hazard. These risks have to be controlled; the results of the control have to be supervised. The process must be documented and contain the following elements:

- ***Risk analysis***
- ***Risk assessment***
- ***Risk control***

("risk", "danger", "harm" are defined in **ISO12100**, while **EN ISO14121** defines the procedure of risk management). This means that at the end of the project the risk analysis should be done by:

Risk Assessment (designer)	Risk Analysis	Definition of intended use Foreseeable misuse Hazard identification Risk estimation
	Risk Evaluation	Determine whether the tolerable risk has
Risk Reduction	Protective measures taken by the designer	Inherent safe design measures Safeguarding and complementary protective measures Information for use - at the machine (warning signs, signals, warning devices) - instruction handbook
	Protective measures taken by the user	Organization (safe working procedures, supervision, permit-to-work system) Provision and use of additional safeguards Use of personal protective equipment Training

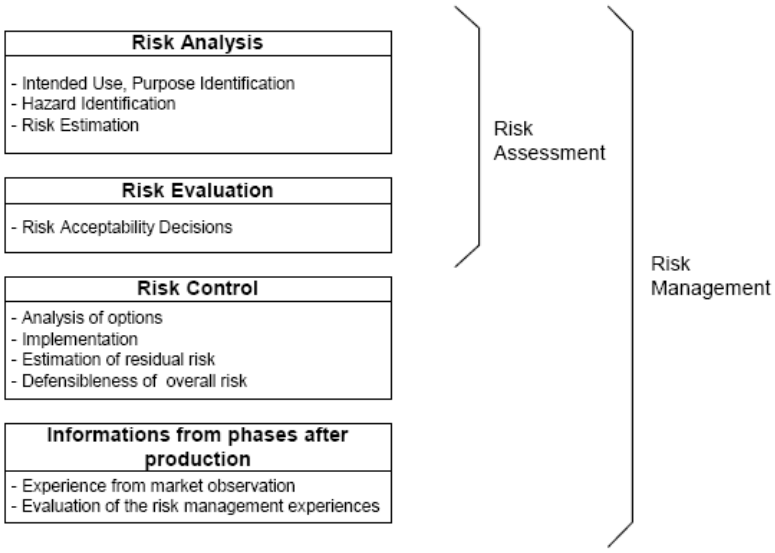


Fig. 3 The risk assessment process, part of the risk management described in EN ISO 14121

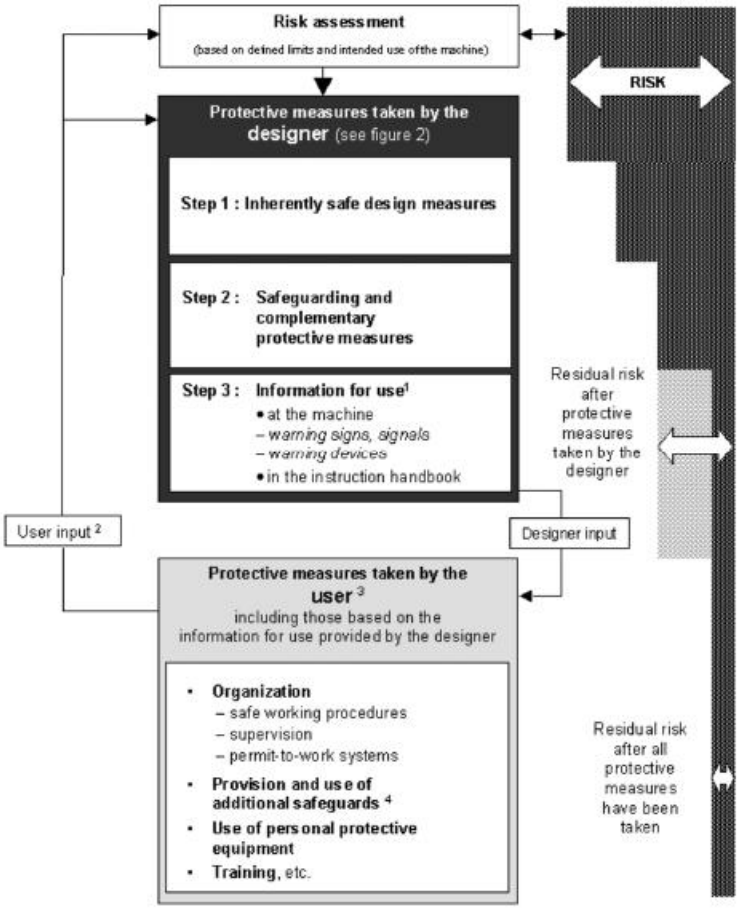


Fig. 4 Risk assessment and management in detail (ISO 14121)

1.4 Safety of Robots and Robot cells

Due its peculiarity, various standards have been delivered on robots and robot cells. Fig. 5 reports the inner organization of the ISO groups that are working on safety and robotics. Briefly:

- The WG 1 is focused on definition of the Vocabulary for safety, and its main output was the standard ISO 8373:1994;
- WG 7 deals with Personal Care Robot, and it has printed in 2011 the ISO 13842 (Non-medical personal care robot – Safety requirements);
- WG 8 deals with Service Robot, and it has not provided any relevant standard yet;
- The **WG3** deals with Industrial Robot, has in approved in the last year the ISO 10218-2 and in 2006 the 10218-1 that by now the two main important standards focused on safety of the Industrial Robots Safety. Despite of this they are still incomplete in the definition of the specification for challenging where collaboration human/robot is necessary. Since these limitations, the WG3 is working on the Technical Specification ISO TS 15066 completely focused on collaborative task. This document should be delivered by the end of the 2012.

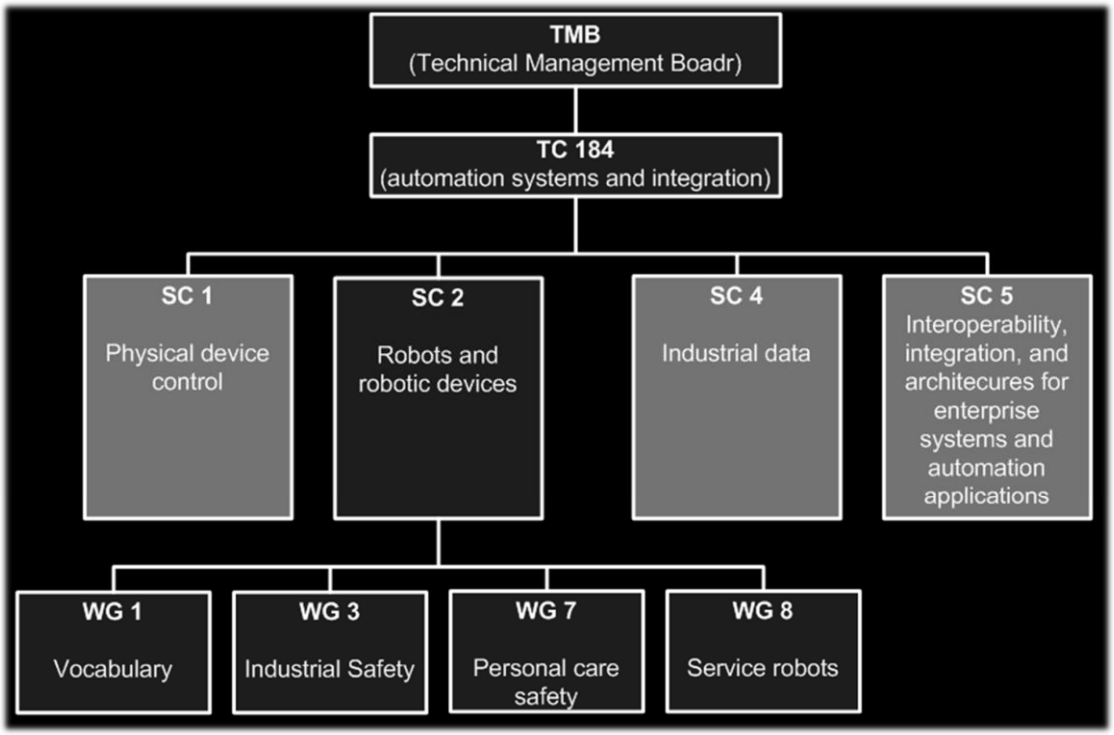


Fig. 5 Short description of group working with safety and dealing with robotics

1.5 EN-ISO-10218-1 Robot, EN-ISO10218-2 Robot system and integration

The standards **EN-ISO-10218-1/2** are quite revolutionary with respect to previous standard on robot safety. In fact they introduce application/cells with

- More than one coordinated robots under the control of one or more operators;
- Robot cooperants with human operators;
- Advanced programming tools (e.g. wireless teach pendant);
- Dynamic workspace limiting;
- Collaborative workspace.

The standard details the provisions of ***Machinery Directive*** for ***robots and robotic workcells***. Among the other issues, it describes basic hazards associate with robots and provides requirements to eliminate (or reduce) the associated risks, and it defines the requisites and the prescriptions to achieve safety (design, realize, operate, maintain, decommissioning training) for industrial robot and industrial robotic systems.

Non-industrial robots are not considered, but safety principles established in these standards should be useful for these other robots also. The most important basic principle at the basis of ISO 10218-1/2 consists on the fact that the robot and all the components must be realized in accordance with general safety principles (see appropriate standards), and the hazard identification and risk assessment are mandatory.

Beside this principle, various and important innovation are in the standard. First of all, beyond the classical **Operational Modes** (e.g., ***Automatic and Manual with reduced speed***)
the new "**Manual high speed**" operation mode is introduced. In fact, the standards introduces the idea that in some cases manual movement of the robot can be performed with speed

that may exceeds 250 mm/s but “with care” (default 250 mm/s, higher speed only for deliberate choice, high speed cannot be memorized,…).

Another important innovation consists on the fact that **also camera systems are listed among the safety devices/sensors**.

Finally, the standard introduces the definition of **collaborative workspace**, as the area within the safeguarded space where the robot and a human can perform tasks simultaneously.

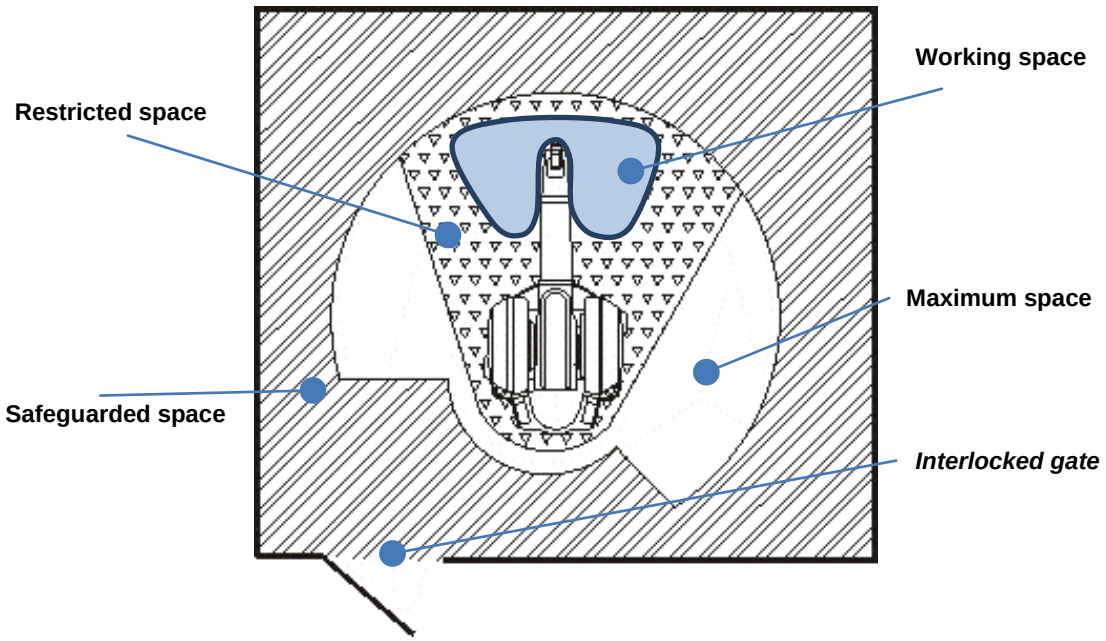


Fig. 6 Areas around robot and its definition

Despite the introduction of the collaborative workspace is to be considered a great evolution for advanced use of robots in unstructured environments, few specifications are listed in the standard, and the **Technical specification TS 15066** is under developing in order to clarify and fulfill all the possible aspect of collaborative tasks.

1.5.1 Safety-related control system performance, ISO10218-2

Full of interest is the general section concerning the safety-related control performance of the standard 10218-2. In fact, standard identify that the target application has to be designed in the whole accordance with standards **ISO13849** and/or **IEC62061** (specification of **IEC61508**). Robotics applications are recognized as machine components, and functional safety requirements are identified as the minimal-safety requirements. In addition, as for any mechanical or electronic part, all the data and criteria used in design phase have to be listed in the information for use, that is, all the procedures and choices related to the safety have to be listed and explained.

Main performance requirements are:

- ***A single fault of any part does not lead the loss of safety function***
- ***Whenever reasonably practicable, the single fault shall be detected at or before the next demand upon the safety function***
- ***When single fault occurs, the safety function is always performed and a safe state shall be maintained until the detected fault is corrected***
- ***All reasonably foreseeable faults shall be detected.***

1.5.2 Traditional versus cooperative tasks (EN-ISO10218-1)

The section describes briefly the innovation introduced by the standard with respect to the new category of the ***cooperative tasks***.

Traditionally, human operator must be at the extern of safeguarded space, and only exceptionally his presence was allowed. Furthermore, in this situation, various and strong restrictions were imposed:

- only trained operators
- ***no automatic mode***
- velocity below 250 mm/s
- holding teach pendant
- three-position enabling device
- no command from extern (except emergency stop)
- ***…supplementary measures…***

The new ***cooperative tasks*** scenario allows human operator to enter in the safeguarded space and/or interacting with manipulator if some protection measures are foresaw. Some limitations are obviously still present:

- the operator must have control over manipulator
- sensors or other means limits interacting velocity, force, energy

The cooperative tasks correspond to the Collaborative operation mode, and it is equivalent to traditional “automatic mode” and “manual mode”.

Robots designed to operate in collaborative mode with human operators in a predefined space should satisfy different requirements:

- *Collaborative mode shall be indicated in a visual way*
- *The robot **shall stop** when a human is in the **collaborative workspace** and **may resume automatic** operation when the human leaves the collaborative workspace*

  OR

- *The robot **decreases the speed** (or stops) when the operator approaches it*
- *if a “**manual guidance**” device is available, it must be located near the end-effector, it must have an emergency stop and a hold-to-run control, low speed must be selected (max 250mm/s),*

  AND

- ***the robot shall operate with a safety-rated monitored speed function active***
- ***Force and power shall be limited** by intrinsically safe design (max 80W 150N on end-effector) and limits by control units*
- *Near singularity configurations: **signaling the approaching of the singular condition and asking for confirmation** or acoustic signal and assurance of low speed*

Safety measure of the robot speed is fundamental during cooperative tasks if the robot is hand-guided, or more in general, it is moving. In addition, as underlined by the standard, “the safety-rated monitored speed limit shall be determined by the risk of assessment”.

1.5.3 Collaborative tasks (10218-2)

Special provisions regulate the access to the collaborative space, each operator or the presence monitoring system must have full control, the size of the collaborative space may be variable and controlled by sensors, Different cases are foresaw:

- ***robot stops when operator approaches***
- ***robot reduces speed when operator approaches***
- ***robot maintain a safe distance from operator***
- ***robot is moved under direct operator control (manual guidance)***

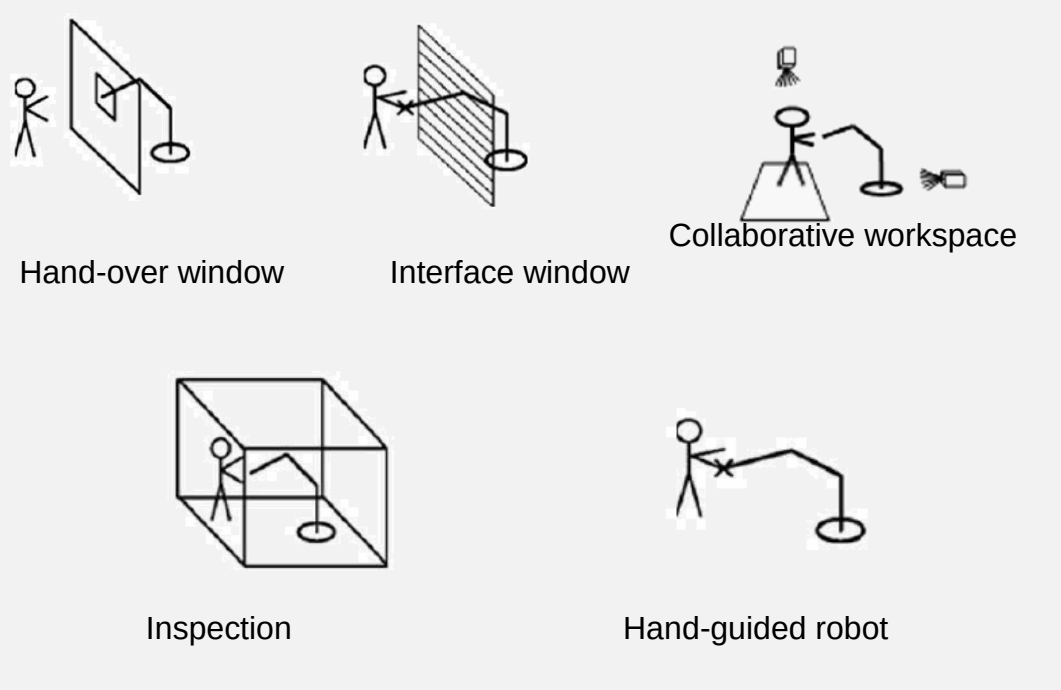


Fig. 7 Collaborative Modalities listed in the standard ISO10218-2

1.5.3.1 Classification

Hand-over window

- The robot does not exit from window
- no interruption of automatic operation during access
- limited velocity near window
- fixed or sensitive guards around window
- example: loading, unloading, testing, benching, cleaning, service

Interface window

- autonomous automatic operation within safeguarded space
- robot stops at an interface window and can then be moved manually outside the interface
- fixed or sensitive guards around the workspace
- reduced speed and reduced workspace outside and near the window
- hold-to-run control for guided movement

Collaborative workspace

- autonomous automatic operation within a common (collaborative) workspace
- robot reduces speed and/or stops when a person enters the common (collaborative) workspace
- person-detection system using one or more sensors
- reduced speed according to the distance
- robot stops safely when prohibited space accessed and possible automatic restart after clearance if properly safeguarded

Inspection

- autonomous automatic operation within safeguarded space
- a person enters the collaborative workspace while robot continues operation with reduced speed and reduced travel
- fixed or sensitive guards around the workspace
- person-detection system or enabling device
- reduced speed and reduced workspace after entering the workspace
- measures against misuse

Hand-guided robot

- application-specific workspace
- **moving by hand guiding**
- moving hand guided along a path
- reduced and safeguarded speed
- hold-to-run control
- **collaborative workspace depending on hazards of the application**
- hand guiding equipment located near end-effector

Anti-crushing distance

- Tasks requiring the use of manual **high-speed mode** shall be provided a **minimum clearance of 500 mm**.
- Specific standards regulate distances to prevent crushing **EN ISO349:2008** "***Safety of machinery - Minimum gaps to avoid crushing of parts of the human body"***.

1.5.3.2 Verification and validation

From **EN-ISO10218-1,** the **verification** and **validation** must be guaranteed by the means of different strategies:

A. Visual inspection;
B. Practical tests;
C. Measurement;
D. Observation during operation;
E. Review of application-specific schematics, circuit diagrams and design material;
F. Review of safety-related application software and/or software documentation;
G. Review of task-based risk assessment;
H. Review of layout drawings and documents;
I. Review of specifications and information for use.

Sub-clause	safety requirements and/or measure	Verification/validation
5.11.1	Information for use contains description of required safeguards and mode selection.	I
5.11.2	Integrator has conducted a risk assessment that considers the entire collaborative workspace	GHI
5.11.2	Robots in collaborative space meet the requirements of ISO 10218-1.	EF-HI
5.11.2	Protective device(s) for presence detection meet the requirements outlined in 5.2.2.	EF-I
5.11.2	Protective device(s) for presence detection meet the requirements outlined in 5.2.2.	EF-I
5.11.2	Safeguarding has been designed and installed to prevent or detect persons from advancing further into the cell (beyond the collaborative workspace).	AB-D-GH
5.11.2	Robot stops and hazards cease if intrusion into the safeguarded space beyond the collaborative workspace occurs.	B-D-H
5.11.2	Perimeter safeguarding prevents or detects persons from entering the non-collaborative safeguarded space	AB-D-H
5.11.2	Other connected machines within the collaborative workspace which have safety-related functions comply with 5.2.2 unless risk assessment deems otherwise.	EFG-I
5.11.3	Collaborative workspace where direct human robot interaction takes place is clearly defined (e.g. floor marking, signs, etc.).	A-D-H
5.11.3	Robot performance features in conjunction with protective devices comply with 5.2.2.	B-EF-I
5.11.3	If more than one person is involved in the collaborative operation, each person is protected with controls complying with 5.2.2.	AB-EFG
5.11.3	The collaborative workspace allows easy performance of tasks	AB-D-G-I
5.11.3	Location of equipment does not introduce additional hazards.	A-D-GH
5.11.3	Additional protective measures are present to prevent exposure to trapping or pinch hazards in areas where less than 500 mm clearance exists.	ABC-GHI
5.11.4	Changing from autonomous operation to collaborative and back does not endanger personnel.	AB-DEFGHI
5.11.5.1	Appropriate collaborative robot operation safety feature(s) has been selected.	G-I
5.11.5.1	Detected failure of selected safety features results in a protective stop in accordance with 5.3.8.3.	B-EFG-I
5.11.5.1	If a detected failure occurs, autonomous operation only resumes after a deliberate restart from outside the collaborative workspace.	B-DEF-H
5.11.5.2	If using safety-rated monitored stop technology, when a person enters collaborative space the robot motion stops and maintains safety-rated monitored stop.	B-DE-GH
5.11.5.3	If hand-guiding, when robot reaches the hand-over position, a safety-rated monitored stop in accordance with ISO 10218-1, is issued.	B-DEF-I
5.11.5.3	The hand-guiding device meets the requirements of ISO 10218.	B-EF-I
5.11.5.3	If hand-guiding, clear visibility of the entire collaborative workspace exists.	A-D-H
5.11.5.3	When the operator releases the hand-guiding device, a safety-rated monitored stop in accordance with ISO 10218-1, is issued.	B-DEF-I
5.11.5.4	If using speed and position monitoring technology, robots in collaborative space meet the requirements of ISO 10218-1.	EF-HI
5.11.5.4	Parameters have been determined by risk assessment and guidance provided by ISO/TS 15066.	GHI
5.11.5.5	If using power and force limiting technology, robots in collaborative space meet the requirements of ISO 10218-1.	EF-HI
5.11.5.5	Parameters have been determined by risk assessment and guidance provided by ISO/TS 15066.	GHI

1.6 Technical Specification TS 15066 (Occupational Safety)

The scope of **ISO/PDTS 15066** is "***Occupational safety requirements for collaborative robots and their work places (work environment)***". The standard title indicates the new concept of **"*Occupational Safety*"** that should be a new instrument at disposal of workcell designer.

It takes into account all the parts of an industrial robot as the end-effector, the tooling and other equipment necessary for performance of the work tasks, supplements or specifies the requirements for collaborative robot operation of **EN-ISO10218**.

As other standards, it does not apply to non-industrial robots although the safety principles may be useful also in this field, e.g. service robotics.

It is worth to underline that any quantitative numbers (e.g. force or pressure limits) stated in TS shall be considered as present state-of-the-art. They might be adjusted according to future research.

The draft resolution **ISO/TS15066** has been published in July 2010, and it includes the decision of splitting the existing work item **ISO10218-2** into a standard and a Technical Specification (ISO/TS). From the ISO Directive:

> *"When the subject in question is still under development or where for any other reason there is the future but not immediate possibility of an agreement to publish an International Standard /…/ the publication of a Technical Specification would be appropriate."*

Basic principle consists on the necessity of a safe control system which provides the relevant **safety related performance** for monitoring safety related parameters, e.g. speed, position, force etc. (robot and environment). Once this condition is satisfied, collaborative robots can be used for collaborative tasks without fixed guards.

The availability of **safe controller** should be an essential contribution to the **reduction of accidents**. Main specifications listed in the TS concern two main collaborative tasks:

- ***Hand Guided;***
- ***Safe Separation Monitoring (SSM).***

1.6.1 Hand Guided

TS15066, establishing various requisites for safety in the modality "***Hand Guided"***. However they are similar to the ones listed in **ISO1028**

Basically, robot guidance is still considered as a low-risk task if a **safe speed monitoring** is activated. Human operation acknowledgement and the safety-three-position dead man allow this task to be faced also with **actual enabling technology.**

> ***NOTE:***
>
> - *"Safe Speed monitoring" does not mean that the velocity and/or positions are measured by redundant sensors but that the actual measuring systems are certifiable as PLd/SIL3 devices.*
>
> - *Hand-guided is intended as a **collaborative task**, and its requirements it should not be guaranteed when hand-guiding devices are used for lead through programming*

If hand-guiding is used in lead-through-programming the requirements are the same as for manual teaching **(ISO10218-1):**

- ***Manual reduced-speed mode*** *meets the requirements of:*

A. Labeling**:** actuating controls shall be labeled to clearly indicate the function;
B. Speed control**:** the speed of the robot end-effector mounting flange and of the tool center point (TCP) shall be controllable at selectable speeds.
B.1 Reduced speed control operation:
when operating under reduced speed control, the speed of the TCP shall not exceed 250 mm/s. It should be possible to select speeds lower than 250 mm/s as the assigned limit.
B.2 Safety-rated reduced speed control:
when provided, safety-rated reduced speed control shall be designed and constructed in accordance with performance requirements (PLd) so that in the event of a fault, the speed of the TCP does not exceed the limit for reduced speed and a protective stop is issued when a fault occurs.
B.3 Safety-rated monitored speed:
when provided, the speed of the TCP or of an axis shall be monitored. If the speed exceeds the limit selected, a protective stop shall be issued.

