Dynamic safety in collaborative robot workspaces through a network of devices fulfilling functional safety requirements

Federico Vicentini, Nicola Pedrocchi, Matteo Giussani, Lorenzo Molinari Tosatti
National Research Council, Institute of Industrial Technologies and Automation, Italy

Abstract

Safety in human-robot cooperation is a non-optional property of any modern robot manufacturing application. All actions and settings, in terms of hardware and software, are intended to prevent hazards. In the domain of workspace sharing, with Speed and Separation Monitoring (SSM) modes, some safety-critical components can rely on safety-rated localization of users and safe computation of kinematical entities. Common to sensing and algorithms, some core functions are likely to be distributed in a network of devices, not necessarily sharing a safety-rated fieldbus. On the computation side, floating-point algorithms are never performed in state-of-the-art applica-tions compliant with current regulation. The purpose of the paper is to discuss the use of unsafe devices and protocols for safety functions considering functional safety at system level. On top of that, realistic conditions in networking are directly affecting one of the principal parameters of safety, which is protection distances. Some experimental data are reported in the case of dynamic SSM, based on trajectory-dependent safeguarding volumes computed at runtime.

1 Introduction

Robot-based manufacturing is experiencing increasing acceptance and commitment in hybrid production systems[1, 2], where human-robot cooperation (HRC) represents a substantial technological assets. While occupational stress in HRC is still under investigation [3, 4], safety protection stands as the topmost property of collaborative workspaces in the sense of ISO 10218-2[5]. Safety of workers and devices is a primary concern for any robotic system, which is required to enable the property of *operational safety*: robotic applications have to provide the set of actions, settings, counter measures and behaviors by all players (systems, users and operators) that is intended to prevent hazards. Although the adoption of some new sensing capabilities (e.g. vision-based user detection, wearable devices) could significantly boost the design of safe operations, the integration of both control strategies and sensing capabilities in safety-certified/certifiable components could be investment-intensive and/or hardly sustainable over time in maintenance or upgrading phases.

In such scenario, the problem is two-fold: there is a urgent need of (i) safety-rated measurement of distances between robots and workers and (ii) it is desirable that safe emergencies are triggered by variable thresholds whose safest minima are trajectory-dependent. This class of problems applies to the implementation of protection buffers according to Speed and Separation Monitoring (SSM) procedures, as introduced by the *draft* ISO/TS 15066 [6] that derives from previous ISO 13855[7] the definition of safety distances. Static implementations of SSM are based on the detection of the logical (on/off) condition of violation of a predefined safeguarded volume (see left side of Fig. 1), regardless where the robot is actually moving inside such protected workspace. Common commercial solutions [8] provide, for instance, 2.5D offline-validated volumes for such purpose. On the contrary, *dynamic* SSM (see right side of Fig. 1) require the online computation of changing geometries between potentially interfering objects.

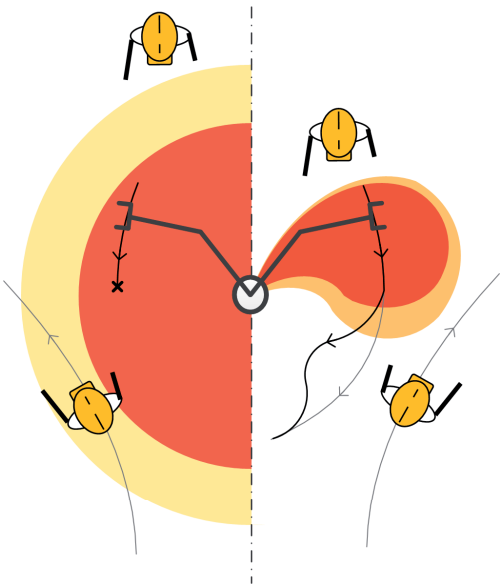

Figure 1: static SSM (left) with buffer distance to take into account reaction timing and braking distances, and dynamic SSM (right) with trajectory-dependent safety distances, including possible evasive trajectories in case of potential collisions.

