2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

A constraint-based strategy for task-consistent safe human-robot interaction

Nicola Maria Ceriani, Andrea Maria Zanchettin, Paolo Rocco, Andreas Stolt, Anders Robertsson

Abstract— **Tight human-robot interaction and collaboration will characterize future robot tasks. Robot working environments will be increasingly unstructured, as safety barriers will be removed to allow a continuous cooperation of robotic and human workers. Such a working scenario calls for novel safety systems capable of combining productivity with workers' safety. In this paper, a method for the definition of a task-consistent collision avoidance safety strategy is presented. A classification of task constraints based on relevance for task completion is introduced. Control of task constraints enforcement is performed through a state machine. A template for such state machine is proposed. Experimental validation of the proposed safety system on a dual-arm industrial robot prototype is presented.**

I. INTRODUCTION

Capability to manage tight interaction and collaboration with humans is one of the most important requirements of the robot in the future. In many robot applications human workers will not only occasionally take part in robot task execution, but human and robotic work will be deeply interlaced, with the continuous alternation of automated and human actions. Such cooperation is a key to increase robot flexibility, the lack of which is still impeding robots penetration in those industry areas where production setups are frequently changed. In fact, robot precision, repeatability, speed and strength complement human ability to adapt and learn, opening new perspectives of productivity.

The different problems arising for human and robotic workers in the future working scenario call for new safety systems capable of combining safety and productivity. Different approaches have been used and contribute to ensure safe HRI. Adoption of robot design criteria for intrinsic safety is an important element towards the achievement of safe HRI. In [1] the overall design of an HRI-fit robot is presented: robot lightness is preferred to positioning accuracy which is hardly exploitable in unstructured environments, torque sensing allows effective collision detection and force/torque control. In [2] injury risk related to impact with the same robot is investigated. Injury criteria are proposed and blunt impact experiments are carried out on a dummy body. Moreover sharp tool impact is experimentally evaluated. In [3] a variable stiffness actuator concept is proposed as an element for intrinsically safe robots design, guaranteeing both performance and defined risk level. In [4] another safety-oriented actuation approach is proposed, decreasing robot effective inertia through splitting each joint actuation between two actuators: a low-frequency, high-inertia actuator located at the robot base and a high-frequency, low-inertia actuator located on the joint. Together with inherently safe design of manipulators to make possible collisions harmless, work has been done on the prevention of collisions. In [5] the potential field approach is introduced, allowing to derive repulsive forces from danger assessment. In [6] a danger index measuring potential impacts with humans is introduced. Moreover an algorithm is proposed to compute a virtual force moving the robot away from the human. In [7] a framework capable of task-consistent obstacle avoidance is proposed. To this intent evasive motions are performed in the null-space of the task.

The motivation of this paper is to contribute to the definition of a method for designing task-consistent collision avoidance strategies. In order to productively combine task execution with achievement of safety, constraints affecting the task should be explicitly taken into account by the safety strategy. The coexistence of task execution and safety actions can be based on the exploitation of unconstrained velocities to perform evasive motions: for this, a classification of constraints based on task relevance is proposed. Constraint relevance for task completion is similar to task priority introduced in [8]. Such approach has been further developed in various works as [9], where a framework is proposed for managing multiple constraints in a task-priority strategy for redundant robots, or more recently in [10], where hard constraints on joint variables are taken into account during the execution of multiple prioritized tasks. Nonetheless, in this work constraints composing the task are consistent and no prioritization is necessary. Task relevance classification supports the definition of a task-consistent safety strategy exploiting possible constraints redundancy or suspending task execution while preserving final task completion.

A template state machine based on the introduced constraints classification is proposed for control of constraints enforcement. The state machine structure defines the different phases of the safety strategy, which imply relaxation of different constraints. A danger assessment known to the state machine is used to define the current state and consequently the enforced constraints, which in turn are used to define the appropriate null-space projector for evasive velocities. The proposed safety system is applied on a dual-arm industrial

The research leading to these results has received funding from the European Community Seventh Framework Programme FP7/2007-2013 - Challenge 2 - Cognitive Systems, Interaction, Robotics - under grant agreement No 230902 - ROSETTA.