Verification and Validation of ***manual reduced speed-mode*** *shall be provided by the means of* ***Visual inspection****,* ***Practical tests,*** *and* ***Review of application-specific schematics, circuit diagrams and design material***

- ***Manual reduced-speed mode allows the robot to be operated by human intervention***

Verification and Validation of ***human intervention*** *shall be guaranteed by* ***Practical tests****,* ***Observation during operation****, and* ***Review of application-specific schematics, circuit diagrams and design material****;*

- ***Manual control from inside the safeguarded space is at reduced speed with a hold-to-run control and an enabling device***

Verification and Validation ***of manual control inside the safeguarded space*** *shall be guaranteed by* ***review of task-based risk assessment.***

1.6.2 Speed and separation monitoring

TS15066, establishing various requisites for safety in the modality "***safe and separation monitoring***".

Among them, it is important the identification of how calculate the minimum separation distance, and the procedure to establishing maximum safe speed. Furthermore, various indications are listed for tracking collaborator position and velocity and the identification of potential collision. **TS15066** foreseen also that **robot controller** has to **implement methodologies to avoid potential collision**, and to notify the collaborator about the robot state (hazards, warning, etc.). Furthermore, it indicates ***the safe position and velocity monitoring of the collaborators*** as an extremely useful instrument to preserve safety.

Some examples on calculation of minimum safety distance is reported (and shown in Fig. 8).

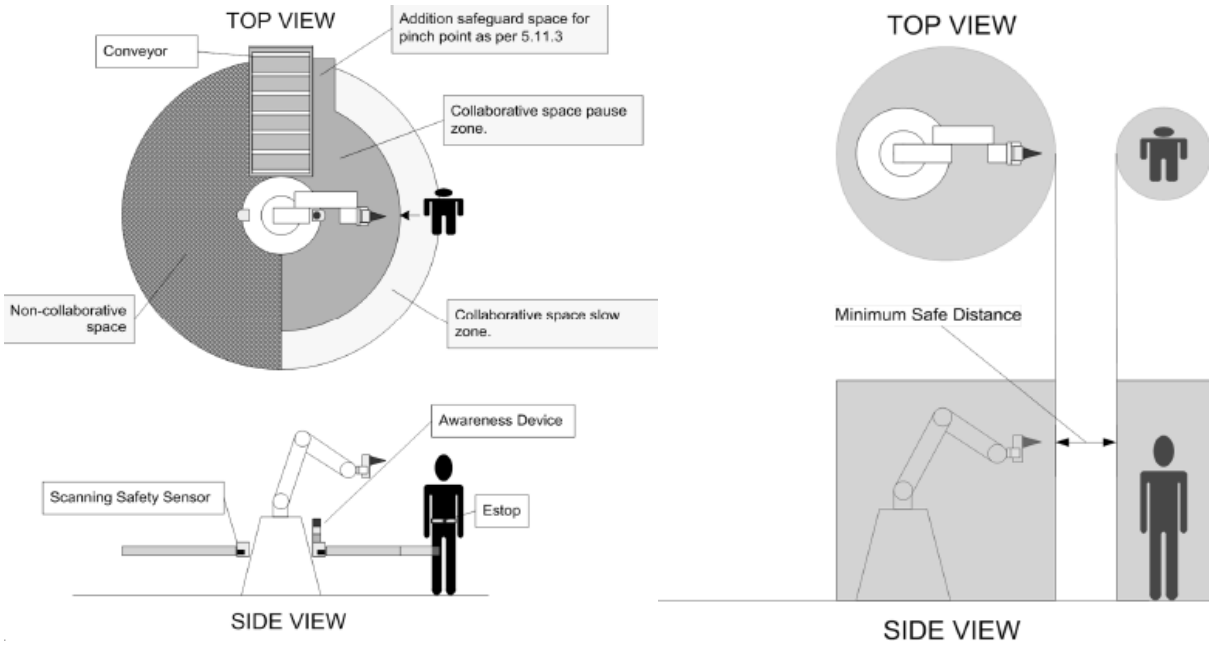


Fig. 8 TS-15066 example of identification of the minimum distance

1.6.3 Power and force limiting

Section on power and force limiting is extremely full of interest. In fact it lists various important aspects, and among them the:

- ***Technological requirements***
- ***Medical/biomechanical requirements***
- ***Ergonomic requirements***
- ***Marking and instructions***
- ***Testing and validation***
- ***Documentation of tests.***

Medical/biomechanical requirements try to investigate how to measure the risk and the potential danger for collaborator when an impact with the robot should be possible.

The **Abbreviated Injury Scale (AIS)** is included in the **TS15066** that has been introduced in 2005 from ICD-10 2006 International Statistical Classification of Diseases and Related Health Problems.

Under no circumstances a risk for injuries with higher severity than category 1 of the Abbreviated Injury Scale (AIS) and more severe than with the codifications for surface injuries of the ICD-10- 2006 can be tolerated.

Taking into account the collaborative use of robots, the injury risk for the sense organs (eyes, ears, nose and mouth) shall be lowered sufficiently through personal protective equipment (e.g. goggles).

Medical/biomechanical requirements

Body model – and individual regions with codification			Limit values of the injury severity criteria (CSF, IMF, PSP) and arranging factor CC			
Main body regions	Individual body regions		CSF [N]	IMF [N]	PSP [N/cm^2]	CC [N/mm]
1 Head with neck	1.1	Skull/Forehead	130	175	30	150
	1,2	Face	65	90	20	75
	1.3	Neck (sides)	145	190	50	50
	1.4	Neck (lateral)	35	35	10	10
2. Trunk	2.1	Back/Shoulders	210	250	70	35
	2.2	Chest	140	210	45	25
	2.3	Belly	110	160	35	10
	2,4	Pelvis	180	250	75	25
	2.5	Buttocks	210	250	80	15
3. Upper Extremities	3.1	Upper arm/Elbow joint	150	190	50	30
	3.2	Lower arm/Hand joint	160	220	50	40
	3.3	Hand/Finger	135	180	60	75
4. Lower extremities	4.1	Thigh/Knee	220	250	80	50
	4.2	Lower leg	140	170	45	60
	4.3	Feet/Toes/Joint	125	160	45	75
SF Clamping/Squeezing force, **IMF** Impact force, **PSP** Pressure/Surface pressing, **CC** Compression constant						

Abbreviated Injury Scale (AIS)

The AIS scales for particular injuries were developed by the Association for the Advancement of Automotive Medicine (AAAM). The scales are very similar to the Organ Injury Scales developed by the Organ Injury Scaling Committee of the American Association for the Surgery of Trauma [20].

AIS Code	Injury Level	Fatality Probab.	Injury
1	Minor	0%	Light brain injuries with headache, vertigo, no loss of consciousness, light cervical injuries, whiplash, abrasion, contusion
2.	Moderate	0.1-0.4%	Concussion with or without skull fracture, less than 15 minutes unconsciousness, corneal tiny cracks, detachment of retina, face or nose fracture without shifting
3	Serious	0.8-2.1%	Concussion with or without skull fracture, more than 15 minutes unconsciousness without severe neurological damages, closed and shifted or impressed skull fracture without unconsciousness or other injury indications in skull, loss of vision, shifted and/or open face bone fracture with antral or orbital implications, cervical fracture without damage of spinal cord
4.	Severe	7.9-10.6%	Closed and shifted or impressed skull fracture with severe neurological injuries.
5.	Critical	53.1-58.4%	Concussion with or without skull fracture with more than 12 hours unconsciousness with hemorrhage in skull and/or critical neurological indications
6.	Survival unsure		Death, partly or fully damage of brainstem or upper part of cervical due to pressure or disruption, Fracture and/or wrench of upper part of cervical with injuries of spinal cord

BG/BGIA risk assessment recommendations according to machinery directive [21-23]

Section Bibliography

[1] www.smerobot.org (FP6 IP CONTARACT N 011838). 2005-2009.
[2] http://www.phriends.eu (FP6 IP CONTARACT N 011838). 2005-2009.
[3] “DR4.10 Experiment assessment of collision avoidance reaction/navigation strategies by means of test beds”, project SMErobot, co-founded by the European Commission within the Sixth Framework Program; (2008)
[4] “DR4.11 Packaging of developed HW and SW solutions for delivery to demonstrators”, project SMErobot, co-founded by the European Commission within the Sixth Framework Program (2009);
[5] “DR4.14 The safe and productive robot working without fences: state of art of the developed HW and SW solutions in the whole work-package”, project SMErobot, co-founded by the European Commission within the Sixth Framework Program (2009);
[6] ISO 12100
[7] ISO 14121
[8] ISO 13849-2
[9] IEC 61508
[10] ISO 13849-1
[11] ISO 13849-2
[12] IEC 61508
[13] ISO 13852
[14] IEC 60947
[15] ISO 13850
[16] ISO 14118
[17] ISO 14122
[18] http://www.robotstandard.or.kr/, Srhim, “Presentation on ISO/TS 15066 Robots and robotic devices – Collaborative robots”, June 2011
[19] “Guida alla nuova Direttiva Macchine 2006/42/CE”, May 2008, Federmacchine
[20] http://www.trauma.org/scores/ois.html
[21] Design of workplaces with collaborative e robots. U 001/2009e October 2009 edition
[22] Institute for Occupational Safety and Health (BGIA)
[23] German Institutions for Statutory Accident Insurance

2 Actual approach to safety in automated- plant design

Nowadays, ***safe application design involves an extremely detailed safe-project***. Common approach to safe application consists on ***designing a cell with all safe-sensors*** since this allows a substantial simplification of the procedures for the certification. However, the integration of safe components on the cell layout ***imposes high cost in design, in certification and in maintenance (safe sensors usually have shorter product life)***.

The section aims to point out the standard way to adopt the norm that corresponds to the usual way adopted in order to guarantee the safety.

> ***Harmonized standards are instruments that help the designer in order to reach easily and correctly the work-cell is safe and no residual risks are still present.***

The Section 2 aims to identify the aspects that are present in the actual guidelines for the safety that should be critical is applied in the footwear scenario.

2.1 Safe design of a generic application

The paragraph introduces the necessary aspects that are mandatory to face when designing a "safe" machine.

2.1.1 Functional safety management:

Standards for functional safety in the automation industry deal with basic requirements for error prevention on: ***Product development*** and ***Application***.

Central aspect in the management and technical activities to achieve the functional safety process consists of the **Certification of the Functional Safety Management**.

A methodology used to enable the certification is the delivery of a **safety plan** covering the following points:

- Procedure and strategy for fulfilling the specified requirements
- Strategy for achieving functional safety
- Persons and departments to perform and monitor activities
- Specification of procedures and means for logging and maintaining information
- Strategy for configuration management (version control)
- Verification plan
- Validation plan

2.1.2 Risk assessment in accordance to EN-ISO14121

The manufacturer of a machine or their representatives must ensure that a **risk evaluation** is carried out in order to determine the safety and health requirements that apply to the machine.

The results of the risk evaluation must then be taken into account when designing and building the machine. The risk evaluation must be performed in a manner that enables the procedures followed and the achieved results to be documented.

The following points must be observed when performing the risk analysis:

- Identify hazards (what are the hazards, who is at risk, what types of injury)
- Evaluate hazards (probability the hazard will arise, preventing the hazard, limiting the effects of the hazard)
- Reduce hazards to a tolerable degree
- Take technical and organizational measures
- Describe residual risks

Risk evaluation is a consequence of logical steps that enable a systematic analysis and estimation of risks. Risk evaluation involves the following points:

- ***Risk analysis***: the risk analysis begins by determining the limitations of the machine while taking all phases of the machine's life cycle into consideration. After the machine limitations have been determined, then the systematic identification of foreseeable hazards, hazardous situations and hazard events is performed. Measures for eliminating hazards or minimizing risks can only be taken once the hazards have been identified. To do this, we have to look at the machine's manufacturing processes and the personnel tasks involved. Finally, all foreseeable hazards, hazardous situations and hazard events must be identified (see Machine hazards). After the risk analysis, a risk evaluation (estimation) must be carried out for each hazardous situation. The risk analysis can be performed in accordance with the following standards: **EN ISO 13849-1**, **or EN IEC 62061**, or Risk analysis according to **EN ISO 13849-1**.

- ***Risk assessment*:** after the risk analysis, a risk assessment must be carried out to decide whether a risk reduction is necessary. If risk reduction is necessary, then suitable protective measures must be selected and applied before performing the risk evaluation again. You must also check whether implementation of the new protective measures creates additional hazards or increasing other risks.

2.1.3 Determining the necessary measures

When doing this, it is necessary to consider the following priorities:

- ***Machine safety at all phases of its life cycle***
- ***Ability of the machine to fulfill its function***
- ***User-friendliness of the machine***

To ensure continuous safe operation of a machine, it is important that the protective measures allow easy use of the machine because otherwise users may attempt to circumvent the protective measures. The assumption is made that any hazards present on a machine will sooner or later lead to damages if no protective measures are implemented. The main goal is to reduce the risk as much as possible. The measures for reducing the risks are structured as a hierarchy. This is known as the "3-step method":

- ***Integrated safety,*** reachable through measures for reducing risks based on machine construction. **Safe construction** is achieved by preventing or avoiding hazards or risks through suitable selection of construction features. That is:
  - ➔ Adoption of constructive measures
  - ➔ Substitution with less dangerous materials
  - ➔ Applying ergonomic principles.

- ***Supplemental safety:*** the reduction of the risk by applying safety equipment and safety components. **Technical protective measures** must be taken in order to prevent injuries that either cannot be appropriately prevented or cannot be sufficiently limited through **safe construction**. That is:
  - ➔ Use of the *risk evaluation* as the guide for choosing exactly the right safety equipment (suitable safety equipment)
  - ➔ a fixed separator must be easy to implement and must be used where there is no need for operating personnel to access the area of danger during normal machine operation (e.g. enclosure). If frequent access is required, then alternative safety equipment must be used (e.g. light curtain).

- ***Informative safety:*** preparation of information for the user and references to residual risks. The **user information** contains communication elements such as text, words and symbols that are used separately or together. That is:
  - ➔ User information must be prepared, which informs the user about proper usage, taking all operating modes into consideration.
  - ➔ It must contain all specifications needed for safe and correct use of the machine.
  - ➔ If the use of technical and supplemental protective measures is not possible or the risk cannot be sufficiently reduced, then the user information must contain a reference to any residual risk.

Finally, **implementation of the measures** results in the creation of safety functions on the machine that must meet certain specified requirements. Technical protective measures can be divided into the following categories:

- ***Safety equipment***
- ***Programmable electrical systems***

2.2 Safety and Programmable electrical systems

Due to the increasing requirements on safety systems, the use of programmable electronic systems (safety CPUs) is the technological standard today.

All activities during the life cycle of safety-related application software must ensure that errors introduced during the software life cycle are prevented.

Standards **EN ISO 13849** and **EN IEC 62061** deal explicitly with software requirements, and it list two types of programming languages:

- **LVL** – Limited Variable Language (e.g. LD, FBD)
- **FVL** – Full Variable Language (e.g. C, C++)

LVLs & FVLs safety-related application software MUST achieve a PL level "e" if the following simplified requirements have been fulfilled:

- Development life cycle with verification and validation
- Documentation of the specification and design
- Modular and structured programming
- Functional tests
- Suitable development activities after modifications
- Operating modes, reaction times
- Suitable tools proven in use, validated function blocks
- Appropriate validation procedures
- All updates and modifications during the life cycle must be documented

Considering **FVLs**, ***further requirements must be additionally fulfilled in order to do this***.

Despite of this possibility, in industrial standard, only ***LVLs*** are used, because of the restriction imposed by **IEC61508** that is the series of standards at the basis for implementing application software in **FVL**.

NOTE:

This series of standards consists of **7 parts** with a total of approximately **430 pages**.

The installation of a standard-compliant development process in accordance to **IEC61508** represents an enormous investment. The **PL** is determined by:

- ***Features of the safety category***
- ***Quality of the components/devices (MTTFd)***
- ***Quality of error detection (DC)***
- ***Observation of common cause failures (CCF)***

2.3 Safety and Sensors

Measure of the robot, tools, objects and cooperants position in the workspace is mandatory in order to satisfy the safety requirements.

Two main measure paradigms can be identified:

- The robot that give a local information around the robot links *(local approach);*
- The cell that give an overall vision of all the workspace *(global approach)*.

Among the others, it is worth to underline two main critical aspects:

- The safety of the transducer;
- The safety of the communication channel.

In order to overcome the latter, common solutions provide safe sensors as stand-alone devices that have at disposal safe-electrical output that can activate/deactivate safety procedures.

Briefly, various safe-sensors are similar to tow-contact safe stop buttons, where the “push” is automatically performed on the basis of a detected condition.

Advanced solutions integrate safe sensors in a net through the use of safe-fieldbuses. This option allows the development of robotic-cell extremely articulated, but it increases quickly the economic costs.

2.3.1 Off-the shelves safety sensors

Various off-the-shelves sensors that are certified safe are available [2]. These devices are stand-alone solutions provided usually by low-power electrical connections that can be integrated in a robotic-cell.

As repo

ted in the table below [1], the sensors are usually costly solutions:

ID	Type	Company	Sensing Principle	Type of measure	Freq. [Hz]	Field of View	Resolu-tion	Distance	Repea-tability
1	Capacitive Sensor	KUKA	Capacitive	capacit-ance (dist)	100	180°	--	200 mm	10 mm
2	Laser scan-ner	Leuze ROTOSC AN	Laser scanner	Distance	12,5-25	190°	70 mm	4m/15m	0.5 mm
3	Laser scan-ner	Schmer-sal LS 30	Laser scanner	Distance	16	190°	70 mm	4m/49m	0.5 mm
4	Laser scan-ner	Sick 3000	Laser scanner	Distance	8-16	190°	70 mm	4m/50m	0.5 mm
5	Laser scan-ner	Sick PLS	Laser scanner	Distance	8-16	180°	30-150 mm	5.5-7m/49m	0.5 mm
6	Laser Scan-ner	Sick RLS 100	Laser scanner	Distance	3.5	300°		6m/7.5m	0.5 mm
7	Laser scan-ner	Siemens SIEGUAR D	Laser scanner	Distance	25	190°	70 mm	4m/15m	0.5 mm
8	Positioning Switch	Teleme-canique	Micro switch	Contact in the wheel	--	--	--	--	--
9	Proximity Switch	Euchner	Electrical Induction	Distance	--	90°	1-2 mm	4-5 mm	0.5 mm
10	Safe Edges	Mayser	Electrical contact	Pressure	--	300 mm	--	--	--
11	Safety Barriers	Techno GR SB4	Laser sensor	Infrared	28.5	--	35 mm	0.2-15 mm	--
12	Safety Bumper	Mayser	Electrical contact (in the bumper)	Force	--	--	--	--	--
13	Safety Bumper	SSZ Systeme Zimmer-mann GmbH	Electrical contact (in the bumper)	Force	--	--	--	--	--
14	Safety Light Grids	various	Laser sensor	Binary signals	20 -140	750 mm	14-300 mm	8m-30m	--
15	Safety Lock	Banner	Optical System	Force	--	--	--	--	--
16	Safety Lock	Schmer-sal	Mechanical contact	--	--	--	--	--	--
17	Safety Mat.	Mayser	Electrical contact (in the mat)	Force	--	--	--	--	--
18	Safety Mat.	SSZ Sicher-heits-Systeme Zimmer-mann GmbH	Electrical contact (in the mat)	Force	--	--	--	--	--
19	Safety Timer	Piltz	--	--	--	--	--	--	--
20	Safety Relay	Piltz	--	--	--	--	--	--	--
21	Safety Camera	Piltz	--	images					

[continue]

ID	Contact	Commu-nication ch.	Signal	Category	Costs (€)	Off-the-shel ves
1	Contact-less	--	DO24V	Cat. 2 DIN 954-1	> 2,000	
2	Contact-less	Infrared	DO	Type3 (EN 61496-1, 61496-3)	> 4.000	
3	Contact-less	Rs232	DO	Type3 (EN 61496-1, 61496-3)	> 4,000	

4	Contact-less	Rs232	DO	Type3 (EN 61496-1, 61496-3)	> 4,000	
5	Contact-less	Rs232	DO	Type3 (EN 61496-1, 61496-3)	> 5,0	
6	Contact-less	Rs232	DO	Type3 (EN 61496-1, 61496-3)	> 4,000	
7	Contact-less	--	DO	Cat. 3 DIN 954-1	> 4,000	
8	Mechanical-contact	--	Voltage	Cat. 1 DIN 954-1	< 100	
9	Contact	--	DO24V	Cat. 3 DIN 954-1	< 100	
10	Contact	--	DO	Cat. 4 DIN 954-1	< 500	
11	Contact-less	--	DO	Cat. 3 DIN 954-1	> 2000	
12	Contact	--	DO24V	Cat. 3 DIN 954-1	< 500	
13	Contact	--	DO24V	Cat. 3 DIN 954-1	< 500	
14	Contact-less	--	Binary	Cat. 3 DIN 954-1	> 5000	
15	Contact-less	--	DO24V	--	< 500	
16	Mechanical-contact	--	Voltage	Cat. 3 DIN 954-1	< 500	
17	Contact-less	--	DO24V	Cat. 3 DIN 954-1	< 500	
18	Contact-less	--	DO24V	Cat. 3 DIN 954-1	> 500 €/m2	
19	--	--	Voltage	Cat. 3 DIN 954-1	< 500	
20	--	--	Voltage contacts	Cat. 3 DIN 954-1	> 500	
21	--	--				

In the market various safe-sensors are already available. All this sensors matches with all the requirements listed above (**IEC61508**).

The image-acquisition and analysis technology has reached in the last year exceptional results. Two outstanding available human detection and tracking systems are ***SAFETeye©*** by Piltz [7] and high technologies sensors as Time-of-Flight camera of ***PMD Technology*** (Siegen, Germany) developed in cooperation with IPA FRAHUNHOFER [6] allow 3D identification of the human, and in generally of all the moving objects. The output of these systems consists on a cloud of points (or a close surface) corresponding to the human (or to the humans) and generally to all the moving objects.

Other sensors as Kinet® [8] allow multiple geometric descriptions that can be extracted from the acquired data by the means of parametrical surfaces, and surfaces hooked to exoskeletons. The paradigm at the basis consists of approximation of moving objects as set of moving geometrical solid.

2.3.2 Safe field-bus

Nowadays, various field-bus protocols support safety requirements that allow the integration of software solution into automated-plans. All of them refer to the concept of "***Functional safety***" (see previous paragraph, standard **IEC61508**). Most industrial networks contain some type of features to conform to functional safety requirements.

Furthermore, there are different safe-plc solutions provided by various devices suppliers for each safe-fieldbus available.

2.3.2.1 Sercos III [9]

Rather than define a unique specification for this functional safety, sercos III Safety is based upon the CIP Safety protocol developed by the Open DeviceNet Vendors Association (ODVA). This provides interoperability at the safety level with all networks based upon the Common Industry Protocol (CIP), including DeviceNet and EtherNet/IP.

CIP Safety on sercos provides for safe data transmission over sercos III up to SIL 3 (Safety Integrity Level). No additional safety bus is required, as the safety information is sent in addition to the standard data on the sercos network. With CIP Safety on sercos, data is sent on

the same medium using the same connections as standard communication. The function of the cross-media CIP Safety protocol is performed by the end units, making it possible to simultaneously operate standard and safety devices in the same network. Reliable communication can take place between all network levels, including peer-to-peer communication and cross-network communication. The master does not necessarily have to be a safety controller. It can also route data without being able to interpret it. This makes it possible for configure the safety network architecture for implementation of safety programmable controllers or peer-to-peer communication between sensors and actuators.

2.3.2.2 OpenSAFETY, Powerlink [10]

OpenSafety allows both publish/subscriber and client/server communication. Safety relevant data is transmitted via an embedded data frame inside of standard communication messages. Measures to avoid any undetected failures due to systematic or stochastic errors are an integral part of the ***security protocol***. OpenSAFETY is in conformance with **IEC61508**. The protocol fulfills the requirements of **SIL3**. **Error detection techniques have no impact on existing transport layers.**

2.3.2.3 Safety over Ethercat [11]

The protocol enhancement called *Safety over EtherCAT* enables safety-related communication and control communication on the same network. The safety protocol is based on the application layer of EtherCAT, without influencing the lower layers. It is certified according to IEC 61508 and meets the requirements of Safety Integrity Level (SIL) 3. Certified products using the Safety over EtherCAT protocol have been available since 2005.

2.3.2.4 PROFIsafe (PROFIBUS safety or PROFINET safety) [12]

This is the first open functional safety communication technology for distributed automation systems worldwide. Its specification for PROFIBUS DP and PROFIBUS PA was published first back in spring 1999. Extensions for the Ethernet based PROFINET IO followed in 2005.

PROFIsafe is designed as a separate layer on top of the fieldbus application layer and reduces the error probability of the data transmission to the level required by or better than the relevant standards. PROFIsafe messages are using the existing standard fieldbus cables in coexistence with the standard messages ("Single Channel"). PROFIsafe does not benefit from any error detection mechanisms of underlying transmission channels and thus supports the securing of whole communication paths, even backplanes inside controllers or remote I/O. PROFIsafe coined the term "Black Channel" for this concept, which now is adopted by most of the other safety fieldbusses. PROFIsafe can be used in safety applications up to Safety Integrity Level 3 (SIL) according to IEC 61508, Performance Level "e" (PL) according to ISO 13849, or Category 4 according to EN 954-1.