ISBN 978-3-8007-3601-0

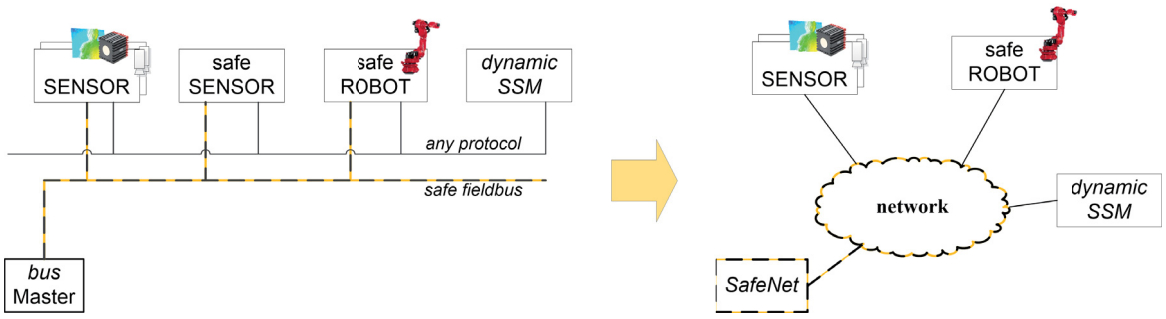

Figure 2: (left) template architecture with safety-related parts of the robot system based on safety-rated fieldbuses; (right) networking of distributed control systems with safety functions supervised by SafeNet.

The set of candidate industrially-safe (*i.e.* IEC61508 SIL3 classified) sensors for worker tracking could ultimately turn out to be limited in number or lacking some of the required functionalities. For instance, dynamic definition and computation of safe areas cannot currently be accomplished without a reboot of safety-related part of control systems (SRP/CSs), cannot use a continuous value from sensors as a thresholds. As a consequence, over-conservative safety distances required around the robot workspace actually could increase the size of the layout[9]. In addition, static SSM with safe sensors does not consider some criticalities in the (problematic) contribution to SSM distances introduced by robot dynamics (*e.g.* stopping distances travelled after that a safety trigger is commanded [10, 9]). Such effects result in an enlargement of the minimum distance considered to be the normative safe metric for both stopping or avoiding collisions.

Under this perspective, a methodology that does not restrict safe applications to the exclusive usage of fail-safe sensing could have some potential benefits in dynamic SSM implementations (Fig. 2) . In such applications, floating-point algorithms and unsafe sensor sources or interfaces could seamlessly be used and frequently updated. Sources of information are, then, likely distributed over a network with generic protocols in place in (part of) the system, generating a non-deterministic stream of data. The methodology discussed in this work, referred as SafeNet [11], is dedicated to encapsulate such uneven streaming into a deterministic, monitored data-flow, given the availability of unsafe data (unsafe is intended as any component, protocol, data format or any combination of such elements that are not classified according to the safest ISO13489/IEC61508 performance/integrity levels). Making use of distributed data, the SafeNet nodes have to consider some architectural effect in the data marshalling process. Inaccuracies in various sources of measurement and the spatial and temporal misalignments between the various sources are, in fact, modelled and computed taking into account offline calibration errors, online tracking errors of the manipulators, online temporal misalignment between sampled poses and real poses, inaccuracies in measurements and the human-robot relative velocity, latencies.

The SafeNet methodology is not intended to per se provide safety-graded components/solutions but aims at introducing systematization in bringing unsafe components into the *functional safety* assessment process (Fig. 3) .

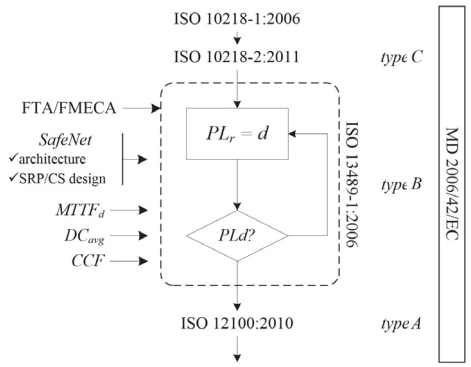

Figure 3: Core normative elements in collaborative robot safety and functional safety standards

Functional safety, in fact, is the key element of system design based on (i) well-tried components and methods and on (ii) the application of the principles of redundancy, diversity, monitoring. Through the integration of functional safety methods, the SafeNet allow the computation of dynamic SSM in some distributed CPUs, *de facto* disclosing the potential use in safe conditions of the many collision avoidance algorithms flourished over time [12, 13, 14]. Furthermore, dynamic SSM allows to properly introduce the contribution of robot dynamics in some parts of the SRP/CS.