N.M. Ceriani, A.M. Zanchettin and P. Rocco are with Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza L. Da Vinci 32, 20133, Milano, Italy (email: nceriani@elet.polimi.it; andreamaria.zanchettin@polimi.it; rocco@elet.polimi.it).

A. Stolt and A. Robertsson are with the Department of Automatic Control, Lund University, PO Box 118, SE-221 00 Lund, Sweden (email: {andreas.stolt, anders.robertsson}@control.lth.se).

robot prototype and experimentally validated on an assembly task. The remainder of this work is organized as follows. In Section II a skill constraints classification is proposed and two examples clarify its application. In Section III a state machine template for skill constraints enforcement control is introduced, which implements the skill constraints-consistent safety strategy. In Section IV the overall control system is analyzed and Section V describes its application to an assembly task performed by a dual-arm industrial robot prototype.

II. CONSTRAINTS CLASSIFICATION

Industrial robot controllers normally express robotic manipulator tasks by means of instantaneous constraints which define the end effector translational and rotational velocities. During task execution, such velocities are used to determine joint speeds by means of kinematic inversion. As tasks may be inherently redundant and are commonly composed also by non-instantaneous constraints, e.g. trajectory endpoints to be reached, some of the instantaneous constraints imposed by the robot controller may be unnecessary for task completion. In the following, a classification of instantaneous constraints based on three levels of significance for task execution is presented. Such a classification is inspired by Mason's natural/artificial constraints theory [11] but introduces a third category of classification and modifies the first two ones. From now on, the term "skill" will be adopted to refer to the single actions performed by the robot, while the term "task" will be used to refer to the complete operation performed by the robot, hence composed of different skills.

A. Hard constraints

This class comprises natural constraints [11] and the subset of artificial constraints [11] whose relaxation would cause the disruption of the skill. Let's consider for example a robot applying glue on a surface: while following the glueing profile, the glueing tool should be kept perpendicular to the surface and with the tooltip at a certain distance from that. Modification of translational velocities lying in the surface tangent plane and of rotational velocities around surface tangent axes would cause a wrong glueing profile and thus the skill disruption. Alteration of velocity perpendicular to the surface in the surface direction would instead cause reaction forces due to the tool-surface contact. All the mentioned velocities, which would be either classified as natural or artificial constraints, belong to the hard constraints class.

B. Skill constraints

The second group includes constraints used to specify the skill which can be relaxed without causing its disruption. Such relaxation would instead cause the temporary suspension of the skill, which can then be resumed and completed. Let's consider for example a positioning skill during which the robot has to move between two points in free space. The relaxation of constraints on translational velocities would cause a deviation from the planned trajectory, that could then be resumed. Translational velocities are therefore skill-constrained. Compared to Mason's classification, this group would be included in the artificial constraints.

C. Soft constraints

In order to control a manipulator, industrial robot controllers define an instantaneous constraint for each degree of freedom of the robot. In case redundancy is present in a skill, constraints corresponding to redundancy can be relaxed without any consequence on successful skill completion. This is the case for example of rotational velocity around the tool axis in the above-mentioned glueing task. Such constraints are defined as "soft constraints". This class of constraints is complementary to the first two, as it comprises the Cartesian velocities that do not take part in any of the first two classes.