PROFIsafe is using expanded fault (errors and failures) detection mechanisms such as

- Consecutive numbering
- Timeout monitoring
- Source/destination authentication
- Cyclic redundancy checking (CRC)

PROFIsafe is standardized in IEC 61784-3-3.

2.4 Application-solution for safety

Finally, different patents have suggested non-standards approaches to safety for robotic cells. Below two important patens are described.

2.4.1 KUKA, Patent US7443124B2

Looking for the state-of-the-art solutions, various patents are extremely full of interest. Among the others, KUKA proposes in US7443124B2 (2008, October, 28th) an extremely interesting solution that represents the evolution of results on researches performed inside the projects [1,2] . Topic of the patent is:

> "*The invention provides a method for operating the machine, which is characterized in that at least one path section is traversed in monitored manner in a reference trip that movement-characteristic operating values are continuously measured and stored as reference values and that during machine operation said operating values are also determined and compared with the stored reference values."*

The idea consists of numerous methods and devices are known for monitoring a robot and it tries to correlate faults and errors on the information given by the robot system. The idea at the basis is summarizing as

> *"The system needs information about the environment and how it "looks" without a human presence."* [13]

The general method introduces the idea that useful information for the safety are coming from comparing actual sensing information with nominal data. This is a standard solution already applied also in vision based analysis. Hence, the system needs to know what sensors will sense when in the actual situation if there would be no human in the danger zone.

The sensor nominal data are normally dependent on the actual robot position. This is especially true if the sensor is mounted on the robot structure. But also sensors that are observing the whole workspace have to account for the actual robot position. Therefore it should be clear that for different robot positions the nominal data will also be different.

So there is the need for a subsystem that can provide the nominal sensor data for the actual robot position. Various methodologies to record the sensor data should be feasible [1,5]:

- **Supervised Reference Trajectory:** if the sensor signal is highly correlated to the robot position there will be a different nominal value for every single robot position.

- **Exploration of the environment:** If the sensor signal is coupled more loosely with the robot position it is possible to build up a complete nominal value model for the sensor data in the whole workspace. If it is possible to build intervals then the number of data to be stored is not that big. The system is not dependent on one single predetermined movement.

- **Calculating nominal data from model:** The most preferred method from our point of view is to calculate the nominal data from the work-cell model and the actual robot position. In this case the nominal value is not measured and stored beforehand. Instead from the actual position those values are calculated.

The patented idea by KUKA consists of the procedure that makes safe a robot application. Robot has to perform the task, and human operator has to

- **Test the execution without any human operator inside the collaborative workspace.**
- **Acquire the sensors information during the execution.**
- **Acknowledge the task after the test execution.**

Once acknowledge is given by the operator, the supervisor compares the data coming from the sensors and the acquired ones, and if it detects some incoherency stop safely the task execution.

This solution allows to overcome path planning errors which can occur during operation can also not be detected with existing methods.

In fact, generally only the position determination is redundantly designed, but this does not apply to the path planning.

A machine or robot system which only reliably monitors the adjacent positions can consequently not know whether the adjacent positions have been correctly planned by the control system.

Section Bibliography

[1] www.smerobot.org (FP6 IP CONTARACT N 011838). 2005-2009.
[2] http://www.phriends.eu (FP6 IP CONTARACT N 011838). 2005-2009.
[3] “TM500 Basics of Integrated Safety Technology”, AUTOMATION AT B&R, www.br-automation.com
[4] “DR4.10 Experiment assessment of collision avoidance reaction/navigation strategies by means of test beds”, project SMErobot, co-founded by the European Commission within the Sixth Framework Program; (2008)
[5] “DR4.11 Packaging of developed HW and SW solutions for delivery to demonstrators”, project SMErobot, co-founded by the European Commission within the Sixth Framework Program (2009);
[6] “DR4.14 The safe and productive robot working without fences: state of art of the developed HW and SW solutions in the whole work-package”, project SMErobot, co-founded by the European Commission within the Sixth Framework Program (2009);
[7] www.piltz.com
[8] Kinet XXXXXXX
[9] http://en.wikipedia.org/wiki/SERCOS_interface
[10] http://en.wikipedia.org/wiki/OpenSAFETY
[11] http://en.wikipedia.org/wiki/EtherCAT
[12] http://en.wikipedia.org/wiki/PROFIsafe
[13] KUKA, Patent US7443124B2
[14] ABB, Patent US7443124B2

3 Footwear production scenario

General aspects concerning working area and approaching strategies of robots in chosen footwear manufacturing applications are discussed in present section.

In the figure below the ROTTA's layout is shown. It is worth to underline that this plant is a paradigmatic solution in footwear industry.

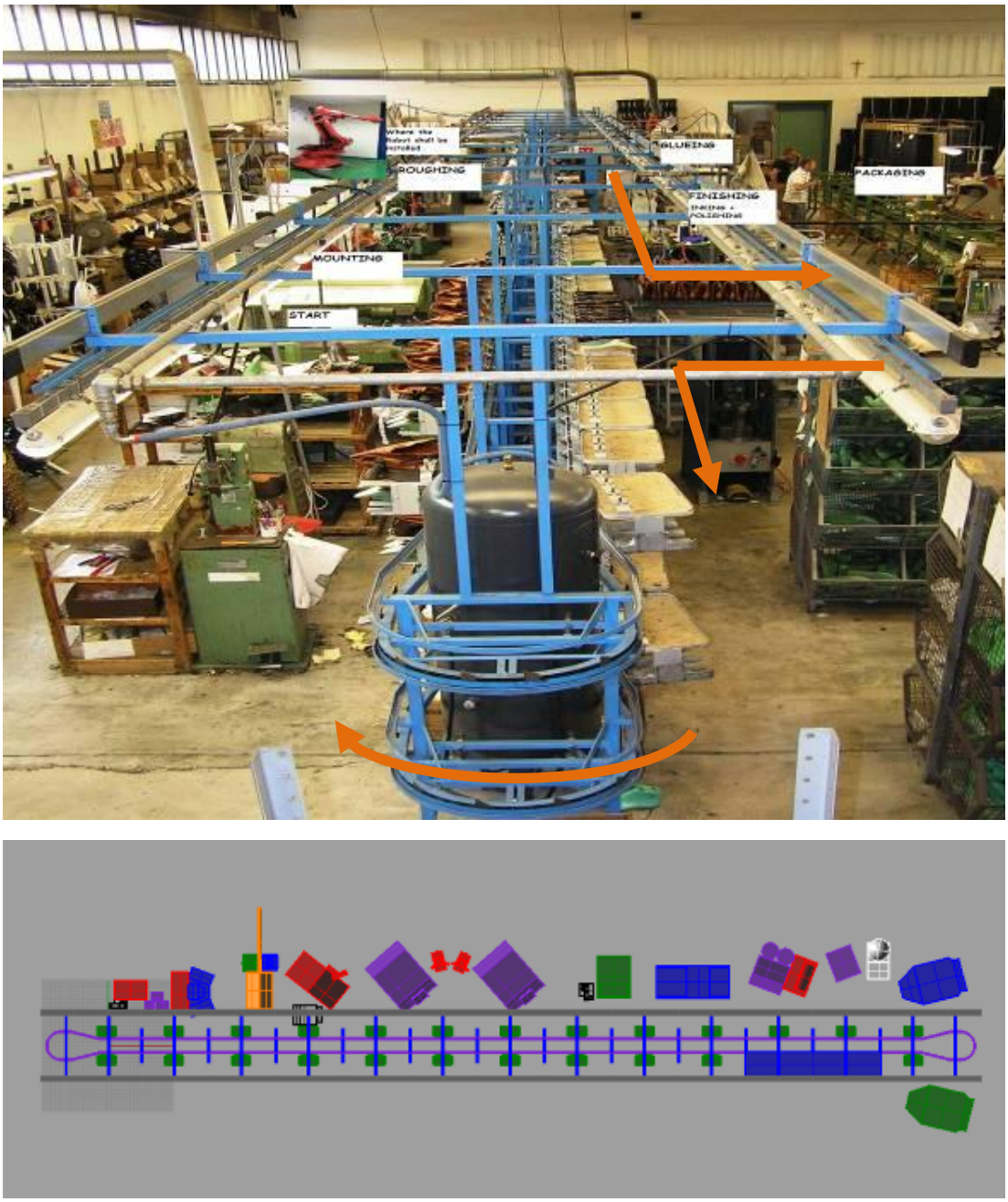


Fig. 9 Roughing and gluing operations carried out in the factory of ROTTA

3.1 Plant requirements

The introduction of robots in such kind of industry prevents the use of physical barriers in order to separate robots workspace and human operator workspace. Furthermore, human operators and robots must access to the same resources, the “manovia”, and human has to control the robot work in order to guarantee high quality for the final shoes. Briefly, the list of requirements is:

- Human must be free to access the shoe during robot operation in order to control the work progress;
- Human operator can access the manovia in the meanwhile the robot is grasping a shoe from the manovia;
- Human operator can move around the robot also when it is working;

These requisites imposes that all the workspace of the robots can be configured as **collaborative workspace**.

3.2 Operation requirements

3.2.1 Last handling (from Robofoot-D2.1)

Normally, lasts are moved along production line by means of trolleys, and are manipulated by operators (pick up, operation performing and place back).

Fig. 10 Last over trolleys in a conveyor line

In the specific cases of project end users, lines are operating having:

Conveyor line	*Rotta*	*Pikolinos*
Average number of trolleys	90	100
Average number of lasts per trolley	2 pairs	2/3 It depends, each worker works his way, one works with two shoes in the trolleys, other with three, other with four, but the important is that at the end of the manovia all shoes are put together according to its size.

Tab. 1 Conveyor line features

Technical requirements for robotized application (from Robofoot-D2.1)

Application requirements for robotized application:

1. Both Human operator and robots must be free to access to manovia
2. No physical fences are allowed

Safety requirements for robotized application

- **All the workspace** of the robot around the manovia is a **collaborative workspace**
- **Tracking of humans** around the manovia has to be **guaranteed**
- **Safe speed monitoring of the robot has to be guaranteed**
- Safety has to be designed taking into account that both **Trained** and **non-Trained operators** should be in the **collaborative workspace,**

  AND

- **Multiple non-Trained** operators should be inside the collaborative workspace

  BUT

  Only **one Trained** operator should be inside the collaborative workspace

- An **access acknowledgment request** should be foreseen in order to identify if the operator is accessing consciously the workspace;
- **Reaction strategies** performed by the robot when human operators are detected inside the collaborative workspace **should be different** w.r.t if the **human** operator is **Trained** or not.

3.2.2 Roughing, gluing, and last milling robotized cell (from D2.1)

Technical specification for robotized solution

Box dimensions, layout and reachability from the robot
The multifunctional box has a width of 1200 mm length of 370 mm and height of 850 mm. it is not necessary to physically divide the box in different areas corresponding to the different operation. The box is partially closed not to allow dust get out of it. All the tools are reachable by the robot and they are positioned to allow performing the operations.

Layout
The robot is placed on an independent platform and the tools (last milling, bottom roughing, side roughing, gluing) are placed on other independent platform.

Description of needed HW devices (equipment, …)

- A 1.5 Kw electro spindle will be used to perform the last milling operation.
- A pneumatic spindle will be used to perform he side roughing
- A 0,5 Kw electro spindle will be used to perform the bottom roughing and a gluing extruding equipment is used to perform the gluing operation.

General considerations on last milling
The last milling has anyway to be referred, in terms of “zero positioning” to the gripping block.

3.2.2.1 Roughing operation

Roughing operation consists of:

- Bottom roughing - removing the excess of (leather) material of the upper , once mounted on the last.;
- Side roughing – creating, according to a predefined pattern, an attaching area on the side of the shoe by roughing under a specific line

Process requirements for robotized solution design (from D2.1)

- The operative space is roughly limited to 1 m.
- The insertion of the robot into this area shouldn't affect the production process from the point of view of distance to the manovia and availability to the operators.
- Human operator should control the evolution of the operation in order to preserve operation quality
- The shoe is grasped from the manovia directly

Safety requirements for robotized solution design

- Operation is only partially interruptible, that is, if instantaneous robot hold is called, the roughing operation damages the shoe,

  ALTHOUGH

  when operation is hold, robot has to restart from the last position reached and a ramp-up procedure has to be designed in order to do not damage the shoe

- Collision avoidance between robot and human has to be designed in order to avoid damage of the shoe and hold the motion consequently.

  HENCE

  human cannot be allowed to be in positions that should be dangerous when robot is moving away from the shoe.

- **Clamping** of human parts **must be prevented**, and workspace must be limited to area where human cannot be clamped

- Only Trained operators can be close to the robot during operations. Acknowledgments procedures have to be implemented to allow Trained operators to be closed to the robot during the operation. Non-Trained operators should be inside the collaborative workspace of the robot in the meanwhile is working on

Fig. 11 Actual disposition of the roughing machines (from D2.1)

3.2.2.2 Gluing operation

The process is very critic from the point of view of the quality assurance of the final products. Gluing process consists of applying a layer of glue to upper as well as to sole surface.

Process requirements for robotized solution design (from D2.1)

- The operative space is roughly limited to 1 m.
- The insertion of the robot into this area shouldn't affect the production process from the point of view of distance to the manovia and availability to the operators.
- Human operator should control the evolution of the operation.
- The shoe is grasped from the manovia directly.

Safety requirements for robotized solution design (similar to roughing application)

- Operation is **interruptible only for short time-window**.
- Collision avoidance involves the hold-on of the robot.
- Humans cannot be allowed to be in positions that should be dangerous when robot is moving away from the shoe.
- **Clamping** of human parts **must be prevented**, and workspace must be limited to area where human cannot be clamped.
- Only Trained operators can be close to the robot during operations. Acknowledgments procedures have to be implemented to allow Trained operators to be closed to the robot during the operation. Non-Trained operators should be inside the collaborative work-space of the robot in the meanwhile is working on.

Fig. 12 Gluing operation and station

3.2.2.3 Last milling operation

The process is not a standard process

Process requirements for robotized solution design (from D2.1)

- The operation is performed into a closed station with physical barriers
- The operation is performed automatically when the plant is not used for shoe-process fabrication
- None human operators interactions are foreseen
- None interaction with the manovia is foreseen

Safety requirements for robotized solution design

- Operation is performed in a physical closed work-cell,

  HENCE

- standard solutions are to be provided

3.2.2.4 Robot workspace

As shown in the figure above, the work-cell is closed to the manovia but the access is open to human operators. The workspace is split into:

- ***Operation workspace:*** areas close to the different machines. In these areas the robot performs its operations, only trained humans operators can access areas; acknowledgments procedures guarantee the operator is conscientiously approaching the machine.

- ***Collaborative workspace:*** all the cell area except the operation workspace. The robot should move fast if no human operators are inside the area; The robot velocity is limited when a Trained human operator is inside the area (acknowledgments procedures guarantee the operator is conscientiously approaching the machines); The motion of a Trained human operator is tracked when he is inside the collaborative workspace;

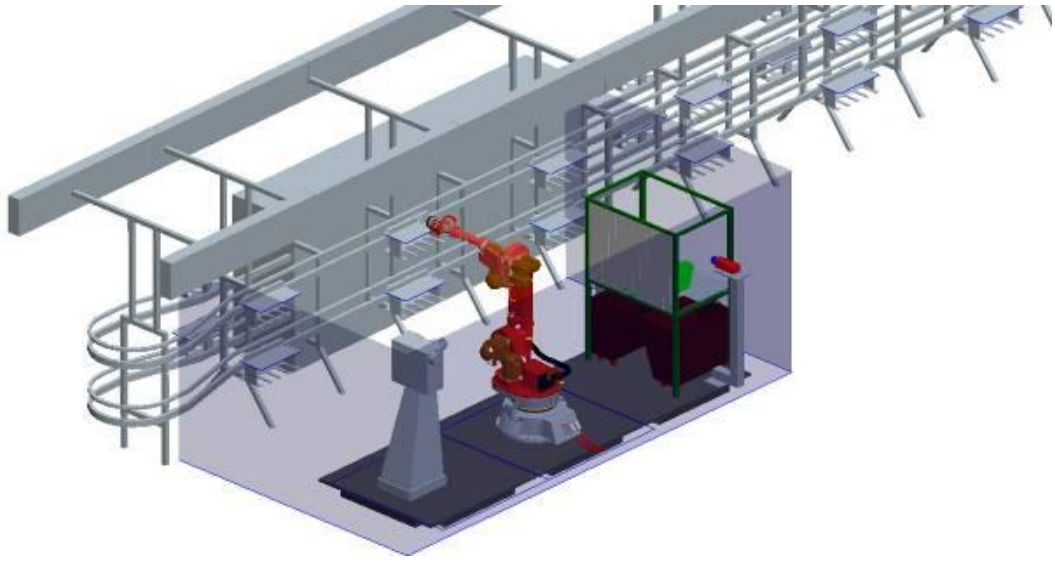


Fig. 13 Rendering of the layout.

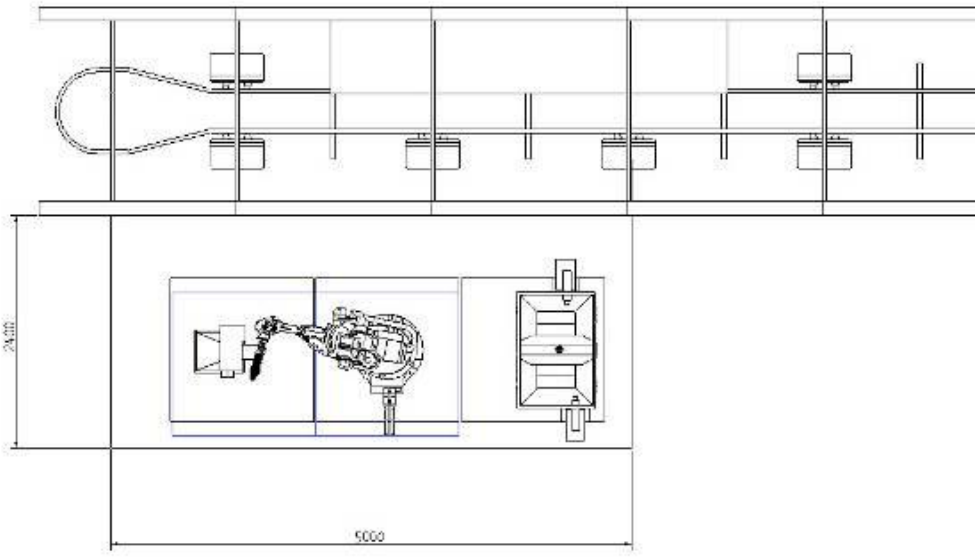


Fig. 14 Roughing, gluing, last milling station workspace (from D2.1)

The robot motion is hold when a Trained human operator is inside the area (operators that have not performed acknowledgments procedures);
Only enter/exit of non-Trained operators from the collaborative area is tracked.
The robot is moved by Trained-operators if inside the collaborative area there are both Trained and non-Trained operators

3.2.2.5 Sensors

The workspace is complex and unstructured.

Safe sensors should be useful only to identify the entering/exit of human operators from the robot workspace.

Since various objects should be inside the workspace, and, in addition, various operators should be inside the workspace, in order to track the motion of human operators various sensors should be available.

Since the high costs of safe sensors to track humans inside the workspace, **the use of low-cost redundant unsafe sensors seems the optimum solution to allow an overall control of the robot workspace.**

3.2.3 Inking, polishing and last pulling cell layout (from D2.1)

Technical specification for robotized solution (from D2.1)

- *Removal of the intermediate drying oven for inked shoes*
- *Change of the position of the cold oven.*
- *Reduction of the distance to the polishing work station*
- *Dimensioning of cabinets and devices*

3.2.3.1 Inking process description

Inking consists in spraying a chemical that is in liquid state onto the shoe. This is an operation carried out by hand by means of a pneumatic spray gun similar to those used in other industries for painting operations. The product is applied to the shoe from 25 to 35 cm. The operation is performed in a special extraction booth by a worker who holds the lasted shoe with a hand and applies the sprayed product with the other.

Process requirements for robotized solution design (from D2.1)

The dimensions of the most frequently used booths are 800x800x600 mm. These dimensions have to be considered when planning the robot automation, due to the fact that the robot arm should be placed inside the booth and be able to move so that the inking operation is performed avoiding any collision.
Inks used may be solvent-based or water-based – nitrocellulose or wax emulsions, respectively, and they are applied using nozzles with a diameter of 0.6 and 1.5 mm. Drying time for this kind of inks is from 3 to 5 minutes.

Safety requirements for robotized solution design

- Operation is interruptible only for short time-window,

  ALTHOUGH,

  when operation is hold, robot has to restart from the last position reached.

- **Clamping** of human parts **must be prevented**, and workspace must be limited to area where human cannot be clamped

- Only Trained operators can be close to the robot during operations. Acknowledgments procedures have to be implemented to allow Trained operators to be closed to the robot during the operation. Non-Trained operators should be inside the collaborative workspace of the robot in the meanwhile is working on

3.2.3.2 Polishing process description

Polishing the shoe after inking is necessary to obtain a shiny aspect. For this, the shoe is polished using some rollers made with textile materials.

Fig. 15 Polishing process description

The worker holds the shoe with both hands and buffs it, moving the shoe against the roller and applying the force that is needed to polish the whole surface.

Process requirements for robotized solution design

The polishing area is covered with a metallic plate in order to collect the buffing dust and make its extraction easy. The free space to access the polishing rollers measures about 320x400 mm. This has to be taken into account when planning the automation of this operation using robots, as the robot arm must come close to the roller and move feely avoiding any collision.

Polishing machines have two axes where different types of rollers can be assembled, depending on the type of shoes that are being manufactured. In addition, these machines are equipped with a collecting device for buffing dust that can be connected to suction means.

Polishing rollers are usually made with textile materials (cotton or wool). The dimensions of rollers are about 300 mm in diameter and 40 to 100 mm width.

Polishing rollers wear out due to continuous use, and consequently their diameter may be reduced by 100 mm after a working day. This is an important aspect to be taken into account when considering the automation of the process. That is, the tool wearing must be considered so as to correct the approach.

Furthermore, the worker who holds the shoe applies a constant force to the roller, which ensures suitable contact for polishing. This fact is also to be considered for automation, because excessive or inadequate load would affect the result.

Safety requirements for robotized solution design

SAME REQUIREMENTS OF ROUGHING

3.2.3.3 Last pulling process description

The last pulling process is comprised of two different steps:

1. Getting the hinge open
2. Removing the shoe

When both operations are carried out by hand, the last is placed face-down by inserting a rigid axis into the thimble so as to apply the necessary force. Said axis has a diameter of 10 mm and it is fixed to a structure that leans on the ground to ensure its stability.

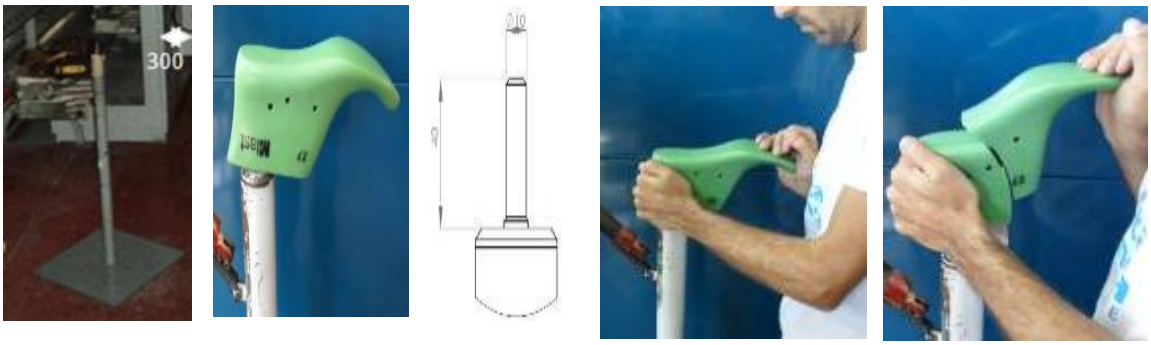


Fig. 16 Basement for last un-pulling

When the last is fixed, the forepart is levered so as to open the hinge.

Technical requirements for robotized solution

According to experimental tests with dynamometer, force needed to open the hinge was about 30 kg. This information shall be taken into account when considering the automation of the operation by means of a robot device as regards its payload.

Safety requirements for robotized solution design

1. Operation is not interruptible

2. **Clamping** of human parts **must be prevented**, and workspace must be limited to area where human cannot be clamped

3. No operators can be close to the robot during operations.

1.1.1.1 Robot workspace

As shown in the figure above, the work-cell is closed to the manovia but the access is open to human operators.