The paper discusses first some architectural aspects related to the functional safety assessment. Then, considering the networking conditions practically applying, the effects of system latencies, inaccuracies and trajectory-dependent robot braking distances on the computation of dynamic SSM are investigated. Such computation is part of the procedures that take place on the architectural nodes dedicated to both marshalling non-deterministic data *and* performing floating-point computation.

ISBN 978-3-8007-3601-0

2 Architecture and functional safety

While SRP/CS logical units are standard-wise implemented with redundant hardware architecture, the physical layer and the transport stack do not necessarily require redundancy in order to attain a safe communication network (*black channel* approach, introduced in IEC 62280-1). Intrinsically unsafe channels (*e.g.* conventional networks, automation filedbuses) may be used as a safety-unaware medium on top of which a safety layer is added in order to connect safety tasks (Fig. 4).

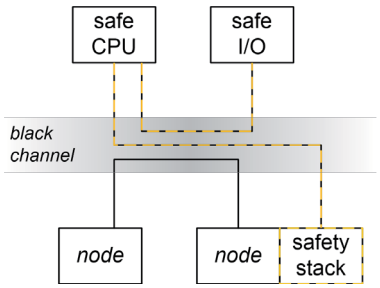

Figure 4: The safety stacks in each node implements safety-related transmission functions and IEC 61131-compliant operatons

One of the advantages of the black channel principle is to share the standard communication network (usually already available at shop floor) for both safe and unsafe (*e.g.* asynchronous diagnostics, logs) tasks and, more importantly, to allow standard networks (*e.g.* ethernet IP) compatible devices to be used in SRP/CS.
The template safety architecture of Fig. 5 reports an example of implementation of SRP/CS distributed over different nodes and making use of two different protocols as black channels.

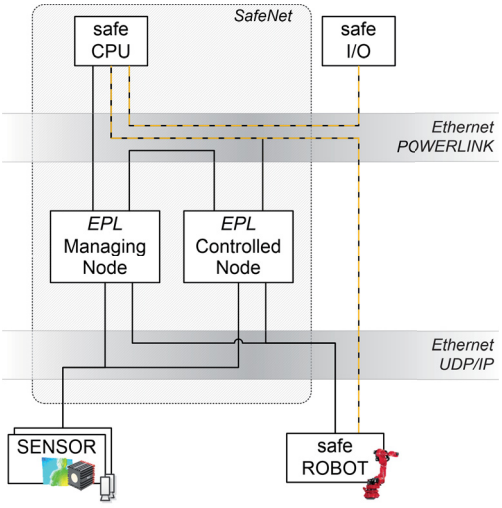

Figure 5: Double black-channel architecture for connecting general purpose unsafe devices (bottom) to CPUs. The Ethernet POWERLINK (EPL) nodes act as gateways for IP transported data to be inserted into the safety network.

The prerequisite is to ensure a non-degraded data chain to the ultimate safety function blocks (computed in safe CPUs) from any safety-relevant node of the network. Such access can be obtained either by adding a safety stack (*e.g.* openSAFETY) to the network nodes, binding the entire network to a safety automation protocol, or using some entry-points for the black channels which the safe CPUs are connected to. In the architecture of Fig. 5, Ethernet POWERLINK is used as the black channel for the application layer where safety functions and devices are integrated (safeLOGIC and I/O components by B&R Automation, Austria). Non-deterministic data are instead exchanged through a standard UDP/IP full stack on top of which an application layer enables a fail-safe methods running in real time on the nodes where critical accesses to the network sockets are implemented (*i.e.* EPL nodes). Such methods monitor and verify all possible message failures (see Fig. 6).

type of error \ measures	running number	timestamp	timeout	echo	send/recv identifier	check of data consistency	different data integrity check
unintended repetition	✓						
loss	✓		✓				
insertion	✓						
incorrect sequence	✓						
falsification						✓	
delay			✓				
mixing SR/nonSR					✓		

Figure 6: sources of error messages on IP-based black channels and implemented countermeasures.