The workspace is split into:

- **Operation workspace:** areas close to the different machines.
  In these areas the robot performs its operations, only Trained humans operators can access these areas; acknowledgments procedures guarantee the operator is conscientiously approaching the machine
- **Collaborative workspace:** all the cell area except the operation workspace.
  The robot should move fast if no human operators are inside the area;
  The robot velocity is limited when a Trained human operator is inside the area (acknowledgments procedures guarantee the operator is conscientiously approaching the machines);
  The motion of a Trained human operator is tracked when he is inside the collaborative workspace;

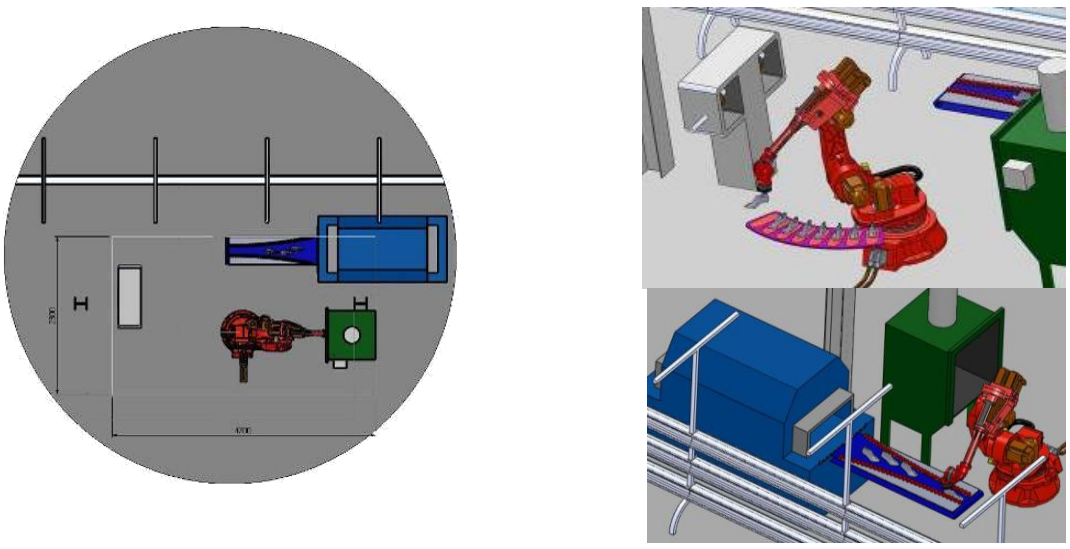


Fig. 17 Layout

3.2.3.4 Sensors

The workspace is complex and unstructured.

Safe sensors should be useful only to identify the entering/exit of human-operators from the robot workspace.

Since various objects should be inside the workspace, and, in addition, various operators should be inside the workspace, in order to track the motion of human operators various sensors should be available.

Since the high costs of safe sensors to track humans inside the workspace, **the use of low-cost redundant unsafe sensors seems the optimum solution to allow an overall control of the robot workspace.**

4 Framework for Safety in Footwear Scenario

Footwear scenario is characterized by:

- ***Unstructured and changeable working scenario;***
- ***Overlap of automatized and hand-crafted operations;***
- ***Products with low-adding value***

Therefore, safety cannot be reached by the means of high-cost safety-sensors because of they should be not sustainable solutions and, as described in Section 2, they usually are not modular, stand-alone devices.

Under these considerations, two main assumptions can be introduced:

- *Robot motion suspension has to be applied **ONLY WHEN HUMAN SAFETY IS UNDER RISK**. When humans are in collaborative workspace, robot should be able to deviate the trajectory in order to*
  (i) Avoid collision with humans and
  (ii) Without hold the movement to reach

- *More humans should be in the collaborative workspace, and the collaborative workspace is complex. Various cheap sensors have to be positioned to cover the whole robot collaborative workspace.*

The Section is organized in two main sub-sections:

- The First subsection is focused on the definition of a finite state machine able to coordinate the work-cell when the collaboration among robot and humans is allowed

- The second paragraph identifies the critical aspects related to the identification of the humans inside the workspace and it describes the solution proposed by ITIA: a safe-net composed by various unsafe sensors and provided by a safety-PLC that guarantees the coherency of all the data coming from the various transducers.

4.1 Finite state Machine for Safe Workspace Sharing

4.1.1 Introduction and state-of-the-art

In cooperative tasks, as previously described, the risk the robot can harm people is high. Even during “normal” operation the robot can seriously hurt the worker, and in a fault condition the risk is even higher. The new standard **IEC10218** proposes several mechanisms to allow the direct physical interaction of a robot and a human (Safe reduced speed, safe observed working space and the use of safe enabling switches are some of the possibilities in order to decrease the risk). Despite it is foreseen from the standard, the application-case that the robot automatically works on a task collaboratively together with a human sharing the same workspace is still problematic.

The first step in order to develop a safe and cost efficient application in industrial scenario consists on the developing of instruments that minimizing the un-operative working condition. An interesting approach to the avoidance of collision among Industrial Robots (IRs hereafter), has been given by Flordal [4]. In this case, the sequence of operations the IRs have to

perform (that correspond to a set of configuration they have to reach), is modeled as a discrete event systems (DES) and the Supervisory control theory has been used to synthesize control systems for DES that is able to schedule the sequence of movements according to the movement-constraints given by the presence of different IRs in the same workspace. This solution should be applied in interaction tasks where humans operator cover only a limited slice of the workspace, and their movements are repetitive as well as foreseeable. Unfortunately, in SMEs scenario a large number of applications do not respect the above assumption; thus, safe interaction through planning and control becomes very important.

A critical issue consists on the fact that despite the collision avoidance problem deals with motion planning and the control of the motion [14], they advance mainly independently of each other [5]. In addition, safe collision avoidance through planning is mainly focused on navigation and on control of redundant manipulators in cluttered environments [31], [32]. In [29] Perry claims (i) collision prevention and configuration optimization to avoid obstacles being the only choice for non-redundant manipulators and (ii) planning approaches are well suited only for achieving a goal position in known static environments. A quite ancient but still actual survey on the major approaches to obstacle collision avoidance is Hwang [12] where four major approaches to motion planning are listed:

- **Skeletons** impose that feasible motions are mapped onto a network of one-dimensional lines; motion planning problem becomes a graph-searching problem [17]–[19];
- **Cell Decomposition** imposes the workspace is decomposed into simple cells with known adjacency relationships (The path from start to goal is then formed using a sequence of connected cells [20], [21]);
- **Potential Fields**, where a potential is constructed throughout the workspace (The robot then seeks the point of lowest potential [22], [23]);
- **Mathematical Programming**, where obstacle avoidance is considered as an optimization problem is solved to find a path from start to goal (Inequality constraints are used to eliminate forbidden regions [24]).

Harden in [13] underlies as each of the above methods relies on determining distances between a manipulator and other objects in its environment. Despite distances are determined directly from sensor measurements (lasers, sonar, etc.), they must be calculated based on a model of the environment.

Among the others, Artificial Potential Field [22] is one of the most historically important and feasible for IRs. The idea is that obstacle and target are generator of artificial potential fields that defines the complete motion of the robot. Reactive motion behavior in dynamic and unstructured environments is based on real-time environment knowledge acquisition based on local information by given by suitable sensor. Unfortunately, as in [30] this approach does not guarantee stability in target reaching and the behavior of the robot is unpredictable. A further strategy allowing the real-time correction of the trajectory is proposed by Reif [33].

A step toward chasm the gap between planning and control is the probabilistic methods in motion planning [18] that can be applied to problems of high complexity. Also the elastic strip framework [16], [27] allows the integration of task-oriented dynamic control and motion coordination (that is global motion planning methods [15] and reactive real-time obstacle avoidance [22]). In this framework, tasks can be specified at the object level, leaving redundant degrees of freedom of the robot unspecified. Using those redundant degrees of freedom, elastic strips allow the integration of motion behavior in addition to task execution, such as obstacle avoidance or posture control. These behaviors can be controlled and changed reactively in real time without violating constraints imposed by the task. Thus, elastic strips provide a powerful approach to motion generation and execution, in particular for robots with complex kinematic structure operating in unstructured and dynamic environments. In last years, new approaches have been suggested, always for safe collision avoidance in naviga-

tion scenario. Kulic and Croft [11] proposed the use of a danger index, D_I, formulated as product of distance factor, f_D , velocity factor, f_V , and inertia factor f_I

$$D_I = f_D f_V f_I :$$

They use it as input for real-time trajectory generation when the index exceeds a pre-defined threshold. The danger index is used to generate a repulsive force similar to artificial potential force proposed by Khatib [22] and move the robot to a safer place in case of danger. The human was considered an obstacle and maximum effort was devoted to avoid it or to stop the robot if there is no way to avoid. Furthermore, in [8] Kulic establishes a cost function consisting of the sum of goal seeking criterion, obstacle avoidance criterion, and danger criterion. The planned path is generated by searching for a set of configurations that minimized the cost function. Liu [9] proposed an interaction strategy with six kinds of planning actions to keep a safe distance and predict collisions in dynamic environment, and the main contribution claimed in this paper is the rapid mapping of a moving obstacle into invalid and dangerous edges in the roadmap. Heinzmann [7] proposed an impact potential control scheme that checks the nominal torque generated by trajectory generator for a safety envelope, that is nominal torque generated by trajectory generator are checked for the safety envelope and clips it if it is outside that envelope. Wosch et al. [10] considered the human-machine interaction scenario in dynamic environments with moderate complexity and proposed an integrated control architecture combining planning and reactive components. They presented a motion planner interacting with reactive plan execution system to avoid obstacles.

For sake of clarity, let us introduce the terminology used concerning the description of the movement of the TCP of the robot in a complex environment.

Definition 1 (Movement terminology):

The motion of the TCP in the space is described by the Path, that is set (continuous or discrete) of the geometrical points that the robot has to follow; by the Motion Law, that is the velocity profile the tool of the robot has to follow along the path, i.e. in the follow the motion law is considered parameterized on the abscissa curvilinear; by the Trajectory that is the composition of motion law and the path.

Definition 2 (Obstacles typologies):

The obstacles the robot has to avoid can be distinguished in Static obstacles, that are objects are already present in the programming phase (desks, boxes, machines, cables, etc.) and they do not change their position during the task execution; Dynamic obstacles, that are humans operators, and/or moveable objects moving through the robot workspace that are already foreseen in the programming phase and they are necessary for the task execution; Unforeseen obstacles, that are humans operators, and/or moveable objects moving through the robot workspace that are completely unknown in design phase.

4.1.2 High Level Finite State Machine Description

In order to integrate collision avoidance strategies in a robot-cell guaranteeing the safety levels imposed by the standards, a model of the interaction of the robots and the human operators is needed.

The fundamental aspect is the definition of the different robot-“behaviors” corresponding at the different situations that should happen when humans are in the **collaborative workspace.**

A feasible representation of the different robot behaviors can be achieved through the adoption of a hierarchical finite state machine (FMS) which is made up by three **superstates** called respectively Safe area, Warning area, and Stop Area:

- Human is in a **SAFE AREA** with respect to the robot: the distance between the operator/object and the robot is greater than an imposed limit. Considering the robot and the human velocity the SAFE distance is, at least, the one which doesn't allow a contact between the robot and the operator;

- Human is in a **WARNING AREA** with respect to the robot: the robot could go in contact with human but the relative distance allows the execution of avoidance control algorithms;

- Human is in a **DANGER AREA** configuration with respect to the robot: the distance it is less than the minimum allowed, the risk of collision is high.

During the normal way to work of the robot enclosed by fences, the active state is always the ***Safe Area***. Although in ***Safe Area*** path re-planning should be needed, for instance to simplify offline programming due to layout changes or cooperation of different robots, the trajectory planning is usually done by the robot controller in a standard way. On the other hand, if an intrusion is detected by the safety sensors, the FMS has a state transition towards superstate ***Warning Area***. Thus another FMS is executed. Such an FMS has two states. The default one is associated with the action of speed reduction. Specifically, the robot slows down its speed to reach the one imposed by the ISO norm during programming (250mm/s) when an interaction with the human operator is allowed. Then the controller can activate one of two possible states according with the active or passive collision avoidance strategy that has been implemented. The identification of different states for the human-robot interaction is not enough because it is necessary to take into account the typology of the applications they have to perform.

In addition, each application requires a different approach. As detailed in [5], the process failure of the task is obviously a mandatory requirement in order to a correct definition of the collision avoidance strategies and algorithm. Furthermore, concerning the fences, their removal is not feasible for all those processes where dangerous tools, material projection (sharp chips or sparks), unsafe temperatures or other dangerous environment conditions are present.

Four different families of tasks may be identified:

- The task **does not allow modifying the path or the velocity** of its execution since this produces a process failure and the working object corruption;

- The task **does not allow modifying the path but allows to slow down** or interrupt its execution without compromising the final result;

- The task **allows modifying the path but it does not allow the modification of cycle time**;

- The task **allows modifying both the path and the velocity (cycle time).**

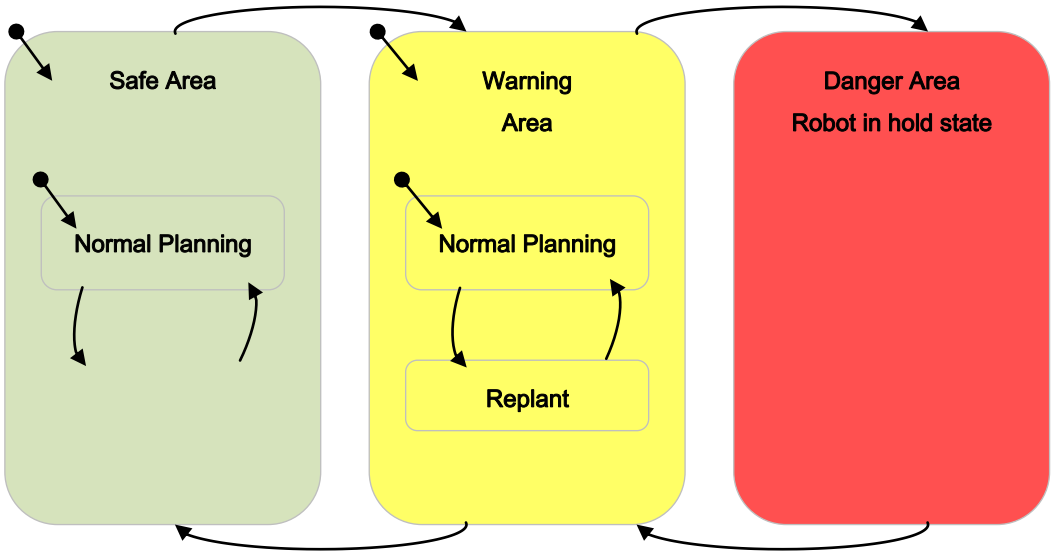


Fig. 18 Collision Avoidance in cooperative space, Finite State Machine

4.2 How Guarantee Safety in Collaborative Workspace

4.2.1 Measure of operator position

The position of the obstacles inside the collaborative workspace can be identified by using sensors fixed to:

- The robot, giving a local information around the robot links *(**local approach**);*
- The cell, giving an overall vision of all the workspace ***(global approach)***.

Nevertheless, the availability of **various safe-sensors** that allow **detection** and **tracking** of human-operators within the collaborative workspace introduces the problem connected to the sustainability costs of the plant. In addition, there is still a trade-off between safety and performance, meaning that safe systems often show a lower performance, and high performance systems often show a lower safety.

A good solution for this trade-off is to combine both approaches; hence, **integration** in the same set-up of **safe-sensors and unsafe-sensors** seems to be the only solution available in order to guarantee the correct detection and tracking of human within the collaborative workspace of the robot.

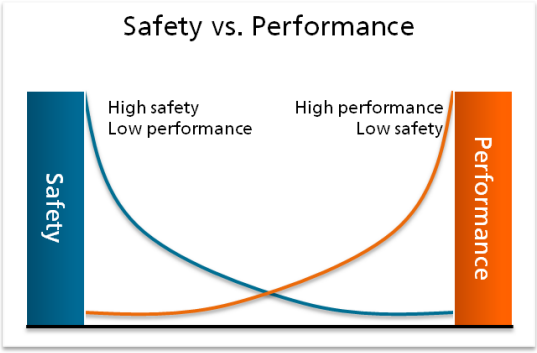


Fig. 19 Trade-off: safety vs. performance

The main technological challenge of this design choice is the integration of several components into an overall safe system resulting in the development of algorithms for the data processing of the safe and unsafe sensor systems that enable a high performance solution.

Safety and performance are very important in order to make human robot cooperation applicable. In turn, human robot cooperation is important in order to meet the demands of the European manufacturing for high flexible and adaptable technologies. The advantages of this solution are valuable:

- The system enables safe and high performing human-robot cooperation
- The system applies the correct solution for the correct work (high-cost and safe sensors only when and where they are mandatory)
- The results should be usable also in industrial scenarios different from footwear
- The system enables the design of the safety on the work-cell design

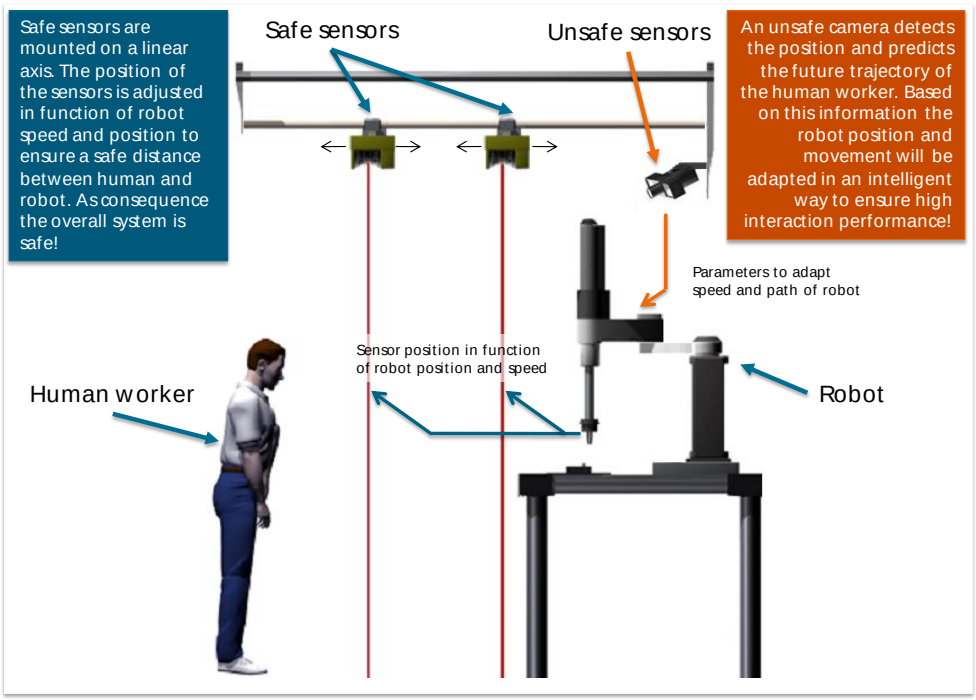


Fig. 20 Approach to overcome safety vs. performance trade-off

Hence, adoption of many and various sensors is needed, and sensor fusion algorithms should be necessary. Centralized fusion algorithms, as kalman filter or bayesian networks (the clients simply forward all of the data to a central location, and some entity at the central location is responsible for correlating and fusing the data) seems the best choice in order to:

- Allow an easy integration in high-automated plant where sensors are usually analyzed by a single PC
- Allow an easier approach to risk analysis.

In fact, decentralized fusion algorithms (where the clients take full responsibility for fusing the data) introduce the problem of a decentralized safe-control, resulting in a complex framework with respect to safety requirements.

The use of various sensors introduces the problem of the management of data described in different frames.

An extremely useful instrument is the adoption of the ***TransducerML*** (Transducer Markup Language) as common language for the data communication among PC and different sensors. *TML* captures when and where a sensor measurement or transmitter actuation occurs. Its system description describes not only individual data sources but also systems of components, including the specific types of components, the logical and physical relationships between the components, and the data produced or consumed by each of the components.

Metadata relating to archiving, indexing and cataloguing is an integral part of TML, since a TML data stream is designed to be self-contained and self-sufficient. Any information about the system, as well as information required to later parse and process the data, are captured in the TML system description. In addition to information about the system that produced the data, precise information about the data itself is captured. Data types, data sizes, ordering and arrangement, calibration information, units of measurement, precise time-tagging of individual groups of data, information about uncertainty, coordinate reference frames (where applicable) and physical phenomena relating to the data are among the details which are captured and retained. The TML system description therefore automatically tags all fields, which can later be stored in a registry for discovery.

A key benefit to TML is that by bringing both data and metadata from multiple time-varying sources of data into a single stream in a common format, data and metadata archiving, retrieval, analysis and processing can be more easily performed across disparate hardware and software systems. The time tagging of both the data and metadata allows precise determination of the state of a system, and therefore whether its data is of interest, regardless of whether that system remains static or has elements removed, replaced or added. This permits searching for data at a finer granularity than previously possible, while still supporting higher-level data discovery if a user so desires, since the use of individual fields within a TML system description is optional.

Adoption of such kind of common structures for the data description allows also easier calibration procedures. Calibration procedures that allow the integration of various sensors have to be designed specifically for each robotic-cell.

In unstructured environments like the footwear scenario, calibration procedures need of a common reference. Use of robot end-effector as calibration reference should be a good solution in order to minimize cost and reach good accuracy.

4.2.2 ITIA's solution: Safe robotic cell as Safe-net of unsafe devices

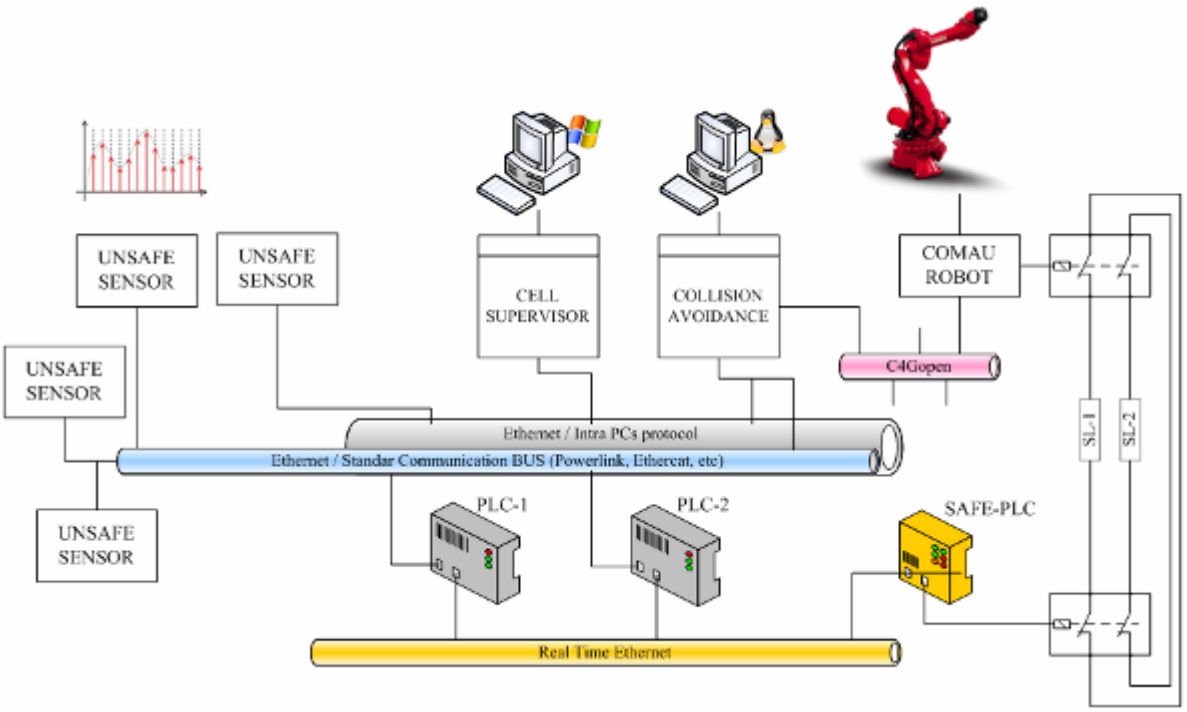


Fig. 21 ROBOFOOT safety architecture

Safety issues are almost entirely related to the risk assessment and the prevention of potential wrong robot motions, including positions, velocities and accelerations.

Two basic features have to be addressed in footwear scenario:

- Since the robot-user workspace is shared, the architecture has to avoid all possible harm for humans workers;
- Since the robot has to perform technological operations (i.e., forces exchanged with between tools and end-effector), the architecture has to avoid the robot exceed forces that could damage the manipulator itself, the tools and the shoes;

Intrinsically safe robot devices are only passive machines and/or light weight compliant machines, and they may not require dedicated solutions because of the limited torques generated by motor joints and the low residual inertia in case of axes runaway.

Unfortunately, footwear industry requires high power machines that usually do not allow a hardware limitation of the power that can be exerted.

Furthermore, in various operations the robots need high power, as in milling and high-speed roughing.

In the case of fairly more massive robots, which are able to provide higher speeds and forces, additional methods and devices, are instead vital for automatically preventing that any source of failure hinders a safe interaction with any user.

The safety architecture includes three layers of robot monitoring at runtime, all of which end up into a controlled stop and a power cutting (represented with the double switch on the double power lines in previous scheme):

i. ***The first layer deals with the safety of the planning and the execution of the trajectory.***

ii. ***The second layer deals with the coherence of the data coming from redundant measurement systems;***

iii. ***The third layer consists on the collision avoidance among the robot and the human workers, and it implements the algorithms to guarantee the safety in the operations.***

The first layer is in charge of the robot controller, whereas the second and third layers are instead provided by a set of double Programmable Language Computers (PLC) and an industrial Safe PLC. These devices are interconnected over a real time signal bus, and a motion detection sensor able to capture both the robot movements and track the position of the user close to the robot workspace. Feasible implementation should involve:

- set of cameras for markers tracking and/or Time of flight camera in the case the accurate tracking of the user movements is mandatory (e.g., user can cooperate with the robot during operations)
- set of laser scanning devices and low cost in the case only raw-information on the user position are needed, that is, the position of the user is described by algorithm based on fuzzy logic
- an Inertial Measurement Unit (IMU) with 3D acceleration and 3D rate sampling to duplicate the position of the robot

In the case of tracking with cameras, a cluster of markers is placed on a relevant robot link (usually close to the wrist). In the case of the IMU, the device is quite compact and is attached to the robot wrist as well. Both systems allow the full kinematics computation, either for derivation or integration of the sampled signals (no details of computational issues are given here). These devices are used as an additional source of sampling for the same trajectory provided by the robot joint sensors.

4.2.2.1 First Safety Layer

The first layer consists on the verification that the robot works correctly, that is:

What has been programmed is what the robot is executing.

The robot controller is in charge of this safety check performing a number of watchdog checks over the joints rates and end-effector position errors. These features are already included in industrial robot.

In addition, industrial controllers should be programmed also in order to verify that programmed forces are coherent with the real execution forces, and in case of differences they should stop the execution as when a following error is raised by the motion interpolator.

In fact, this condition should be used in order to detect collision with human and/or objects in the environment. COMAU controller already implemented this feature for robot with high payload in packaging industries. It would be challenging the integration of these algorithms also for low-payload robots.

This layer has to be implemented taking into account that standard robot controllers are not certifiable, since they do not provide intrinsic redundant calculation. To overcome this technical limitation of standard robot controllers, different solutions can be investigated:

- Modification of industrial controller to address the calculus redundancy and the measure redundancy.
- The planned path has to be verified and an acknowledgment procedure has to be executed by human worker. During the verification phase, the movement has to be stored in another controller, and once the path has been acknowledged, redundant measure systems have to be provided in order to verify that the real movement is close to the acknowledged one (as detailed in the follow sub-section).

4.2.2.2 Second Safety Layer

The second layer consists on the tracking of the robot through external sensors, that is:

Double redundancy of kinematic measurement must be guaranteed .

The same trajectory is sampled from both the robot and the motion sensor and, additionally, the sampling synchronously redoubled by the two PLC.

Double redundant signals are then checked for matching against a given accuracy threshold ε_S that depends on:

- The errors due to the calibration inaccuracy ε_c, which yield a misalignment between the sensor and robot coordinate frames,
- The undetected compliance of robot mechanics ε_r, which causes the real position of the links carrying the sensor to be different from the nominal one provided by the joints sensors,
- The latency of transmission of signals eventually carried by asynchronous buses ε_t , which causes a time shift of actual values sampled
- The intrinsic accuracy of each sensor $\varepsilon_{Senosrs}$

A way to impose the safety stop consists on comparing the computed trajectories from the robot controller and movement measured from the redundant sensors and verifies that the difference is more than a proper threshold. This safety layer helps in ensuring that the robot is where it was supposed to be.

4.2.2.3 The third layer

The third layer provides the verification of robot positioning with respect to the human workers that is:

Computation of virtual walls around the human operator

The virtual safe volume is of course updated at runtime and displays the minimum thickness around the human worker.

This layer introduces the correct modeling of the humans as a set of rigid volumes connected to an exoskeleton.

Despite it is still a challenging topic in industry, nowadays this problem should be considered practically solved from a theoretical point of view, as some commercial devices display (Microsoft-Kinect® among others products).

In authors' opinion safe devices or low-cost devices easy to integrate in automated plants will be soon available when their utility and profitability will be proven by industry-applications.

Section Bibliography

[1] www.smerobot.org (FP6 IP CONTARACT N 011838). 2005-2009.

[2] http://www.phriends.eu (FP6 IP CONTARACT N 011838). 2005-2009.

[3] R. Alami, A. Albu-Schaeffer, and al., "Safe and dependable physical human-robot interaction in anthropic domains: state of the art and challenges Workshop on Safe and Dependable Physical Human-Robot Interaction" in *Anthropic Domains,* Beijing, China, 10 (2006).

[4] Hugo Flordal. Compositional *"Approaches in Supervisory Control with Application to Automatic Generation of Robot Interlocking Policies"*, Department of Signals and Systems Chalmers University of Technology, Goteborg, Sweden

[5] Oliver Brock and Oussama Khatib; "Elastic Strips: A Framework for Motion Generation in Human Environments", *The International Journal of Robotics Research*, **21**:12 (2002)

[6] K. Ikuta, H. Ishii and M. Nokata; "Safety evaluation method of design and control for human-care robots", *The Intl. J. of Robotics Research* 22 (5) (2003) 281-297.

[7] J. Heinzmann and A. Zelinsky, "Quantitative safety guarantees for physical human-robot interaction", *The Intl. J. of Robotics Research* 22 (7-8) (2003) 479-504.

[8] D. Kulic and E. A. Croft, "Safe planning for human-robot interaction", *J. of Robotic Systems* 22 (7)(2005) 383-396.

[9] H. Liu, X. Deng and H. Zha, *A planning method for safe interaction between human arms and robot manipulators*, Proc. of IEEE/RSJ 2005 Intl. Conf.Intelligent Robots and System, Alberta, Canada, (2005) 2724- 2730.

[10] T. Wosch, W. Neubauer, G. V. Wichert and Z. Kemeny, Robot motion control for assistance tasks, Proc. of 11th IEEE Intl. Workshop on Robot and Human Interactive Communication, (2002) 524- 529.

[11] D. Kulic and E. A. Croft, Real-time safety for human-robot interaction, Robotics and Autonomous Systems 54 (2006) 1-12.

[12] Hwang, Y.K., and Ahuja, N., 1992, Gross Motion PlanningA Survey, ACM Computing Surveys, V24, No. 3, pp. 219-291.

[13] T. Harden, Kapoor C., and Tesar D. Obstacle Avoidance Influence Coefficients For Manipulator Motion Planning, Proceedings of IDETC/CIE 2005 ASME 2005 Design Engineering Technical Conf. September 24-28, 2005, Long Beach, California, USA

[14] La Valle. Planning Algorithms. University of Illinois, 2005.

[15] J.-C. Latombe, Robot Motion Planning. Kluwer Academic Publishers, Boston, 1991.

[16] Brock, O., and Khatib, O. 1997. Elastic strips: Real-time path modification for mobile manipulation. In Proceedings of the International Symposium of Robotics Research, pp. 117122.

[17] Canny, J.F., 1988, The Complexity of Robot Motion Planning, The MIT Press, Cambridge, Mass.

[18] Kavraki, L. E., Svestka, P., Latombe, J.-C., and Overmars, and M. H. 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation 12(4):566580.

[19] Svestka, P., and Overmars, M.H., 1998, Probabilistic Path Planning, Robot Motion Planning and Control, J.P. Laumond, Editor, Springer-Verlag, pp. 255-304.

[20] Schwartz, J.T., and Sharir, M., 1983, On the "Piano Movers" Problem. II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds, Advances in Applied Mathematics, V4, pp. 298-351.

[21] Lozano-Perez, T., 1987, A Simple Motion-Planning Algorithm for General Robot Manipulators, IEEE Journal of Robotics and Automation, Vol. RA-3, No. 3 (June), pp. 224-238.

[22] O. Khatib, Real-time Obstacle avoidance for robot manipulator and mobile robot. The International Journal of Robotics and Automation, 5(1): 90 - 98, 1986

[23] Koren, Y., and Borenstein, J., 1991, Potential Field Methods and Their Inherent Limitations For Mobile Robot Navigation, Proceedings, IEEE International Conference on Robotics and Automation, pp. 1398-1404.

[24] Maciejewski, A. A. and Klein, C. A., 1985 Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments, The International Journal of Robotics Research, V4, No. 3, pp. 109-117.

[25] Quinlan and Khatib, 1993, Elastic Bands: Connecting Path Planning and Control, Proceeding, IEEE Conference on Robotics and Automation, pp 802-807.

[26] M. Khatib, R. Chatila. An extended potential field approach for mobile robot sensor-based motions, In Proc. of Intl. Conf. on Intelligent Autonomous Systems, pages 490-496, 1995.

[27] O. Brock, O. Khatib High-speed Navigation Using the Global Dynamic Window Approach. Proc. IEEE Int. Conf. on Rob. and Aut., Detroit. Michigan USA, May 1999.

[28] O. Brock, O. Khatib, Executing Motion Plams for Robots with Many Degrees of Freedom in Dynamic Environments, In Proc. Int. Conf. on Robotics and Automation, volume 1, pages 1-6, 1998.

[29] Perry, B. R. and Tesar, D., The Development of Distance Functions and Their Higher-Order Properties for Artificial Potential Field-Based Obstacle Avoidance. University of Texas at Austin, February, 1995.

[30] Ge, S.S.; Cui, Y.J., Dynamic Motion Planning for Mobile Robots Using Potential Field Method, Autonomous Robots **13**(3), November, pp. 207-222, 2002,

[31] F. Aurhennamer, Voronoi Diagrams A Survey of a Fundamental Geometric Data Structure ACM Computing Surveys, Vol 23, No 3, September 1991.

[32] K. Jiang, L. D. Seneviratne and S. W. E. Earles, A Shortest Path Based Path Planning Algorithm for Nonholonomic Mobile Robots, Journal of Intelligent and Robotic Systems 24: 347366, 1999.

[33] J. Reif, M. Sharir, Motion Planning in the Presence of Moving Obstacles Journal of the ACM (JACM), v.41 n.4, p.764-790, July 1994

[34] Gilbert, E.G., Johnson, D.W., and Keerthi, S.S., 1988, A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space, IEEE Journal of Robotics and Automation, V4, No. 2, April 1988, pp. 193-203

[35] Cameron, S., 1997a, Enhancing GJK: Computing Minimum and Penetration Distances Between Convex Polyhedra, Proceedings, IEEE International Conference on Robotics and Automation, Albuquerque, NM, April 1997, pp. 3112-3117.

[36] Del Pobil, A. P., Prez, M., and Martnez, B., 1996, A Practical Approach to Collision Detection Between General Objects, Proceedings, IEEE International Conf. on Robotics and Automation, Minneapolis, Minnesota, April 1996, pp. 779-784.

[37] Greenspan, M. and Burtnyk, N., 1996, Obstacle Count Independent Real-Time Collision Avoidance, Proceedings, IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, April 1996, pp. 1073-1080.

5 Redundant Collision Avoidance Strategy

The previous Section has defined a framework that seems feasible to guarantee safety in Footwear scenario; the problem when collaborative task are performed by humans and robot has been deeply described and ***"Safe-net of unsafe-transducers***" has been suggested as feasible solution.

This Section describes the design of all the modules necessary in order to allow **safe collision avoidance among robot and humans operator**. The Section is organized as follows:

- First sub-section describes the framework developed in order to face all the aspects related to collaborative tasks among robot and humans operators;
- Second sub-section describes the collision avoidance algorithms.

HW/SW detailed description of the actual framework implementation is reported in the following Section 6.

5.1 Redundant Collision Avoidance Framework

The Framework designed by ITIA in order to face the safety in collaborative tasks, is based on some assumptions:

- **Collisions** among humans and robot **should happen during shoe-picking** from the manovia and the approaching of the robot to the machines.
- In shoe-picking **trajectory can be modified** both in position and velocity**.**
- When robot is **working on the machine tools, no modification of the trajectory** is allowed.
- When robot is **working on the machine the human has to be outside the clamping-area** close to the robot.
- **SAFE-AREA, WARNING-AREA**, and **SAFE-AREA** (see section 4) geometry has to be modified with respect to the robot operation (***shoe-picking*** or ***working on machine***).
- Within the workspace there are obstacles that can be modeled in a CAD/Vision system.

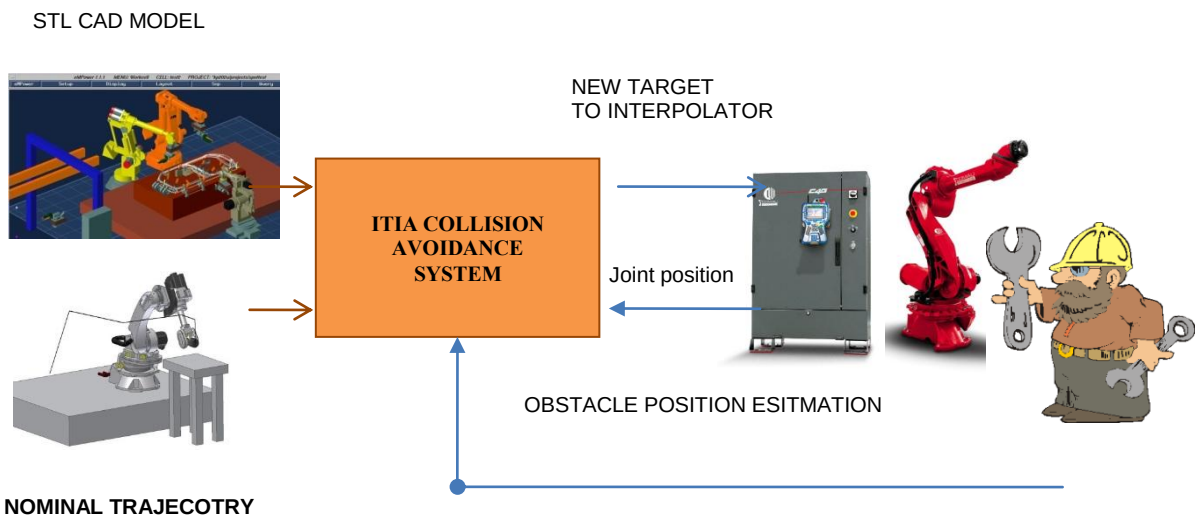


Fig. 22 ITIA's paradigm in collision avoidance strategies

5.1.1 Functional Modules in ITIA's Collision Avoidance strategy

Collision avoidance algorithms developed by ITIA consist of three main software modules:

A. An off-line pre-processing toolbox:
Its role consists of analysing the model of the work cell (that can be acquired by 3D vision system or by available CAD model) and of calculation of possible escape directions for the robot, defined as **pass-through points**.
These points are via-points that are always safely reachable by the robot during the movement. The robot will travel along escape points in case an unwanted obstacle is detected. In fact, for points along the trajectory, escape directions are calculated as resultant force of repulsive forces generated by each the environment model.

B. An on-line routine:
Its role consists of determining which point among the **pass-through points** (calculated in the off-line module) can be selected as new target point for the robot controller.
The selection is done on the basis of the relative distance between the robot and human operators/obstacles in the workspace.
Further control to avoid an impending collision is delegated to speed control, in case avoiding the coming object is not sufficient.

C. Redundancy Check:
All information available in the safe-net has to be verified.
In addition, also the trajectory generated from the robot controller must be redundantly available.
The output of the collision avoidance off-line routine is sent to the robot controller as **new target for its motion interpolator.**

5.1.1.1 Off-line pre-processing toolbox

The off-line routines are implemented in a PC and results are saved in a file. In order to guarantee safety, two checks are foreseen:

A.1 A simple routine (developed in a programming language different from the one used for the calculation of the pass-through points) verifies that each pass-through point is reachable by the robot;

A.2 A simple program that imposes the robot reaches all the points calculated is developed in the robot-programming language (the part program is generated by a **c++ program** that loads the file with the points and generates the PDL2 program for COMAU controller). **Trained human operator has to execute the program and acknowledge if no problem arose during the program execution**.

5.1.1.2 On-line

Since sensors data are available in the net and accessible by different CPUs, the **on-line** routine is executed by two different programs in two different systems.

B.1 The **on-line** data processing is performed by a simple program in the robot controller. The program is written in PDL2, and it is executed each 20ms;

B.2 The **on-line** data processing is performed by a program in an external PC connected to the robot controller by the **c4gopen.** The program is written in **c++,** the **O.S** is GNU/Linux with Xenomai patch. The execution time is fixed to each 10 ms.

The output of the two on-line programs is sent to the **PLC** (see below) and the coherence of the two results is checked. Whether the two software modules have calculated different targets, the system is hold.

5.1.1.3 Redundancy Check

Finally, a further safety control is implemented. The PC connected to the robot controller through the **c4gopen** communication channel is provided by the **ORL-COMAU library (D1.2)**. This library integrates also the robot-interpolator, and, thanks to this functionality, the external **PC** can recalculate the target trajectory once the target has been defined. Hence, the external PC can verify

C.1 the coherence between the target generated by the robot interpolator and the virtual interpolator running on the PC

C.2 the following error of the robot

All the components and the actions are described in figure 22 and 23.

5.1.2 High level description of safe-net implementation

Main limitation of actual safe-plc solutions consists of that they have limited functionalities. In fact they allow only comparing Boolean data and simple logic operation.

Due to these limitations, a safe-net requires also redundant nodes that verify the coherence of all the data in the net. Furthermore, communication with safe-PLC is constrained to the adaptation of a safe protocol of communication. Despite these protocols are published as open standards, easier solution is the integration in the same net of commercial products that support the communication in that safe- protocol.

Under these considerations, consider the figure below:

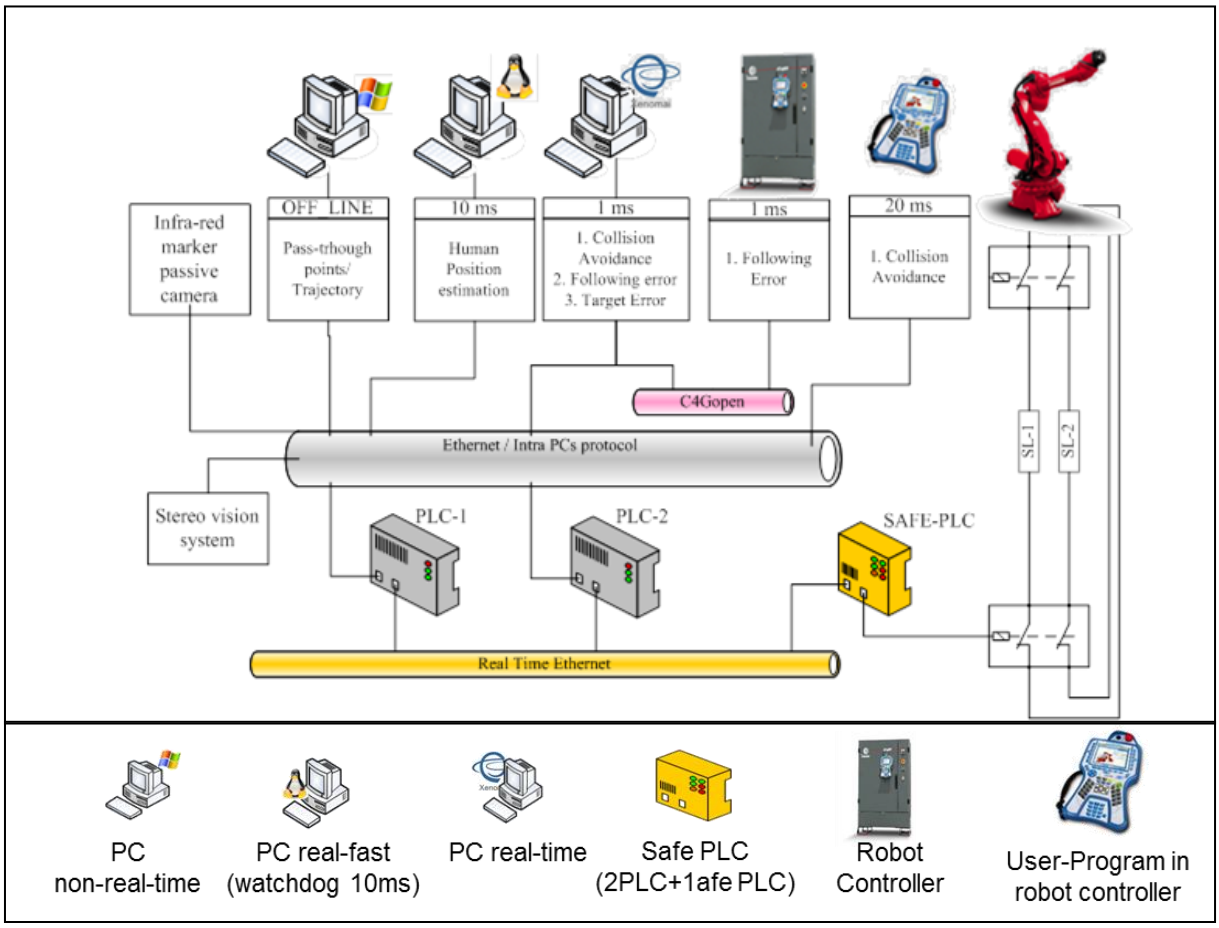


Fig. 23 Safe-net and its components

Attention:

- Real time PC means that the O.S. guarantees a deterministic behavior for the HW/SW, and that it provides high-accurate timer and scheduler that guarantee the cycle time of the application.

- Real-Fast PC means the O.S. must not guarantee deterministic performance, but that is provides high-speed timers, and scheduler guarantees that is the program take too much time, reaction strategies are available.

- **R**obot position transducers are often not redundant**;** this means that robot position knowledge is not certifiable. This should be the bottleneck of the system.
  Two solutions should be provided:
  1. **Use of robot with redundant positions sensors (available in the market);**
  2. **Use of an external vision system for the robot measure.**

Note (refer to Fig. 22):

- Two PLCs verify the coherency of all the data passing through the net, if incoherency is detected they enable the safety procedure in the safe-PLC;

- The two off-the-shelves PLCs implement the safe-communication protocol with the safe-PLC

- A PC elaborates the design data (environment description and trajectory specifications) and defines a grid of safe pass-through points;

- A real-fast PC integrates the sensors fusion algorithms and estimates the human positions to send to the collision avoidance modules;

- If differences between the estimation position and measured data is detected by one of the PLCs, the safety-procedure is enabled;

- Collision avoidance procedures run both in a real-time PC and in a part-program running on the robot controller (20ms).
  These two programs write the outputs in the Ethernet channels, if differences are detected the PLCs enable the safety procedures;

- Following errors are detected by the real-time PC and the robot controller;

- Real time PC integrates the ORL-COMAU library (see DX.X), thanks to this library, it duplicates the robot interpolator. Through c4gopen-channel, the PC check the robot interpolator output if incoherency is detected a signal is sent to the safe-PLC.

Figure below reports a brief flow-chart of the data flow and safety checks foreseen in the architecture.

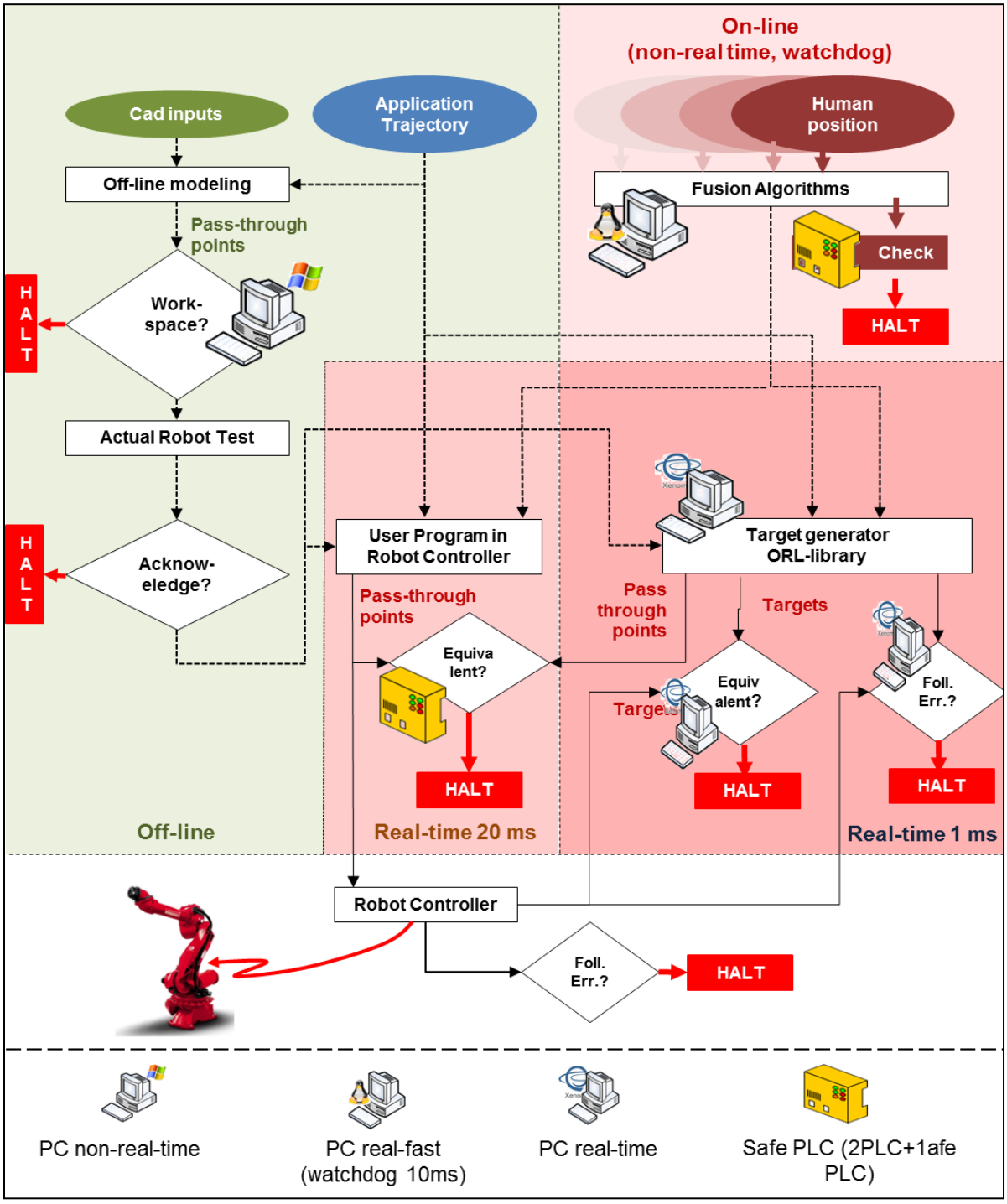


Fig. 24 Data-flow among the components of the safe-net

NOTE: The “bottleneck” of this solution consists of the fact that only one c4gopen-channel is available, resulting the impossibility of making redundant the verification of the targets error calculation. To overcome this problem, a duplication of the c4gopen channel seems the only solution. Duplication should be both at data level and physical layer.