The UDP/IP black channel is twice used for every information source (robot, sensor(s)), directly connecting such sources with both EPL nodes (see Fig. 5). As a result, the SRP/CS is distributed among 3 CPUs: the (commercially) safe CPU is in charge of performing standard safety output (*e.g.* speed reduction, emergency stop, etc). The 2 standard CPUs instead provide the connectivity and fail-safe monitoring methods (see Fig. 6), architecturally spanning over the input and the logic layers of the SRP/CS (see Fig. 7).

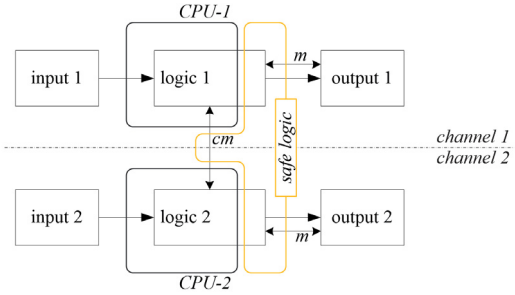

Figure 7: ISO13489-style SRP/CS of class PLd or above, given by the dual channel and cross-monitoring options (cm). Physical CPUs composing the SafeNet are highlighted in black and yellow.

ISBN 978-3-8007-3601-0

The Logic unit of the SRP/CS is therefore configured as a *1oo2* element in the sense of IEC 61508. With a full coverage of transportation errors, the probability of undetected failures is null ($\lambda_{su} = \lambda_{du} = 0$ in IEC 61508). Additionally, being the same CPUs responsible of relaying the failure information directly to the safeCPU (via EPL black channel), the failure mode is set to *safe* ($\lambda_{dd} = 0$). Therefore the contribution of failures ($\lambda_{sd} > 0$) to the main indicators of functional safety is transparent: contribution to Probability of Failures per Hour (PFH) is null and Safety-Failure Fraction (SFF) is unitary. The direct consequence is that the black channel has negligible influence on the IEC 61508 SIL assessment. Comparably, depending on the hardware implementation of the withstanding sensory equipment, the safety rate of the chain Inpu-Logic can comply with SIL2 or SIL3, depending on the redundancy of interfaces and dedicated channels (equivalently, single/dual channels in ISO 13489-1:2006 with variable DC rate). Highly available local networks could implement, for instance, the IEC 62439 Parallel Redundancy Protocol (PRP) [15] with an extra duplication of the polling nodes, eventually provided by the pair of EPL nodes (see Fig. 5).

Once the Safenet architecture allows for safe networking via general purpose entry points, the effects of such possibilities entail the operational computation of safety figures. First, the availability of computational units directly in the SRP/CS enables some floating-point capabilities for safety-critical applicatins, *e.g.* the dynamic SSM. Without kinematics only static safeguarding (predefined region violation) is possible. Second, in all SSM cases, distributed data allowed to enter the safety layer are affected in accuracy by calibration and latencies.

The following section introduces such effects in the case of variable safeguarding volumes computed at runtime to wrap either the robot or the user, on the basis of mutual distances and velocities. The net effect is an increase in the boundary of the safe region in order to take into account the uncertainty introduced by networking f unsafe components.

3 Realistic Safety Minimum Distances

The dynamic SSM principle is essentially based on the concurrent measurement of the trajectories of an obstacle and a robot. The Euclidean distance d_t between the robot R and an obstacle H changes over time t and cannot decrease below a safety threshold S without triggering low-speed modes or, if undetected, momentarily losing its operational safety status ([10]). The safety distance S is normatively computed as a function of reaction timing (of sensors, robots and humans) and velocities/accelerations. SSM conservatively considers, in fact, the worst-case deceleration and average speed along the workspace in order to assemble a safe speed multiplier for a given set of reaction times, as in [6]. Alternatively, a minimum safety distance $d_t^{safe} \leq S$ is here computed as a function of the *actual* velocity $\mathbf{v}_R$ and $\mathbf{v}_H$ vectors, resulting in a reshaping of the safety virtual envelope (red shadow in Fig. 8) depending on the current trajectories.

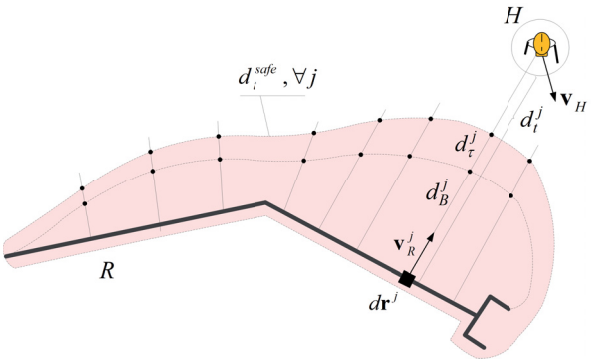

Figure 8: Actual robot (R) to obstacle (H) distance d_t, to be kept greater that $d_t^{safe} = d_\tau + d_B$, w.r.t. all elements of the robot.

For any given time frame, the safety condition is met iff

$$d_t^j \geq d_t^{safe}, \quad \forall j \tag{1}$$

considering the entire robot body. The safety threshold d_t^{safe} is composed by two main contributes that rearrange the terms of reaction timing and brake distance in [6] according to the real system conditions, preserving all the dynamics effects. In particular, the sensor and robot reaction times are reformulated considering the *total* system latency τ between the physical event (*i.e.* the instantaneous $H - R$ distance) and its actual availability at the SRP/CS (Fig. 9). Integrating the current velocity, τ produce d_τ.

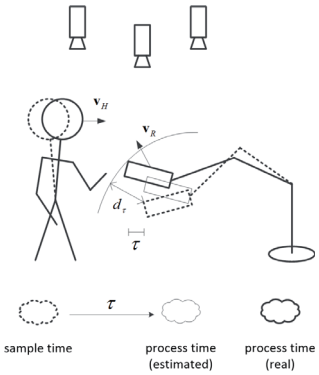

Figure 9: effect of the generalized latency τ on safety distances.

The latency τ and inaccuracies in calibration/registration has the physical equivalent in introducing an uncertainty area around the point of interest (*e.g.* the worker) which has to be further enveloped by the dynamic SSM region (see Fig. 10). In this sense Fig. 8 and Fig. 10 are computationally equivalent.

ISBN 978-3-8007-3601-0

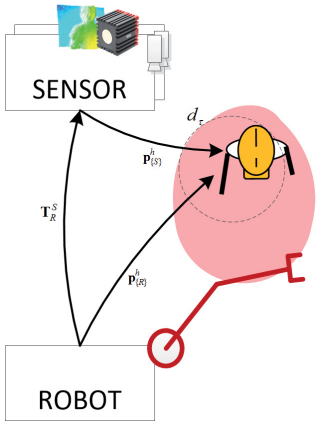

Figure 10: effect of the generalized latency τ on safety distances.

The safe stopping blind-travelling clearance d_B (braking distance) can be computed either analytically, knowing the closed-loop robot dynamical model, or empirically, sampling the braking distances in different kinematical configurations.

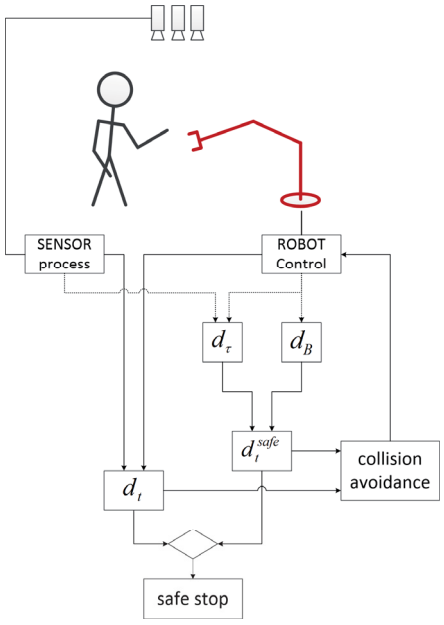

Figure 11: flowchart of dynamic SSM procedure.