5.2 Collision Avoidance Strategies

The subsection describes the algorithms that have been designed in order to allow the modification of the trajectory of the robot during the execution of the task.

It is worth to underline that when the robot is working on the machine-tools, no modification of the trajectories is allowed and only holding the robot in its position is a feasible solution. Hence, the problem is shifted to the human identification position and measurement of the distance from the robot.

5.2.1 Off-line Environment modeling

5.2.1.1 Obstacles modeling and offsetting

Objects and obstacles inside the collaborative workspace must be modeled in order to allow automatic algorithms for avoidance of collisions among robot, humans and obstacles.

Simplest idea consists in describing the objects through mathematical models. More feasible solutions seem the use of tessellated approximation of the surfaces as methodology to model the environment. In fact, the STL format (that consists in a collection of triangles where each element of the model is detailed by the unit normal and by the three vertexes) is characterized by various positive aspects:

- **It is a widespread standard, introduced by 3D Systems in the late eightieth, created to model geometries in Stereo lithography CAD software.**
- **3D images acquired by different camera system can be easily converted in STL format**

Furthermore, due to simplicity of the mathematical description of solid objects, it is possible to develop algorithms that “offset” the nominal surface in order to allow the model of active “cushions” around the objects.

ITIA has developed an algorithm for the offsetting of tessellated surfaces. Briefly, the idea consists of moving vertexes of triangles, *i.e.*, modifying nodes which define the surface shape. Kim et al. **¡Error! No se encuentra el origen de la referencia.** suggested an algorithm based on multiple normal vectors of a vertex: in the case the node lies on an edge, it is split in various nodes, and both edges and vertex are offset; hence, an increasing of elements with respect to the original model is present. A different approach based on a weighted sum of the normal vectors of the facets that are connected to each vertex has been proposed by Qu at al. **¡Error! No se encuentra el origen de la referencia.**. It is worth to noting that they propose a preprocessing phase to reconstruct some geometry topological information. A limit, as exposed by the same authors, consists on the need of a correct STL model without missing element and/or holes in the triangular mesh. The algorithms provide a new offset algorithm called *Offset Weighted by Angle* (OWA). The identification of the offset direction is based on an evolution of the MWA algorithm and the offset distance is modified on the basis of the local topological properties of the object, *i.e.*, the methodology implements different approach to solve convexity, concavity and saddle nodes.

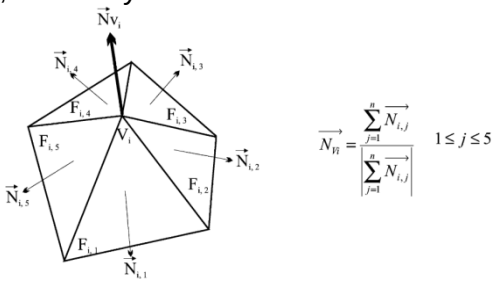


Fig. 25: **Averaged surface normal method for vertex offsetting**

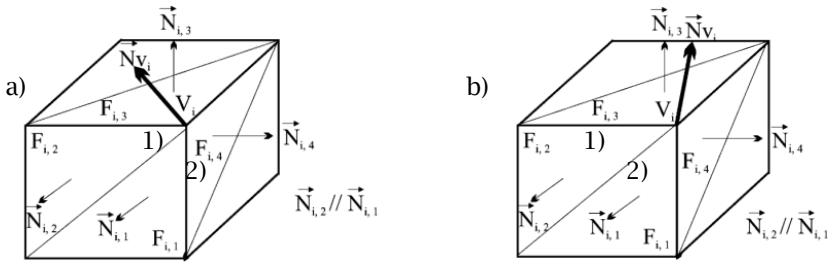


Fig. 26 Drawback of calculating vertex offsetting by the averaged surface normal method. In order to calculate the unit normal vector, all facets adjacent to the vertex are meant in (a) while only facets 1 and 2 are taken into consideration.

5.2.1.2 Pass-through point definition

The algorithm for the definition of the pass-through safe points is below described.

Input: STL-CAD model of the work cell and the Trajectory of the robot TCP

Output: Grid of "escape" points along the nominal trajectory, a virtual force that describes the "repulsive" force of the environment in that point is associated to each point.

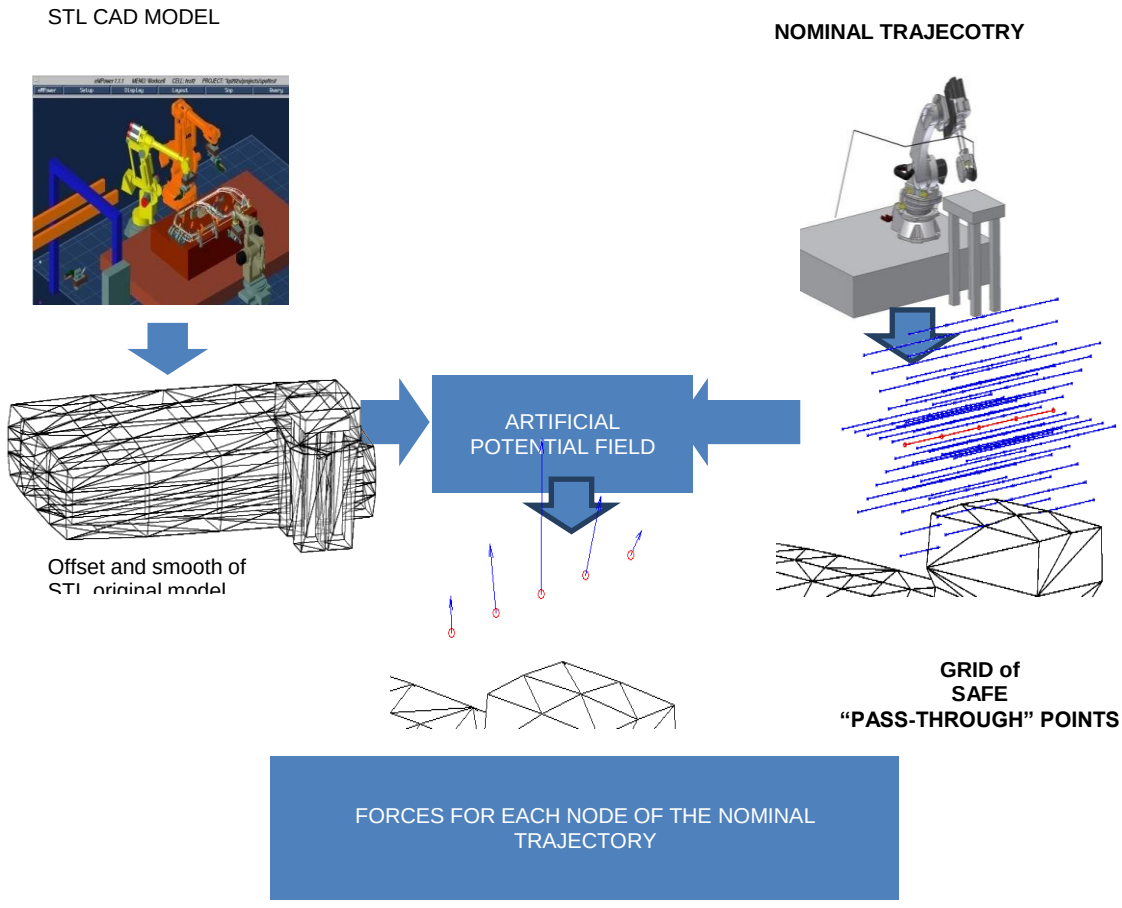


Fig. 1 OFF-LINE Module for the environment modeling and for the definition of the safe grid of "pass-through" points

In order to define the grid of safe “pass-through” points the elaboration phases of the SW modules are:

- ***Refinement of the STL model***;
- ***Offsetting of the STL model by a predefined distance as safety factor. Direction of offset is weighted on the angles between edges converging in the considered node;***
- ***Computation of repulsive force for each point of the trajectory due to each STL face where the repulsive force is given by***

$$\text{RepulsiveForce} \propto \frac{A}{d^n}$$

Where

- A = area of the STL
- d = distance from the trajectory point
- n = distance weight

The repulsive force direction for each point is computed as the vector sum of all STL face contributes. This direction corresponds to evasive normal directions.

On the basis of the repulsive forces direction, the evasive points are computed as “moving” trajectory points normally to the nominal trajectory (i.e. only trajectory normal component – F_n - of the repulsive force is considered; F is the vector sum of all STL faces).

Displacement from nominal point along repulsive direction is predefined (distance d). Interference of the robot with environment objects is pre-verified. The algorithm is applied recursively for each newly computed point.

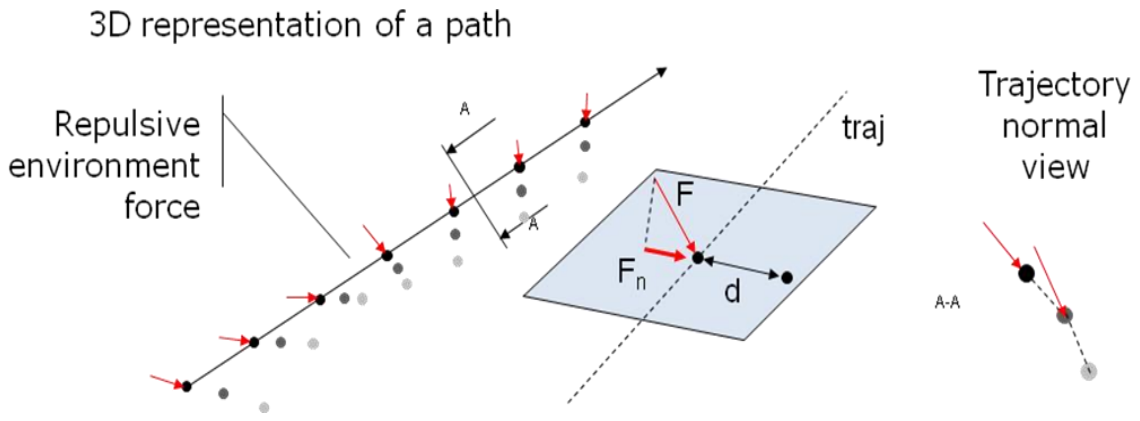


Fig. 2: Schematic representation of the definition of the grid of evasive points

5.2.2 On-line collision detection and avoidance

The control SW is based on these assumptions:

- Only one trained human operator can cooperate actively with the robot.
- The human operator is approaching towards the robot.
- The size of the robot is major or comparable with the human operator.

The control SW for the safe movement is based on the evaluation of:

Input:

- Position of the center of mass of the dynamic obstacle;
- Velocity of the obstacle. In absence of it, an internal algorithm estimates the velocity by filtering the centered first-derivative of the history of the position of the obstacle.
- Trajectory of the robot TCP

Output:

- At each instant in time a new position target for the robot control is chosen within a predefined grid of pass-through points defined in the off-line preprocessing phase of the nominal trajectory.
- The velocity override for the execution of the robot trajectory.

It should be noted that the algorithm chooses one point within a set of pre-defined points and sends the new target to the robot controller. The motion control is completely managed by the robot controller. If the distance between the robot and the human decreases below a safe value the robot task is held (not stopped).The algorithm has been written in PDL2 language which is the high level programming language of the COMAU controllers.

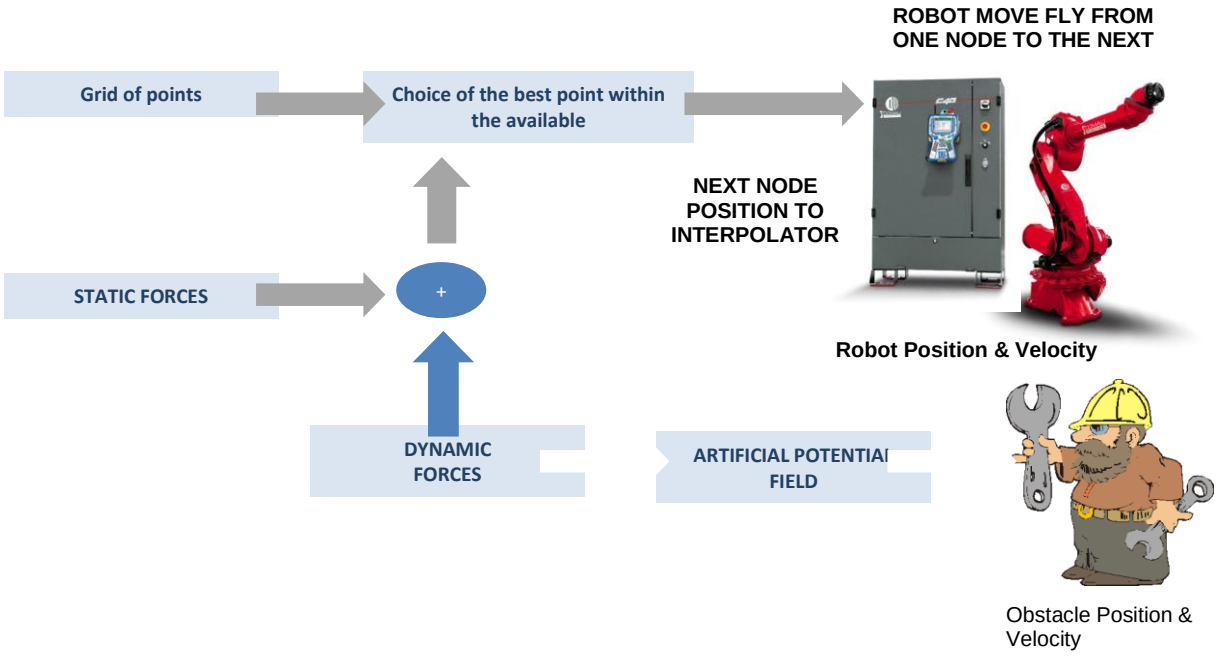


Fig.3 ON-LINE Module: the new target for the robot is automatically calculated in order to maintain the safe distance.

Section Bibliography

[1] STL file format, http://www.3dsystems.com, 3D Systems.

[2] Kim S.-J.; Yang M.-Y.: Triangular mesh offset for generalized cutter, 37(10), 2005, 999-1014.

[3] Chuang C.-M.; Yau H.-T.: A new approach to z-level contour machining of triangulated surface models using fillet endmills, Computer-Aided Design, 37(10), 2005, 1039-1051.

[4] Pedrocchi, N.; Malosio M.; Molinari Tosatti, L.; Ziliani, G.: Obstacle Avoidance Algorithm for Safe Human-Robot Cooperation in Small Medium Enterprise Scenario, 40th International Symposium on Robotics - ISR 09, May 10-13, 2008.

[5] Jang, D.-G.; Park, H.; Kim, K.: Surface offsetting using distance volumes, The International Journal of Advanced Manufacturing Technology, 26(1-2), 2005, 102-108.

[6] Kim S.-J.; Lee D.-Y.; Yang M.-Y.: Offset Triangular Mesh Using the Multiple Normal Vectors of a Vertex, Computer-Aided Design and Application, 1(1-4), 2004, 285-292.

[7] Qu, X.; Stucker, B: A 3D surface offset method for STL-format models, Rapid Prototyping Journal, 9(3), 2003, 133-141.

[8] Koc, B.; Lee, Y.-S.: Non-uniform offsetting and hollowing objects by using biarcs fitting for rapid prototyping processes, Computers in Industry, 47(1), 2002 , 1-23.

[9] Jin, S; Lewis, R.-R.; West, D.: A comparison of algorithms for vertex normal computation, The Visual Computer, 21(1-2), 2005, 71-82.

[10] Thürmer, G; Wüthrich C.-A.: Computing vertex normal from polygonal facets, Journal of Graphics Tools, 3(1), 1998, 43-46.

6 HW/SW solutions and experiments

The Section aims to describe briefly the actual implementation of the CNR-ITIA's framework for ***redundant collision avoidance***

The HW/SW has been designed and chosen in order to:

- ***Validate the concept of SAFETY-NET of UNSAFE SENSORS;***
- ***Identify the technical limitations of actual off-the-shelves devices;***
- ***Validate the redundant collision avoidance algorithms.***

NOTE:
no risk analysis neither risk assessment have been performed. However, authors are confident that a similar solution should face the requirements listed in the standards.

6.1 Redundancy and safety chain

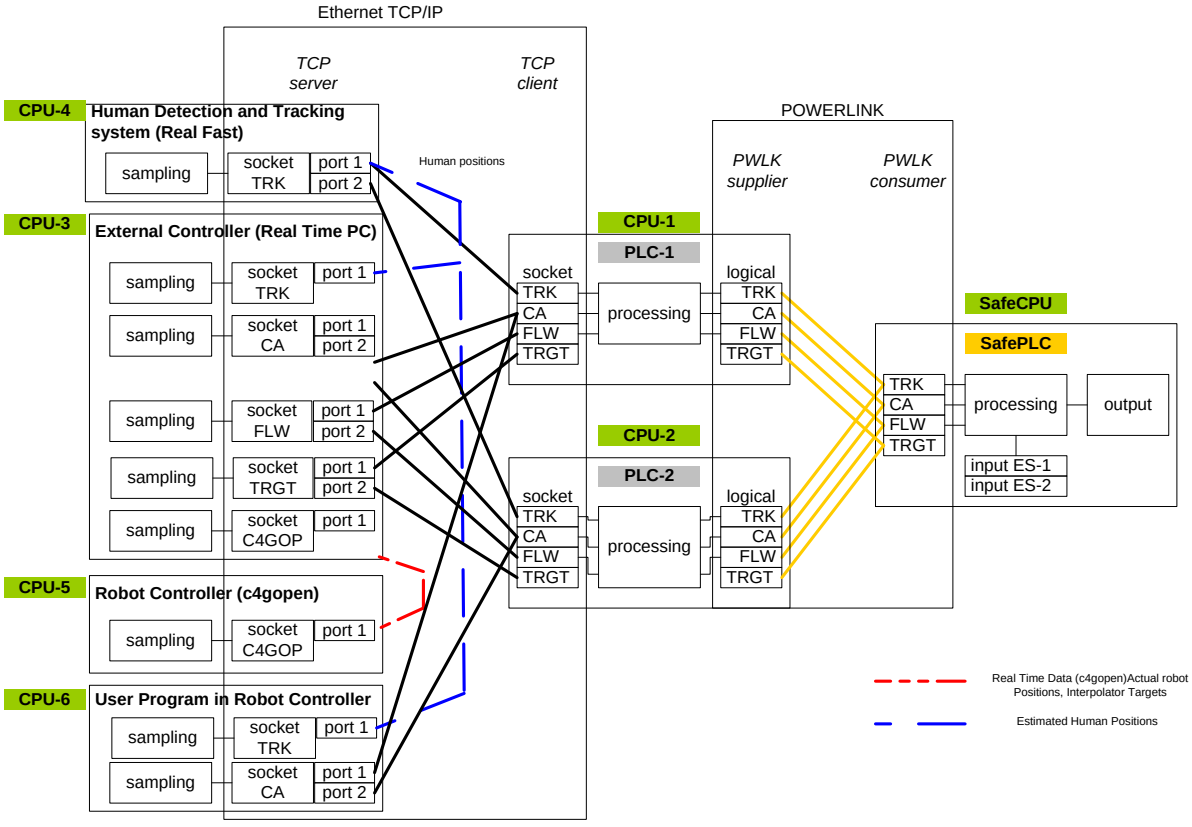


Fig. 27 Data Flow Connection Diagram

Component	Sub-modules	Function&Description	Safe	Comm. channel	Supplier	O.S.	Program. Language
PLC-1 X20CP1484-1	**CPU-1**	Net-observer, Main PLC (Management Node)	No	■ UDP ■ Powerlink	**B&R**	VxWorks	C
X20SI4100	I	Digital Input (4x)	Yes	■ X20 (B&R)	**B&R**	--	--
X20SC2432	IO	Digital Input/Output (4x)	Yes	■ X20 (B&R)	**B&R**	--	--
PLC-2 X20CP1484-1	**CPU-2**	Net-observer, Redundant PLC ((Imaging Control Node)	Yes	■ UDP ■ Powerlink	**B&R**	VxWorks	C
SafePLC	**Safe CPU**	**SafeLOGIC, Open the safe circuit**	Yes	■ X20 ■ Powerlink	**B&R**	VxWorks (safe)	**Proprietary (safe)**
PC-Embedded	**CPU-3**	**Calculus of (i) collision avoidance, (ii) following error, (iii) target coherency**		■ **TCP/IP** ■ **UDP** ■ **c4gopen**	**AAEON**	GNU/Linux patch Xenomai	**C++**
PC-desktop	**CPU-4**	**Sensors fusion, human position estimation**	NO	■ **TCP/IP**	**HP**	GNU/Linux pre-emptible	**Python**
c4gopen	**CPU-5**	**Robot controller (Following error)**	NO	■ **c4gopen**	**COMAU**	N.A.	**N.A**
Teach Pendant	**CPU-6**	**Robot User programming system (Collision avoidance)**	NO	■ **TCP/IP**	**COMAU**	N.A.	**N.A**
PC-desktop	**CPU-7**	**Off-line modeling**	NO	■ **TCP/IP**	**OEM**	Windows	**Python**

The security features are provided by a variety of devices and applications; they are devoted to verification of data integrity and the management of traditional emergency contacts (emergency buttons). The security features offered by the equipment hardware allow the creation of applications with maximum degree (PL**e** EN ISO 13849 and IEC 62061 SIL 3, SIL 3 IEC 61508, IEC 61511 SIL 3).

The actual degree of safety is determined at certification. At this stage the degree of protection is considered according to risk analysis and related risk reduction measures taken.

The monitoring application platform provides two safety levels:

- ***Primary,*** that is achieved through dedicated devices and applications used and developed in accordance with guidelines and standards listed above,

- ***Secondary*** made to level control software residing mainly in the robot controller in order to prevent unfavorable conditions or risky in the handling phase of the machine, such as speed or tracking errors of the position control excessive. The secondary safety does not act on the channels of power of the robot (safety lines SL-1 and SL-2 in Fig. 26) but only at the application level of control and does not respond to verification of redundancy.

The data gathering is performed by a pair of identical PLC (**PLC-1** and **PLC-2**) in order to obtain redundant measurement of the same data. Once collected data is transmitted to the safety PLC (**SafePLC**) which verifies the conditions of

- integrity,
- consistency and
- validity of data
- manual intervention of the primary safety.

Conditions affecting the primary safety due to automatic processing are triggered when the data verification by the **SafePLC** there is evidence of at least any of the following violations:

1. **Violation of the connection: the connection data from at least one PLC-1 and PLC-2 is interrupted;**

2. **Violation of redundancy: the channel data from PLC-1 and PLC-2 does not verify the condition of equality;**

3. **Violation of coherence: data values exceed predetermined threshold values with respect to absolute magnitudes or relative magnitudes of different sources of homogeneous;**

4. **Violation of the operational space: the values of the processed data into information referred the workspace do not match the conditions of belonging to a predetermined volume of job security.**

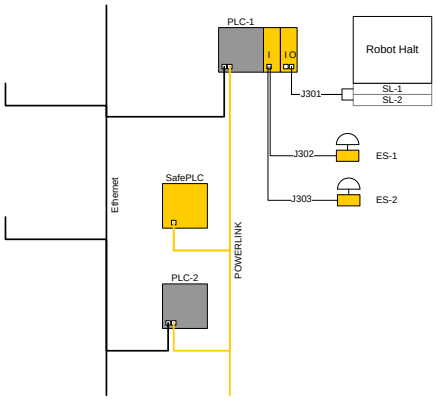


Fig. 28 PLCs connection: the bottleneck of the configuration consists of the PLC-1 is the Powerlink Managing Node and not the SafePLC. It introduces a difference in use of PLC1 and PLC2. This limitation should be considered a minor problem, due the fact that if some problems at Powerlink level, the SafePLC immediately detect them and halt the system.

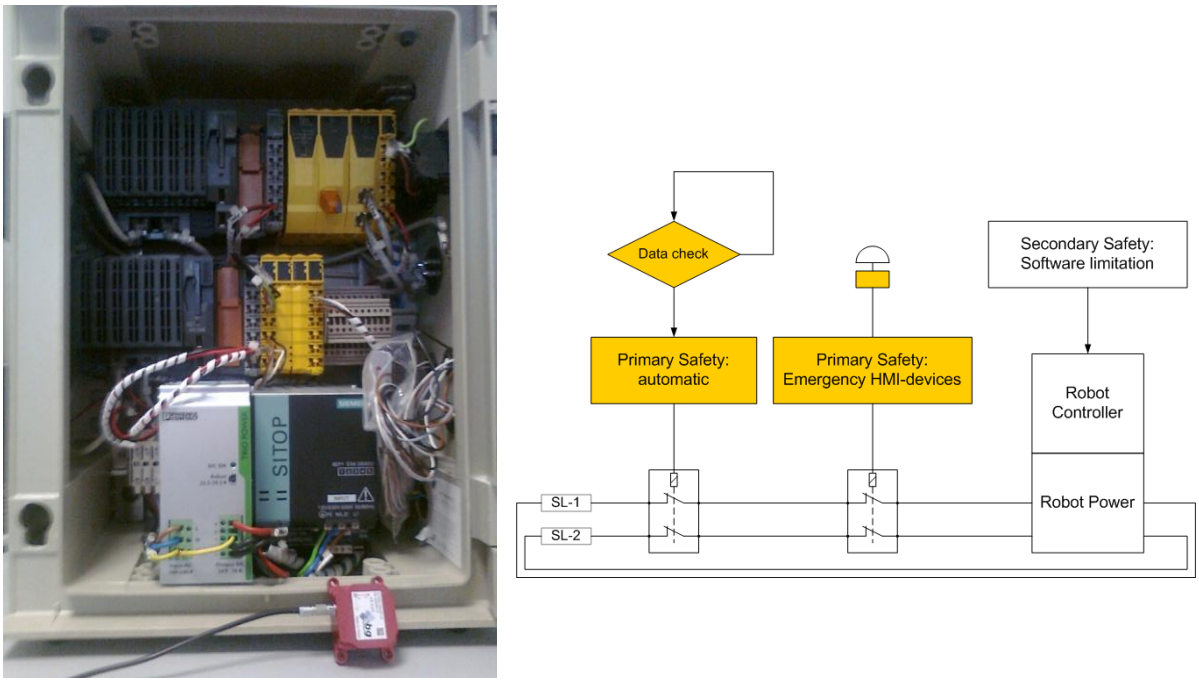


Fig. 29 Safe Rack with the two PLC and the Safe-PLC

6.2 SW Description

Below a list of the SW modules developed and the corresponding platform.