As a matter of application, at any given timeframe, the condition in (1) is applied, eventually triggering stop-1 robot braking (see Fig. 11), or can be the input information for performing any kind of trajectory-dependent evasive motion in order to prevent such critical condition.

3.1 Latency-dependent Distance

Recalling that the computation of safe distances requires the calibration of sensors w.r.t. the robot reference frame that introduces a *spatial* inaccuracy in distance values d_t^j in (1), sensor and robot sources of runtime trajectories are rarely connected to the same synchronous filedbus. Euclidean geometries are, in fact, computed out of an asynchrounous streaming of data (see Fig. 5), so that all figures in Fig. 11 are affected in accuracy by

- the time offset T_{offset} between the robot and sensor CPUs timestamping;
- the jitters $\sum_k jit_k$, for each k-th channel, in data processing in case of non real time processing;
- the different rate of information processing ΔT_{proc} between the two sources.
- the different latencies in information transmission ΔT_{TX} between the two sources.

Such *temporal* inaccuracy can be evaluated in terms of equivalent overall latency

$$\tau = T_{offset} + \Delta T_{proc} + \Delta T_{TX} + \sum_k jit_k \geq 0 \quad (2)$$

that solely depends on the sensor properties and the system networking characteristics.
The latency τ then integrates the local velocity $\mathbf{v} = f(\mathbf{v}_R^j, \mathbf{v}_H, \forall j)$ for obtaining the trajectoy-dependent d_τ (Fig. 9). Such latency offset on the virtual safety envelope (Fig. 8) is therefore larger along the resultant direction of colliding velocities. Being a measure of the blindness on input data, a backward distance offset is also added, accounting for potentially undetected motion inversion.

3.2 Braking Distance

Braking distances are probably the most challenging figures to establish in safety actions because they depends on robot machine and robot control closed-loop dynamical properties. Additionally, the effects of braking (either in stop 0 or 1 modes) take place only *after* that the safety state has been triggered, which, in turn, depends on such dynamical properties to be computed. A general approach, aimed at determining a stopping maximum *time*, would either have to rely on some common background knowledge of the robot dynamics, for all manipulator, or depend on - possibly overconservative - general assumptions [9]. Both cases eventually pose substantial problems in standardization of procedures and metrics. For the present work, an approximated local-scope approach can preliminarily be based on the empirical estimation of braking *distances*, directly. In this case a look-up table can be figured out in a subset of the workspace for travel distances after a safety stop trigger, depending on the robot configuration, velocity and payload.

4 Experimental Results

An experimental setup is prepared for the execution of the routine in Fig. 11 and for estimating in advance the

ISBN 978-3-8007-3601-0

Table 1: Experimental Results: τ in different system conditions. All figures in ms. Deviation $\epsilon \in [8.2 \times 10^{-6}, 9.0 \times 10^{-6}]$. † nominal robot control frequency (A = 200 Hz, B = 66Hz); nominal sensor sample frequency (1 = 3Hz, 2 = 13Hz, 3 = 20Hz, 4 = 30Hz).

†	T^R_{proc}	T^R_{TX}	T^S_{proc}	T^S_{TX}	T_{offset}	τ
A.1	5.41±ϵ	0.015± 0.001	273.23± 0.012	0.014± 0.001	5.22	273.1
A.2	5.23±ϵ	0.019± 0.001	77.79± 0.013	0.015± 0.001	5.14	77.7
A.3	5.00±ϵ	0.023± 0.001	50.31± 0.014	0.022± 0.001	4.87	50.2
A.4	5.10±ϵ	0.021± 0.002	33.02± 0.014	0.020± 0.001	5.51	33.4
B.1	16.00±ϵ	0.015± 0.001	274.55± 0.014	0.016± 0.001	5.20	263.8
B.2	15.49±ϵ	0.015± 0.001	77.94± 0.0112	0.026± 0.002	5.46	67.9
B.3	15.86±ϵ	0.019± 0.001	51.55± 0.014	0.025± 0.002	5.34	41.0
B.4	15.41±ϵ	0.020± 0.002	32.11± 0.013	0.014± 0.002	5.20	21.9

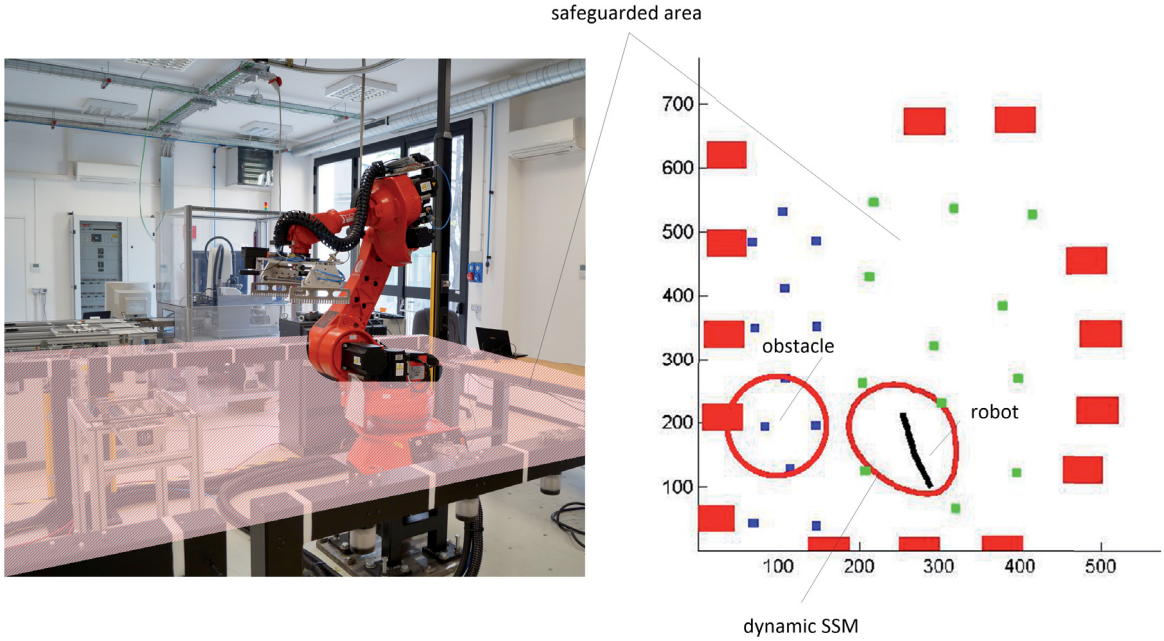

Figure 12: (left) overview of the experimental setup with a fenceless COMAU NS16 cell and MESA ToF ceiling-mounted cameras; (right) data representation with snapshot depiction of dynamic SSM region during robot motion.

trajectory-dependent figures τ and d_B required in the runtime verification of SSM condition in (1). The setup (Fig. 12-left) includes a fenceless cell with time-of-flight sensors and SafeNet set as in Fig. 7 for (i) collecting the sensors and robot data, (ii) evaluating the distances and (iii) computing the dynamic SSM conditions. The accuracy (and rate) of sensory equipment is of the utmost importance in the overall performance of SSM tasks, yet playing a primary role in possible loss of safety status [10]. The SafeNet components, together with the network diagnostics tasks, provide the unified timing measurements used to compute the τ parameter as in (2). Clock synchronization is done with a basic implementation of IEC61588 standard[16], using the Check Node as the reference clock for the round-trip delays, and obtaining a compound T_{offset}. A set of experiments, varying the nominal frequencies of sensor and robot processing, has been done and reported in Tab. 1 in order to evaluate the impact of several system configurations. The resulting d_t^{safe}, in a given instant t of the robot trajectory and along the entire robot body, provides the boundary coordinates for the safety virtual envelope in Fig. 8. Experimental values along a sample trajectory are depicted in Fig. 12-right.