Component	
CPU-1	**Net-observer**
	i) It is the managing node for the safety, because of **OpenSafety (safety on powerlink)** requires its presence as central communication hub. ii) The module is programmed in **c-language (non-posix, proprietary modification of the standard by B&R);** iii) Kinematic Library have been created and compiled in the **PLC;** iv) Verify that data coming from sensors are coherent with themself, and with the calculation exit of sensor-fusion algorithm; v) Verify that the new pass-through point calculated from the collision avoidance strategies running in **CPU-3** and **CPU-6** are coherent vi) Send results of the checks to **SafeCPU through OpenSafe communication channel (safety on powerlink)**
CPU-2	**SAME OPERATION OF CPU-1** **REDUNDANT CALCULUS NODE**
SafePLC	**Safety Reaction Node**
	i) It implements ***the emergency stop algorithm***. ii) The program is implemented by the means of a graphical **LVL-language (similar to a function block language).** iii) All the functions used in the program are **safe-certified.** iv) Physical input/output are **certified modules.** This design choice makes completely safe the reaction of the system to danger situation.

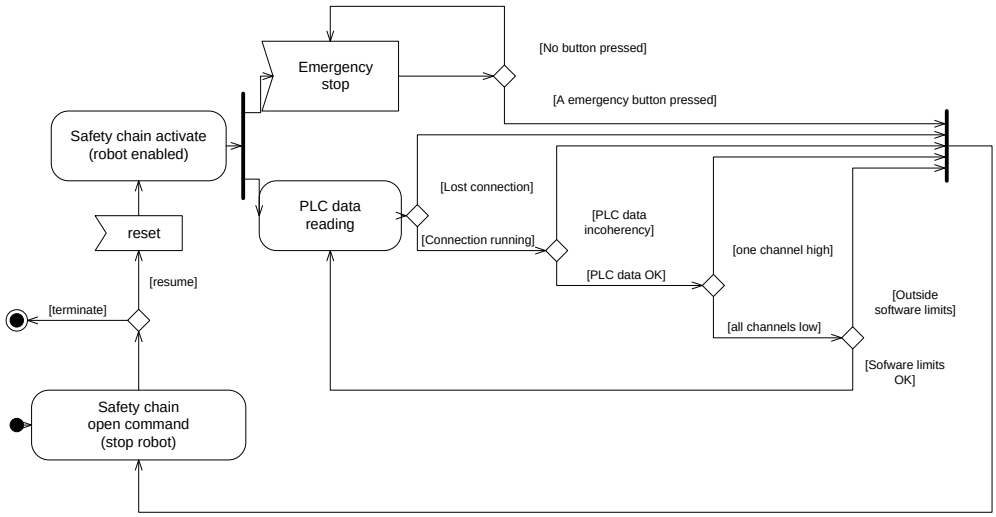


Fig. 30 Emergency stop algorithm implemented in the SafeCPU

CPU-3	**COLLISION AVOIDANCE CALCULUS**

The HW consists of a PC-embedded, with GNU/Linux and real-time Xenomai patch.

The software is **real-time;**

The software is cyclic, with **cycle time equal to 1 ms**

The software has in charge:

i) On the basis of the relative position between the human and the robot it calculates the corresponding state for the **Finite-State-Machine** that manages the robot behavior.
   - ➔ The resulting state is sent to the two PLCs **CPU-2** and **CPU-1** that will compare it with the results of the user-programs running on the robot.

ii) If the human is in the **WARNING-AREA** it calculates at each instant time what will be the best pass-through point for the robot motion planner.
   - ➔ The resulting pass-through point is sent to the two PLCs **CPU-2** and **CPU-1** that will compare it with the results of the user-programs running on the robot.

iii) If the human is in the **DANGER-AREA** it sent an alarm message to both the two PLCs **CPU-1** and **CPU-2**

CPU-3 **(same PC of previous point but different program)**	**ROBOT FOLLOWING ERROR AND TARGET COHERENCY**

The HW consists of a PC-embedded, with GNU/Linux and real-time Xenomai patch.

The software implements the **C4GOPEN SERVER** and it its connect directly to the robot controller

The software is **real-time;**

The software is cyclic, the cycle trigger is given by the c4gopen communication channel

The c4gopen server implement a watchdog with **time equal to 1 ms**

The software has in charge:

i) Calculation of the following error

   → **REDUNDANT WITH RESPECT TO THE COMAU ROBOT CONTROLLER**

ii) Calculation of the robot interpolator target by the means of the COMAU-ORL library

   → **REDUNDANT WITH RESPECT TO THE COMAU ROBOT CONTROLLER**

iii) Comparison between the calculated target position (previous point) and target calculated by the robot controller (available in the c4gopen channel)
   → **IF DIFFERENCES ARE DETECTED, AN ERROR CODE IS SENT TO THE TWO PLCs**

NOTE:

THIS OPERATION IS THE ONLY NON-REDUNDANT OPERATION OF THE ENTIRE FRAMEWORK.

INFORMATION OF THE INTERPOLATOR TARGET IS AVAILABLE ONLY AT C4GOPEN THAT IS NOT A SAFE-COMMUNICATION CHANNEL, AND IT IS A POINT-TO-POINT COMMUNICATION CHANNEL

CPU-4	**SENSORS FUSION, HUMAN POSITION ESTIMATION**
	The HW consists of a PC-desktop, with GNU/Linux (pre-emptible). **NAMELY, THIS CPU should be devoted to the sensor fusion algorithm (and it is necessary in the suggested schema). However, within the ROBOFOOT scenario, NO SENSOR FUSION ALGORITHMS HAVE BEEN DEVELOPED. Simply, in this CPU runs a software that simulates the motion of various human operators inside the workspace and send the measures to the other CPUs.** **THE DATA HAVE BEEN SIMULATED WITH PYTHON (numpy package)**
CPU-5	**ROBOT CONTROLLER**
	The **COMAU** robot controller (SMP+) implements the - **interpolation (redundancy with CPU-3)** - **following error verification (redundancy with CPU-3)**
CPU-6	**ACKNOWLEDGE PROGRAM** **(USER PROGRAM IN ROBOT CONTROLLER)**
	The programming instruments of the standard robot controller have been used in order to develop a software module that on the basis of the humans position calculates i) The program takes as **input the trajectory programmed by the user**. ii) The program sends the trajectory to **CPU-7 (off-line modeling)** that transforms the trajectory in the grid of pass-through points. iii) The program received the **grid of pass-through points** calculated from **CPU-7** iv) The program generates a path that pass-through **ALL THE POINTS THAT CONSTITUTE THE PASS-THROUGH GRID**

	v) The human operator **MUST EXECUTE THE PROGRAM** and verify that all the points calculated are **no-risk** point for the robot (that is, no collision with static obstacles are reachable) vi) The human has to **ACKNOWLEDGE** the points generated vii) When the grid of pass-through-points has been confirmed by human operator, it is stored in the user-space of the robot controller and it can be used by the program that manages the actual movement
CPU-6	**USER TRAJECTORY MANAGER (USER PROGRAM IN ROBOT CONTROLLER)**
	The programming instruments of the standard robot controller have been used in order to develop a software module that transforms a standard path in a list of points The software is written in PLD2 (pascal-like COMAU-property language for user applications) i) The program loads data stored by the acknowledge program **(see previous point)** ii) During the execution of the program it receives from a second program running in **CPU-6 (see next point)** the index of the point among the available of the grids of pass-through points iii) On the basis of the received index it imposes as new target to the robot the new position.
CPU-6	**COLLISION AVOIDANCE MANAGER (USER PROGRAM IN ROBOT CONTROLLER)**
	The programming instruments of the standard robot controller have been used in order to develop a software module that on the basis of the humans position calculates i) What is the state of the FINITE-STATE-MACHINE that manages the robot behavior

	➔ **THE CALCULATED NEW STATE IS SENT TO THE TWO PLCS IN ORDER TO COMPARE THE RESULTS WITH THE ONES CALCULATED FROM CPU-3** ii) On the basis of the humans positions and of the FSM state, the best pass-through point is selected among the available ➔ **THE NEW ROBOT TARGET IS SENT TO THE USER PROGRAM RUNNING IN CPU-6** ➔ **THE NEW ROBOT TARGET IS SENT TO TWO PLCs IN ORDER TO COMPARE THE RESULTS WITH THE ONES CALCULATED FROM CPU-3**
CPU-6	**ACKNOWLEDGE PROGRAM** **(USER PROGRAM IN ROBOT CONTROLLER)** The programming instruments of the standard robot controller have been used in order to develop a software module that on the basis of the humans position calculates i) The programs take as **input the trajectory programmed by the user**. ii) The program sends the trajectory to **CPU-7 (off-line modeling)** that transforms the trajectory in the grid of pass-through points. iii) The program received the **grid of pass-through points** calculated from **CPU-7** iv) The program generates a path that pass-through **ALL THE POINTS THAT CONSTITUTE THE PASS-THROUGH GRID** v) The human operator **MUST EXECUTE THE PROGRAM** and verify that all the points calculated are **no-risk** point for the robot (that is, no collision with static obstacles are reachable) vi) The human has to **ACKNOWLEDGE** the points generated vii) When the grid of pass-through-points has been confirmed by human operator, it is stored in the user-space of the robot controller and it can be used by the program that manages the actual movement

PC-desktop	**OFF-LINE CALCULATION OF THE GRIDS OF PASS-THROUGH-POINTS**
	i) The program takes as input an **STL model of the static obstacles and of the cells** ii) The program takes as input the target trajectory iii) The program generates the grids of pass-through-points iv) The program generates a **PDL2** program for the COMAU controller in order to **test the grids of generated of pass-through points** v) The program send the PDL2 program the COMAU controller

NOTE

Here the PDL2 programs are reported in order to show simplicity in the set-up use from the humans-operators

Three programs are briefly reported

- Socket routines:
  - Read data from tracker module in order to know the estimated position of the humans operator inside the collaborative workspace
  - Send data to the off-line PC (**CPU-7)** for the elaboration of the grids of points and it waits for the results;
  - Send to the PLC the **pass-through point** calculated from the collision avoidance algorithm. The PLC verifies that the point calculated is equivalent to the point calculated by the **CPU-3**
    NOTE: this point is obviously set to the robot controller as new point to reach
- Collision avoidance algorithm:
  the collision avoidance algorithm implements the procedure described above. Simply, a point among the grid of points calculated off-line is chosen as next target on the basis of the position of the human operators.
  NOTES:
  - COLLISION AVOIDANCE IS ACTIVE ONLY IN THE PICK_AND_PLACE MOVEMENT FROM THE MANOVIA TO THE MACHINE
  - All the trajectories are split in a grid of pass-through points, calculated from CPU-7 (the calculus is off-line, but synchronized with the PDL2 program)

- The movement program:
  The program is extremely easy, since, the standard **MOVE LINEAR** instruction is replaced by a **ITIA_PRPL_MOVE_LINEAR_TO** instruction. This procedure makes transparent both the socket communication and the collision avoidance algorithm to the user.

The movement program:

```
PROGRAM ITIA_PRPL_avoidance HOLD,  STACK = 10000
-- ********************************************************************************************
-- PDL2 PROGRAM developed by ITIA for Automatica fair
--        authors   : nicola pedrocchi
--        contact   : nicola.pedrocchi@itia.cnr.it  +39 (0)2
--
-- BASIC IDEA:
-- the motion from generic point A to generic point B is a collection of transit nodes (the resultant path is
-- named "pth" in the PDL2 program). Before the start  of the motion, an algorithm create a set of
-- evasive transition points for each node of the path (these points are stored in the variables "pth0, pth0tr,
-- pth1,.." ). The robot is required to follow the path, and at each instant time the next node it has to reach
-- is chosen among the pre-defined points.
-- ********************************************************************************************



============================================================================================
-- HEADER SECTION ==========================================================================
============================================================================================
TYPE
nd = NODEDEF -- PATH node definition
     $MAIN_POS
     $COND_MASK
     $COND_MASK_BACK
ENDNODEDEF

CONST
VAR
   ... (some lines of code are not reported )

    LOCK_MOVEMENT_ACTL        : BOOLEAN           EXPORTED FROM prog_ipa2itia
    LOCK_MOVEMENT_NEXT        : BOOLEAN           EXPORTED FROM ITIA_PRPL_functions
    MOVEMENT_LOCKED           : BOOLEAN EXPORTED FROM ITIA_PRPL_functions
    NUM_NODES                 : INTEGER EXPORTED FROM ITIA_PRPL_functions
    ROUTINE ITIA_INC_IND_ACT_NODE : INTEGER       EXPORTED FROM ITIA_PRPL_functions
   ... (some lines of code are not reported )

-----------------------------------------------------------------------
-- MAIN ROUTINE FOR THE MOVEMENT OF THE ROBOT
ROUTINE ITIA_PRPL_MOVE
VAR
     I : INTEGER
BEGIN

     WRITE LUN_CRT('AV...INIT PREPLANNED AVOIDANCE ALGORITHM: RUN')
     SIGNAL         INITFLAG
     WAIT           INITFLAG
     WRITE LUN_CRT(' / OK ',NL)
     ENABLE CONDITION[ 1 ]

     WRITE LUN_CRT('AV...MOVE: RUN / ',NL )
     SIGNAL         RUNFLAG
     I := ITIA_INC_IND_ACT_NODE
     WHILE  I <= NUM_NODES DO

          MOVEFLY LINEAR TO pth.NODE[ I ].$MAIN_POS ADVANCE -core of the application
          I := ITIA_INC_IND_ACT_NODE

     ENDWHILE

     WAIT           RUNFLAG
     WRITE LUN_CRT('/ DONE',NL)

END ITIA_PRPL_MOVE
--
--
ROUTINE ITIA_MOVEMENT_LOCK_AND_RESUME                              -- to manage when human is in the DANGER AREA
BEGIN
    IF (LOCK_MOVEMENT_ACTL OR LOCK_MOVEMENT_NEXT) THEN
        LOCK
        MOVEMENT_LOCKED := TRUE
        IF NOT ALREADY_PRINTED THEN
            WRITE LUN_CRT(NL)
            WRITE LUN_CRT ( 'AV****OBST. TOO CLOSE LOCK (', LOCK_MOVEMENT_ACTL)
            WRITE LUN_CRT ( '/', LOCK_MOVEMENT_NEXT, ')', NL)
            ALREADY_PRINTED := TRUE
        ENDIF
    ENDIF
    IF (NOT LOCK_MOVEMENT_ACTL ) AND (NOT LOCK_MOVEMENT_NEXT) THEN
        IF MOVEMENT_LOCKED THEN
            WRITE LUN_CRT('AV**** RESUME (',LOCK_MOVEMENT_ACTL,'/',LOCK_MOVEMENT_NEXT,')',NL)
            UNLOCK
            RESUME
            MOVEMENT_LOCKED := FALSE
            ALREADY_PRINTED := FALSE
        ENDIF
    ENDIF
    $TIMER[1] := 0
    ENABLE CONDITION[1]
END ITIA_MOVEMENT_LOCK_AND_RESUME
--
-- ============================================================
-- MAIN PROGRAM SECTION  ======================================
-- ============================================================
BEGIN
-----------------------------------------------------------------
CONDITION[1] :
    WHEN $TIMER[1] > 2 DO
        ITIA_MOVEMENT_LOCK_AND_RESUME
ENDCONDITION
-----------------------------------------------------------------
ALREADY_PRINTED              := FALSE
MOVEMENT_LOCKED              := FALSE
```

```
101.      LOCK_MOVEMENT_ACTL  := FALSE
102.      LOCK_MOVEMENT_NEXT  := FALSE
103.      $SPD_OPT                    := SPD_LIN
104.      $LIN_SPD                    := 0.25
105.      $FLY_TYPE           := FLY_CART
106.      ATTACH    $TIMER[ 1 ]
107.      $TIMER[           1 ]       := 0
108.
109.      MOVE JOINT TO POS(  -1000, -1500, 1000, 100, 170,-122 ,'')
110.
111.      WHILE TRUE DO
112.          ITIA_PRPL_MOVE( POS(   1000, -1500, 1000, 100, 170, -122,'W'))
113.          ITIA_PRPL_MOVE( POS(  -1000, -1500, 1000, 100, 170,-122 ,'W'))
114.      ENDWHILE
115.
116.      END ITIA_PRPL_avoidance
```

PDL2 program of Collision avoidance algorithm

NOTE: a similar implementation of this algorithm is in CPU-3, the real time program. The calculation redundancy is so guaranteed.

```
1.        PROGRAM ITIA_PRPL_functions DETACH, NOHOLD, STACK = 50000
2.        ============================================================================================
3.        HEADER SECTION ==============================================================================
4.        ============================================================================================
5.        TYPE
6.        nd = NODEDEF -- PATH node definition
7.            $MAIN_POS
8.            $COND_MASK
9.            $COND_MASK_BACK
10.       ENDNODEDEF
11.
12.       CONST
13.
14.       LINEAR_MOVEMENT_TYPE                    = 0
15.       CIRCULAR_MOVEMENT_TYPE        = 1
16.       OBST_WARNING_RADIUS           = 1100
17.       OBST_HAZARD_RADIUS            = 750
18.       OBST_PROHIBITED_RADIUS        = 600
19.       ROB_WARNING_RADIUS            = 2650
20.       ROB_PROHIBITED_RADIUS                   = 900
21.
22.       VAR
23.
24.       global_obstacle_position      : ARRAY[3] OF REAL  EXPORTED FROM  prog_ipa2itia
25.       global_start_position         : POSITION          EXPORTED FROM ITIA_PRPL_functions
26.       global_via_position           : POSITION          EXPORTED FROM ITIA_PRPL_functions
27.       global_end_position           : POSITION          EXPORTED FROM ITIA_PRPL_functions
28.       MOTION_TYPE                   : INTEGER EXPORTED FROM ITIA_PRPL_functions        --linear/circular
29.       NUM_PATHS                     : INTEGER
30.       NUM_NODES                     : INTEGER EXPORTED FROM ITIA_PRPL_functions
31.       pth                           : PATH OF nd        EXPORTED FROM ITIA_PRPL_functions
32.       pth0                          : PATH OF nd        -- is the nominal path
33.       pth1                          : PATH OF nd        --
34.       pth2                          : PATH OF nd        -- is the path corresponding to a low level of danger
35.       pth3                          : PATH OF nd        --
36.       pth4                          : PATH OF nd        -- is the path corresponding to a medium level of danger
37.       pth5                          : PATH OF nd        --
38.       pth6                          : PATH OF nd        -- is the path corresponding to a critical level of danger
39.       pth7                          : PATH OF nd        --
40.       pth8                          : PATH OF nd        -- is the path corresponding to a high level of danger
41.
42.       IND_ACT_NODE                  : INTEGER           -- index that store the last node of the for wich tcp is passed
43.       TRIGGER_ACT_NODE    : BOOLEAN           -- store if the transition has been locked
44.
45.       ALREADY_PRINTED     : BOOLEAN
46.       LOCK_MOVEMENT_NEXT  : BOOLEAN EXPORTED FROM ITIA_PRPL_functions
47.       MOVEMENT_LOCKED     : BOOLEAN EXPORTED FROM ITIA_PRPL_functions
48.
49.       ALGDONE             : BOOLEAN
50.       RUNFLAG             : SEMAPHORE EXPORTED FROM ITIA_PRPL_functions   NOSAVE
51.       INITFLAG            : SEMAPHORE EXPORTED FROM ITIA_PRPL_functions   NOSAVE
52.
53.       ROUTINE ITIA_INC_IND_ACT_NODE : INTEGER EXPORTED FROM ITIA_PRPL_functions
54.
55.
56.
57.       ------------------------------------------------------------------------------------------------
58.       -- 1) This routine defines the corresponding path to the desired motion
59.       ROUTINE ITIA_PATH_INIT( move_type : integer) -- move_type = LINEAR [0] or CIRCULAR [1]
60.       VAR
61.           -- the function initializes ALL THE GRID OF POINTS:
62.           --        ALL THE PASS-THROUGH POINTS IDENTIFIED IN DESIGN PHASE ARE STORED IN THE LIST OF NODES
63.           --        pth-k
64.           --        Identification of the pass-through points has to be fixed on the basis of the application
65.           --        and of the environment description
66.
67.       END ITIA_PATH_INIT
68.
69.       ------------------------------------------------------------
70.       ROUTINE ITIA_PREPLANNED_AVOIDANCE
71.       VAR
72.       J                                       : INTEGER
73.       distance_pth0_obstacle        : REAL
74.       distance_pth1_obstacle        : REAL
```

```
75.       distance_pth2_obstacle       : REAL
76.       distance_pth3_obstacle       : REAL
77.       distance_pth4_obstacle       : REAL
78.       distance_pth5_obstacle       : REAL
79.       distance_pth6_obstacle       : REAL
80.       distance_pth7_obstacle       : REAL
81.       distance_pth8_obstacle       : REAL
82.
83.       BEGIN
84.
85.       IF        (IND_ACT_NODE < NUM_NODES - 2 ) THEN
86.
87.         pth.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS          := pth0.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS
88.         distance_pth0_obstacle := dist(global_obstacle_position - pth0.NODE[ IND_ACT_NODE ].$MAIN_POS )
89.         distance_pth1_obstacle := dist(global_obstacle_position - pth1.NODE[ IND_ACT_NODE ].$MAIN_POS )
90.         distance_pth2_obstacle := dist(global_obstacle_position - pth2.NODE[ IND_ACT_NODE ].$MAIN_POS )
91.         distance_pth3_obstacle := dist(global_obstacle_position - pth3.NODE[ IND_ACT_NODE ].$MAIN_POS )
92.         distance_pth4_obstacle := dist(global_obstacle_position - pth4.NODE[ IND_ACT_NODE ].$MAIN_POS )
93.         distance_pth5_obstacle := dist(global_obstacle_position - pth5.NODE[ IND_ACT_NODE ].$MAIN_POS )
94.         distance_pth6_obstacle := dist(global_obstacle_position - pth6.NODE[ IND_ACT_NODE ].$MAIN_POS )
95.         distance_pth7_obstacle := dist(global_obstacle_position - pth7.NODE[ IND_ACT_NODE ].$MAIN_POS )
96.         distance_pth8_obstacle := dist(global_obstacle_position - pth8.NODE[ IND_ACT_NODE ].$MAIN_POS )
97.
98.         ----------------------------------------------------
99.         IF distance_pth0_obstacle > OBST_WARNING_RADIUS THEN
100.          pth.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS := pth0.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS
101.          IF LOCK_MOVEMENT_NEXT THEN
102.             WRITE LUN_CRT('FN**** UNLOCK AND RESUME OF THE MOTION ',NL)
103.             LOCK_MOVEMENT_NEXT := FALSE
104.             ALREADY_PRINTED         := FALSE
105.           ENDIF
106.        ELSE
107.        ----------------------------------------------------
108.        IF distance_pth1_obstacle > OBST_HAZARD_RADIUS THEN
109.          pth.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS := pth1.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS
110.          IF LOCK_MOVEMENT_NEXT THEN
111.            WRITE LUN_CRT('FN**** UNLOCK AND RESUME OF THE MOTION ',NL)
112.            LOCK_MOVEMENT_NEXT := FALSE
113.            ALREADY_PRINTED         := FALSE
114.          ENDIF
115.        ELSE
116.        ----------------------------------------------------
117.        IF distance_pth2_obstacle > OBST_HAZARD_RADIUS THEN
118.          pth.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS := pth2.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS
119.          IF LOCK_MOVEMENT_NEXT THEN
120.             WRITE LUN_CRT('FN**** UNLOCK AND RESUME OF THE MOTION ',NL)
121.             LOCK_MOVEMENT_NEXT := FALSE
122.             ALREADY_PRINTED         := FALSE
123.           ENDIF
124.        ELSE
125.        ----------------------------------------------------
126.        IF distance_pth3_obstacle > OBST_HAZARD_RADIUS THEN
127.          pth.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS := pth3.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS
128.          IF LOCK_MOVEMENT_NEXT THEN
129.            WRITE LUN_CRT('FN**** UNLOCK AND RESUME OF THE MOTION ',NL)
130.            LOCK_MOVEMENT_NEXT := FALSE
131.            ALREADY_PRINTED         := FALSE
132.          ENDIF
133.        ELSE
134.        ----------------------------------------------------
135.        IF distance_pth4_obstacle > OBST_HAZARD_RADIUS THEN
136.          pth.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS := pth4.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS
137.          IF LOCK_MOVEMENT_NEXT THEN
138.             WRITE LUN_CRT('FN**** UNLOCK AND RESUME OF THE MOTION ',NL)
139.             LOCK_MOVEMENT_NEXT := FALSE
140.             ALREADY_PRINTED         := FALSE
141.          ENDIF
142.        ELSE
143.        ----------------------------------------------------
144.        IF distance_pth5_obstacle > OBST_HAZARD_RADIUS THEN
145.          pth.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS := pth5.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS
146.          IF LOCK_MOVEMENT_NEXT THEN
147.            WRITE LUN_CRT('FN**** UNLOCK AND RESUME OF THE MOTION ',NL)
148.            LOCK_MOVEMENT_NEXT := FALSE
149.            ALREADY_PRINTED         := FALSE
150.          ENDIF
151.        ELSE
152.        ----------------------------------------------------
153.        IF distance_pth6_obstacle > OBST_HAZARD_RADIUS THEN
154.          pth.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS := pth6.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS
155.          IF LOCK_MOVEMENT_NEXT THEN
156.             WRITE LUN_CRT('FN**** UNLOCK AND RESUME OF THE MOTION ',NL)
157.             LOCK_MOVEMENT_NEXT := FALSE
158.             ALREADY_PRINTED         := FALSE
159.          ENDIF
160.        ELSE
161.        ----------------------------------------------------
162.        IF distance_pth6_obstacle > OBST_HAZARD_RADIUS THEN
163.          pth.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS := pth7.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS
164.            IF LOCK_MOVEMENT_NEXT THEN
165.              WRITE LUN_CRT('FN**** UNLOCK AND RESUME OF THE MOTION ',NL)
166.              LOCK_MOVEMENT_NEXT := FALSE
167.              ALREADY_PRINTED         := FALSE
168.            ENDIF
169.        ELSE
170.        ----------------------------------------------------
171.          pth.NODE[ IND_ACT_NODE + 2 ].$MAIN_POS := pth8.NODE[   IND_ACT_NODE + 2 ].$MAIN_POS
172.          IF (distance_pth8_obstacle < OBST_PROHIBITED_RADIUS) THEN
173.            LOCK_MOVEMENT_NEXT      := TRUE
174.            IF NOT ALREADY_PRINTED  THEN
175.              WRITE LUN_CRT(NL)
176.              WRITE LUN_CRT('FN**** LOCK OF THE MOTION OBSTACLE TOO CLOSE TO THE TCP',NL)
177.              ALREADY_PRINTED       := TRUE
```

```
178.            ENDIF
179.            DELAY 200
180.          ELSE
181.          IF LOCK_MOVEMENT_NEXT THEN
182.            WRITE LUN_CRT(NL)
183.            WRITE LUN_CRT('FN**** UNLOCK AND RESUME OF THE MOTION ',NL)
184.            LOCK_MOVEMENT_NEXT      := FALSE
185.            ALREADY_PRINTED         := FALSE
186.          ENDIF
187.        ENDIF
188.        ENDIF
189.        ENDIF
190.        ENDIF
191.        ENDIF
192.        ENDIF
193.        ENDIF
194.        ENDIF
195.        ENDIF
196.        TRIGGER_ACT_NODE := FALSE
197.      ENDIF
198.
199.      END ITIA_PREPLANNED_AVOIDANCE
200.
201.
202.
203.      ------------------------------------------------------------------
204.      -- ROUTINE INC_IND_ACT_NODE and ROUTINE DEC_IND_ACT_NODE
205.      -- are two functions that trigger the transition for a new node of the path.
206.      ------------------------------------------------------------------
207.      ROUTINE ITIA_INC_IND_ACT_NODE : INTEGER
208.      VAR
209.          distance_pth0_obstacle    : real
210.          distance_pth1_obstacle    : real
211.          distance_pth2_obstacle    : real
212.          distance_pth3_obstacle    : real
213.          distance_pth4_obstacle    : real
214.      BEGIN
215.          IND_ACT_NODE    := IND_ACT_NODE + 1
216.          TRIGGER_ACT_NODE          := TRUE
217.          IF IND_ACT_NODE < NUM_NODES THEN
218.              ALGDONE := FALSE
219.          ELSE
220.              ALGDONE := TRUE
221.          ENDIF
222.          RETURN( IND_ACT_NODE )
223.
224.      END ITIA_INC_IND_ACT_NODE
225.      ------------------------------------------------------------------
226.
227.
228.      -- ===============================================================================================
229.      -- MAIN SECTION ==================================================================================
230.      -- ===============================================================================================
231.      BEGIN
232.      --------------------
233.          CANCEL          RUNFLAG
234.          CANCEL          INITFLAG
235.
236.          LOCK_MOVEMENT_NEXT        := FALSE
237.          MOVEMENT_LOCKED := FALSE
238.          NUM_PATHS                 := 9
239.          NUM_NODES                 := 50
240.          WRITE LUN_CRT('NUM_NODES ',NUM_NODES ,NL)
241.          NODE_APP( pth  , NUM_NODES )
242.          NODE_APP( pth0 , NUM_NODES )
243.          NODE_APP( pth1 , NUM_NODES )
244.          NODE_APP( pth2 , NUM_NODES )
245.          NODE_APP( pth3 , NUM_NODES )
246.          NODE_APP( pth4 , NUM_NODES )
247.          NODE_APP( pth5 , NUM_NODES )
248.          NODE_APP( pth6 , NUM_NODES )
249.          NODE_APP( pth7 , NUM_NODES )
250.          NODE_APP( pth8 , NUM_NODES )
251.
252.      CYCLE
253.
254.          INITIALIZATION PHASE: RUN
255.
256.          WAIT   INITFLAG                   -- intialization
257.          ITIA_PATH_INIT( MOTION_TYPE )
258.          SIGNAL          INITFLAG
259.
260.          WAIT  RUNFLAG                               -- running
261.          ALGDONE := FALSE
262.
263.          WHILE NOT ALGDONE DO
264.          IF TRIGGER_ACT_NODE OR LOCK_MOVEMENT_NEXT THEN
265.            ITIA_PREPLANNED_AVOIDANCE
266.          ELSE
267.            DELAY 100
268.          ENDIF
269.
270.      ENDWHILE
271.
272.
273.      WRITE LUN_CRT(' STOPPING ')
274.      DELAY 1000
275.

END ITIA_PRPL_functions
```