5 CONCLUSIONS

The paper considers the case of safe conditions built around robots and users updated at runtime, *i.e.* a case of dynamic SSM. The concept of safeguarded workspace is reversed and optimized because it is not a priori fixed offline but is rather computed online according to current conditions, applying the safety guarding only around the operationally working point.
The safety threshold (safe minimum distance) is a function

ISBN 978-3-8007-3601-0

of the trajectories of potentially colliding bodies, the dynamical characteristics of interacting bodies and the system infrastructure. All conditions change over time, requiring the computation of extra contributions to the nominal standard-defined safety distance. The practical result is a variation in size of the virtual envelopes that represent the safeguarded space around the interacting bodies. Contributions in the computation of such augmented distances are defined, presented and experimentally evaluated. Such variably-inflating safeguarded volumes can be in turn be considered as the critical boundary for field-based collision avoidance algorithms. The execution of algorithms over distances and dynamic thresholding require kinematics computations. Floating point computation is invariably denied in safe software in the form of general purpose computation. With the purpose of keeping such solution flexible, the computations can be redundantly executed in non safe components, then safely checked for consistency within a SRP/CS. Computation, monitoring and unsafe data safeguarding are implemented by an ad-hoc architecture called SafeNet.

ACKNOWLEDGMENT

This work has been partially suported by CNR Flagship Program “Factory of the Future”, FdF-SP1-T3.1, Project FACTOry Technologies for HUMans Safety.

References

[1] J. Krüger, T. Lien, and A. Verl, “Cooperation of human and machines in assembly lines,” *CIRP Annals - Manufacturing Technology*, vol. 58, no. 2, pp. 628 – 646, 2009.

[2] L. Wang, “Collaborative robot monitoring and control for enhanced sustainability,” *The International Journal of Advanced Manufacturing Technology*, pp. 1–13, 2013.

[3] T. Arai, R. Kato, and M. Fujita, “Assessment of operator stress induced by robot collaboration in assembly,” *{CIRP} Annals - Manufacturing Technology*, vol. 59, no. 1, pp. 5 – 8, 2010.

[4] D. Kulić and E. Croft, “Pre-collision safety strategies for human-robot interaction,” *Autonomous Robots*, vol. 22, no. 2, pp. 149–164, 2007.

[5] ISO, *ISO 10218-2:2011: Robots and robotic devices – Safety requirements for industrial robots – Part 2: Robot systems and integration*. Geneva, Switzerland: International Organization for Standardization, 2011.

[6] ——, *ISO/TS 15066:2011: Robots and robotic devices – Collaborative robots*. Geneva, Switzerland: International Organization for Standardization, 2011.

[7] ——, *ISO 13855:2010: Safety of machinery – Positioning of safeguards with respect to the approach speeds of parts of the human body*. Geneva, Switzerland: International Organization for Standardization, 2010.

[8] Pilz GmbH & Co. (2012) Safe camera system SafetyEYE. [Online]. Available: https://www.pilz.com/en-INT/eshop/00014000337042/SafetyEYE-Safe-camera-system

[9] T. Salmi, O. Väätäinen, T. Malm, J. Montonen, and I. Marstio, “Meeting new challenges and possibilities with modern robot safety technologies,” in *Enabling Manufacturing Competitiveness and Economic Sustainability*, M. F. Zaeh, Ed. Springer International Publishing, 2014, pp. 183–188.

[10] J. Marvel, “Performance metrics of speed and separation monitoring in shared workspaces,” *Automation Science and Engineering, IEEE Transactions on*, vol. 10, no. 2, pp. 405–414, April 2013.

[11] F. Vicentini, N. Pedrocchi, and L. M. Tosatti, “Safenet of unsafe devices - extending the robot safety in collaborative workspaces.” in *ICINCO (2)*, J.-L. Ferrier, O. Y. Gusikhin, K. Madani, and J. Z. Sasiadek, Eds. SciTePress, 2013, pp. 276–283.

[12] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, Mar 1985, pp. 500–505.

[13] N. Pedrocchi, M. Malosio, and L. Tosatti, “Safe obstacle avoidance for industrial robot working without fences,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, Oct 2009, pp. 3435–3440.

[14] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, “A depth space approach to human-robot collision avoidance,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 338–345.

[15] H. Kirrmann, M. Hansson, and P. Muri, “Iec 62439 prp: Bumpless recovery for highly available, hard real-time industrial networks,” in *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*, 2007, pp. 1396–1399.

[16] “IEEE standard for a precision clock synchronization protocol for networked measurement and control systems,” *IEC 61588:2009(E)*, pp. C1–274, Feb 2009.

ISBN 978-3-8007-3601-0