6.2 Experiment Description

NOTES:

- **No real sensors have been used in the framework**
- **Positions of humans inside the cell have been simulated**
- **Only basic fusion algorithms have been implemented due the virtual set-up. Further investigations and researches in this field of activities are necessary**

The algorithm has been tested on a COMAU NS16 available at CNR-ITIA laboratory.

It is a serial anthropomorphic robot arm with a maximum extension of 1.650 [m].

A toolbox for the analysis of the STL file and of the nominal path has been developed in **PYTHON** (by the means of ***numpy*** package).

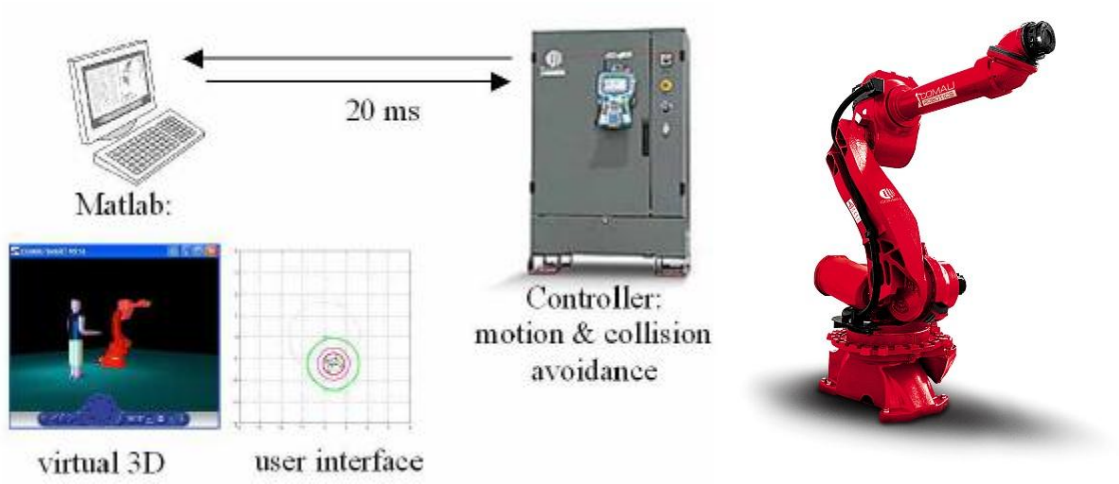
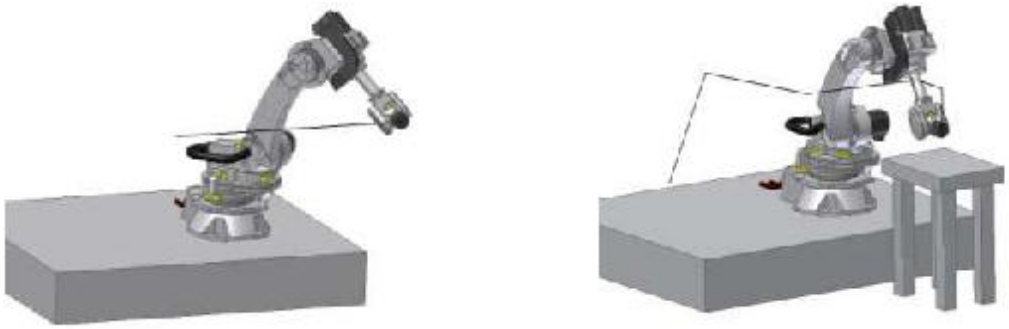


Figure 4 Experimental set-up: the human obstacle has been virtualized and a delay in the communication has been introduced



(a) (b)

Fig. 31 Different test to verify the feasibility of the control strategy investigated

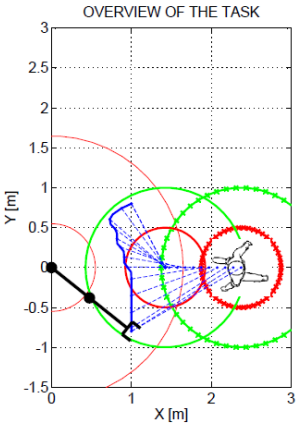
The test cases have been developed using a virtual obstacle. The position of the obstacle is imposed by a ***Python*** user interface and it can be controlled both by dragging with the mouse a human icon inside a graph representing the workspace or by imposing a motion law to the obstacle. In both cases, the obstacle position is sent via TCP-IP socket to the C4G controller, while the robot joints position are sent to the ***Python*** application to update a virtual 3D environment and to elaborate the experimental results.

The experiments have been performed imposing an horizontal linear trajectory of 1,600 [mm] length far away from the static obstacles, *i.e.*, the environment forces are neglectable.

The robot nominal linear velocity has been fixed at 1,000 [mm/s], although it is achievable only in absence of obstacles. Note that the ISO norm fixes at 250 [mm/s] the maximum velocity for the TCP of the robot during the programming phase in which the interaction with human is allowed. Four different tests have been performed and reported below.

Test 1

The operator is slowly approaching perpendicularly to the robot trajectory, with a linear velocity of 250 [mm/s].



As soon as the operator is nearer to the robot than the *warning distance D*, the algorithm is activated, *i.e.*, the robot deviates with respect from to the nominal trajectory and the velocity override is calculated. Note that the algorithm tries to maintain the robot outside the ***DANGER-AREA***, however for the last point of the trajectory this behavior would cause the not completion of the task. To face this problem if the last node is outside the ***DANGER-AREA***, the robot is allowed to reduce its distance from the obstacle and achieve the goal of the task, and the override modification is always kept active as shown in the Figure below.

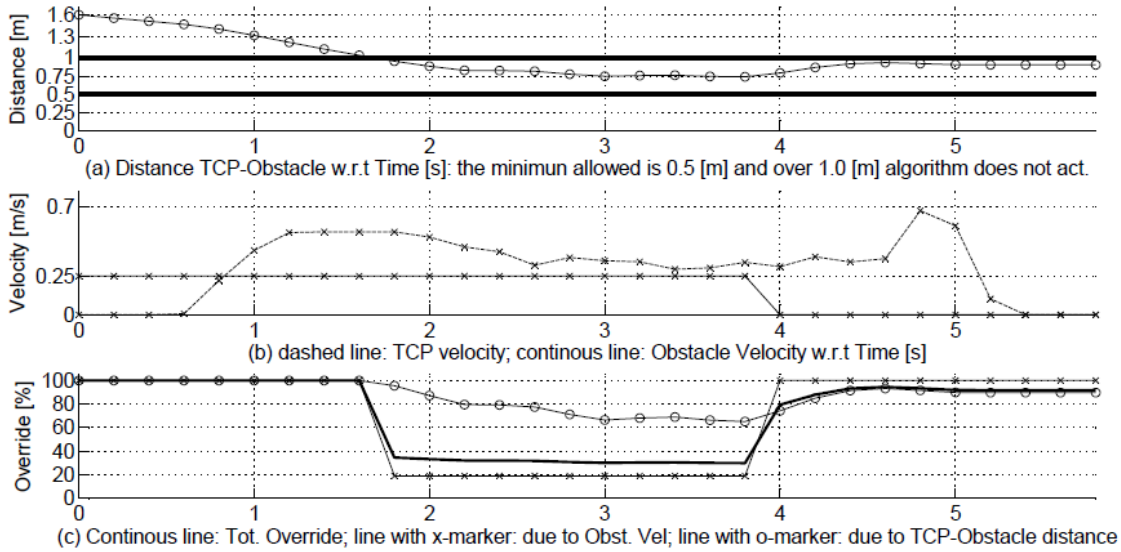
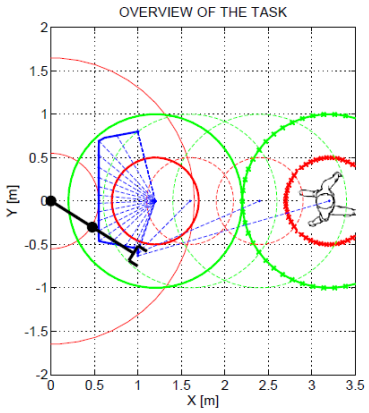


Fig. 32 Test 1

Test 2



The human is quickly approaching perpendicularly to the robot trajectory, with a linear velocity of 2000 mm/s.

By increasing the obstacle velocity any appreciable change is noticed in the robot trajectory.

The robot velocity is quickly reduced and this reaction allows the robot to remain outside the **DANGER-AREA**

Coming back to the test result analysis, when the human stops its motion, the robot controller looks for the trajectory that allows the greatest distance between the TCP and the human, within the available grid of "pass-through" points.

As in the previous case, the target point is inner the **WARNING AREA.**

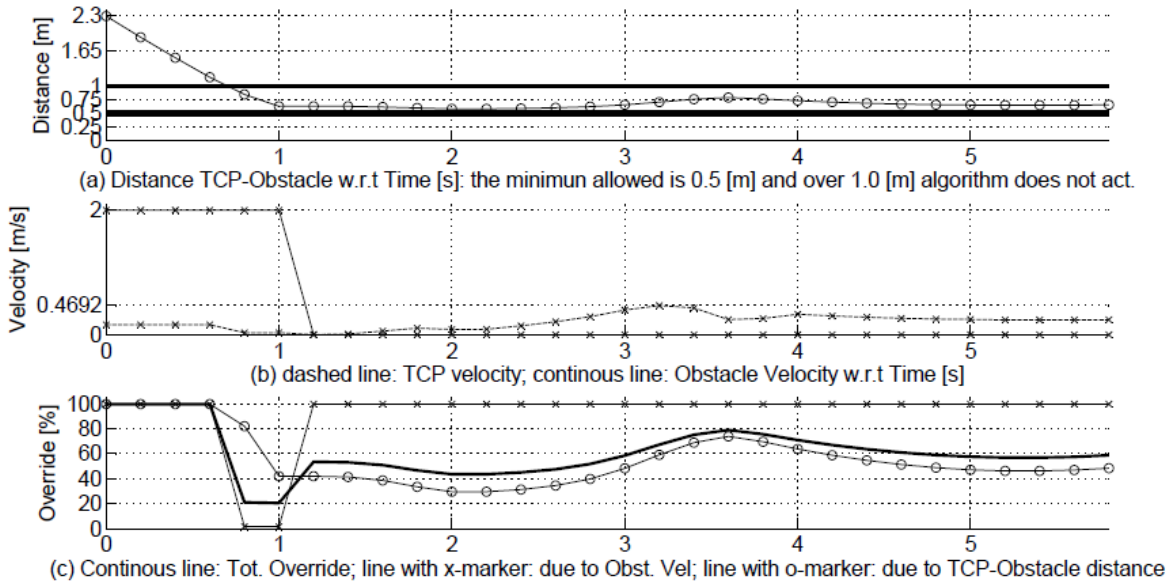
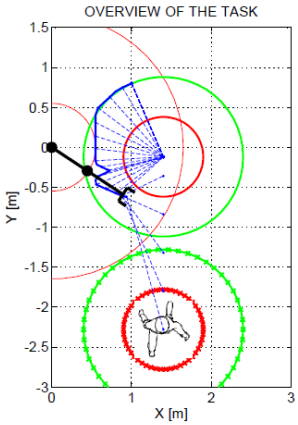


Fig. 33 Test 2

Test 3



The human is moving in parallel to the robot trajectory in the same direction, with an offset of 400 [mm] and with a velocity of 1,200 [mm/s].

Note that at time t=1.4 [s] the TCP goes in the DANGER AREA and the task is suspended.

Due to, the different task priorities in this phase in the COMAU controller, a short queue in the communication between the robot controller and the Python-PC is created and when the robot motion is resumed, *i.e.*, when the obstacle-robot distance gets greater than *d*, the motion of the robot is not clear due to the inconsistency of the information about the position of the obstacle.

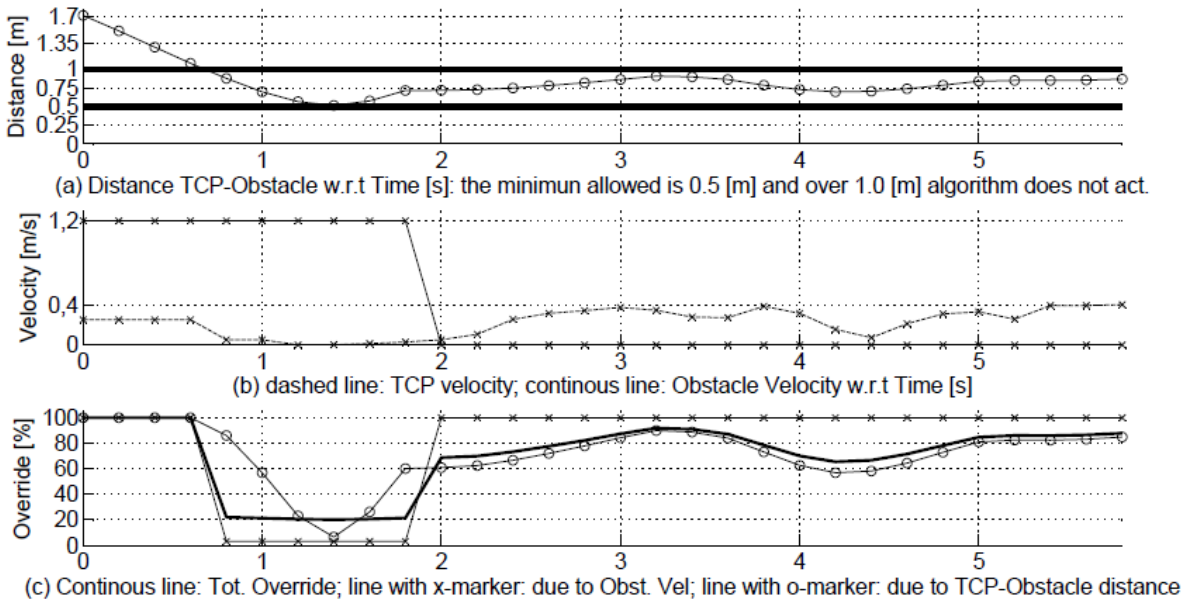
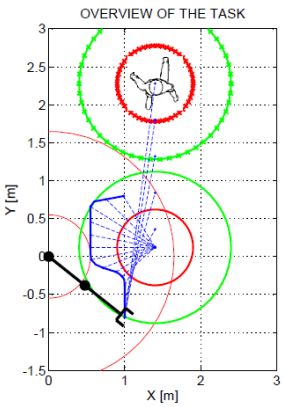


Fig. 34 Test 3

Test 4



The robot **TCP** and the human operator are moving in opposite directions.

The distance between the robot and the obstacle decreases fast.

When the distance decreases below a certain limit the correction due to the algorithm starts to correct and deform the original trajectory. Note that the robot does not stop the motion and is able to avoid the obstacle.

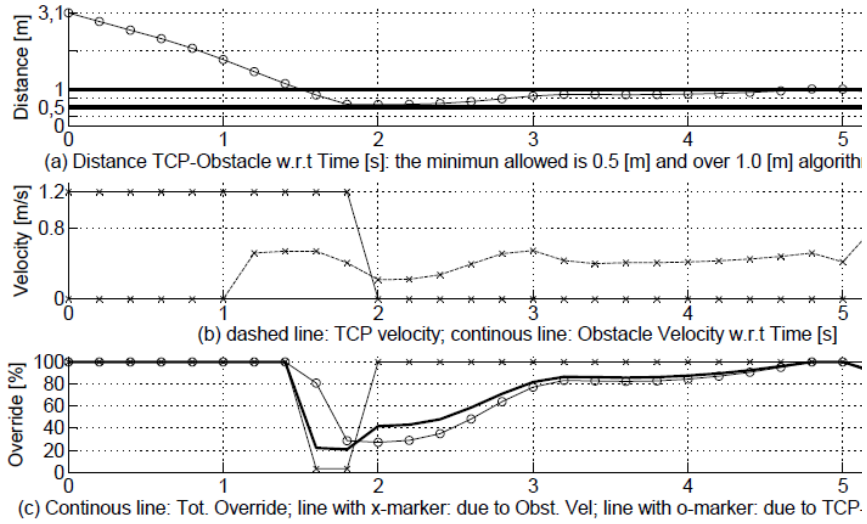


Fig. 35 Test 4

6.2.3 Results

6.2.3.1 Communication performance

- The communication between the **real-time PC (CPU-3)** and the **c4gopen (CPU-5)** is stable at **1 ms;**
- The communication among the net-nodes **CPU-3/4** and the two **PLCs (CPU-1/2)** is stable at **0.8 ms;**
- The communication among the two PLCs **(CPU-1/2)** and the **SafeCPU** is equal to powerlink frame-rate that is **0.8 ms;**
- The communication among the two PLCs **(CPU-1/2)** and the **SafeCPU** is equal to powerlink frame-rate that is **0.8 ms;**
- The communication between the user-program in the controller (**CPU-6)** and the two **PLCs (CPU-1/2)** is stable at **50 ms;**

6.2.3.1 Safety-circuit reaction

- Opening time of the ***Safe-Relé*** requires **50 ms** (measured from command generated from the SafeCPU and the circuit open event);
- Execution of the algorithm running in the **SafeCPU requires less than 10 microseconds;**
- The communication among the two PLCs **(CPU-1/2)** and the **SafeCPU** is equal to powerlink frame-rate that is **0.8 ms;**
- Safe Reaction times:

  1. Incoherence between data coming from **CPU-3** and **CPU-6** requires a reaction time equal to **20 ms** (max communication time among the CPU6 and the CPU-1/2) plus the communication between the PLCs and the **SafeCPU** equal to **0.8 ms, plus the SafeCPU** execution time plus **0.010 ms**, plus **50 ms** that is the time spent from electro-mechanical relé to be opened

     ➔ **total about 70 ms**

  2. Incoherence in target generation is equal to **c4gopen communication time (1ms)** plus calculation time (less than 20 microseconds) plus communication time between **CPU-3** and the two PLCs **(0.8 ms),** plus **the** communication time among the **PLCs** and the **SafeCPU (0.8 ms) plus the safe-relé reaction time (50 ms)**

     ➔ **total about 50 ms**

  3. Incoherence in following error calculation is equal to **c4gopen communication time (1ms)** plus communication time between **CPU-3** and the two PLCs **(0.8 ms),** plus **the** communication time among the **PLCs** and the **SafeCPU (0.8 ms) more the safe relé reaction time (50 ms)**

     ➔ **total about 50 ms**

4. Identification that the human is in the DANGER-AREA requires: tracking sensor data elaboration (DIFFERENT ON THE BASIS OF THE HW/SW CHOSEN), plus communication time among the **CPU-4** and the **CPU-3/6**, plus the elaboration data time equal to the time calculated in point (1).

6.2.3.2 Collision Avoidance Algorithm performance

- As shown in previous paragraph, the collision avoidance strategies allow a safe modification of the trajectory also when human is moving fast in the collaborative area

- Fast tracking of humans is still the bottleneck of the system

- If the robot is moving slower than **250 mm/s <u>(AS STANDARD IMPOSES IN COLLABORATIVE WORKSPACE)</u>** the distance between the nodes of the grid can be around **25 cm**

- If the robot is moving faster than **250 mm/s** the distance between the nodes of the grid can be around **10 cm.**
  The two drawbacks of this situation are:

  - **High number of pass-through-points, and memory problem with the robot controller. Advance solution (streaming of data towards the robot during the movement) should solve this problem.**
  - **The "FLY" option of the robot controller tries to maintain the velocity constant during the execution of the path. If the nodes are too close one to each other, the robot has no space and time to accelerate and reach the target velocity**

  .

- If robot tool speed is faster than 1 m/s, the ***safety-net*** approaches **DOES NOT GUARANTEE** the safety of humans inside the workspace. The delay equal to **50 ms imposed by socket communication between the user program in the robot controller and the PC that detects the human motion becomes critical.**

- Correct modelling and dislocation of the "**SAFE-AREA"** to avoid clamping and the "**WARNING-AREA**" are fundamental especially when the robot is working on the machines.

Acknowledgment

Prof. Giovanni Legnani for the outfit-information on the ISO10218 standard.