D. Formalization of the constraints classification

In order to formalize the classification of constraints, the selection vectors $\mathbf{s}_1$, $\mathbf{s}_2$ and $\mathbf{s}_3$ are introduced. These vectors are used to define which type of constraint is applied to each of the three translational and rotational velocities. Given $\mathbf{v} = [v_x, v_y, v_z, v_\phi, v_\theta, v_\psi]$ the vector of translational and rotational velocities in the current task frame

$$s_{1i} = \begin{cases} 1, & \text{if } v_i \text{ is subject to a soft constraint} \\ 0, & \text{otherwise} \end{cases}$$

$$s_{2i} = \begin{cases} 1, & \text{if } v_i \text{ is subject to a skill constraint} \\ 0, & \text{otherwise} \end{cases}$$

$$s_{3i} = \begin{cases} 1, & \text{if } v_i \text{ is subject to a hard constraint} \\ 0, & \text{otherwise} \end{cases}$$

The three vectors sum up to a vector with all unitary elements, so that each Cartesian velocity is assigned a constraint.

III. A SAFETY STATE MACHINE

Evasive motions aimed at minimizing the level of danger in human-robot interaction can be executed by exploiting unconstrained velocities. Using the proper null-space projector, evasive motions can be projected in the null-space of constrained velocities, thus combining skill execution with safety functionalities.
The relaxation of instantaneous constraints for safety purposes can be based on an assessment of the danger in the human-robot interaction. For this, the danger field [12] can be used. Such measure considers the robot as the source of danger for an object in its workspace and takes into account both relative distance and velocity between the two. A scalar field called danger field is defined, defining the value of danger generated by the robot for obstacles surrounding it. The value of the danger field in the obstacle position can be used to control constraints relaxation, with a gradual relaxation for increasing danger. For this a template of a state machine to manage velocity constraints is proposed.

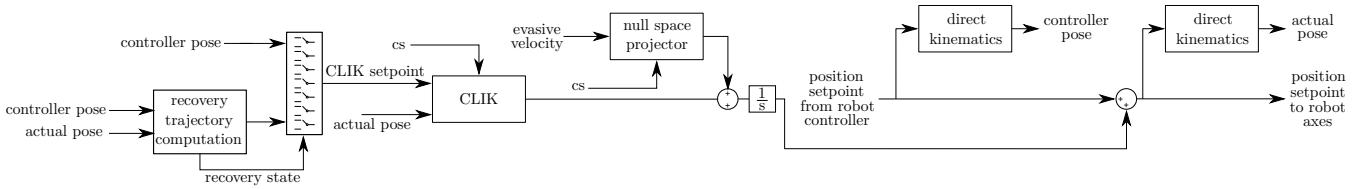


Fig. 1. The overall control scheme.

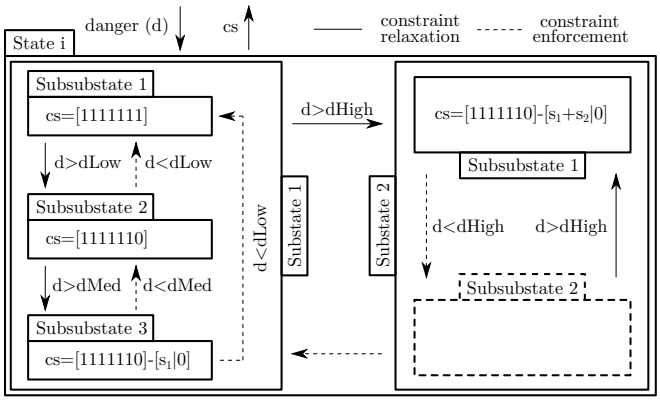


Fig. 2. The safety state machine structure.

State machine structure

The state of each instantaneous constraint (enforced/relaxed) is defined by means of the n_j-by-1 vector **cs** (where n_j is the number of robot degrees of freedom) which is the output of the state machine:

$$cs_i = \begin{cases} 1, & \text{if the constraint is enforced} \\ 0, & \text{if the constraint is relaxed} \end{cases}$$

The first 6 elements of the vector correspond to the elements of **v** while possible further elements correspond to further instantaneous constraints that can be enforced on redundant manipulators, e.g. swivel angle velocity in a 7 degrees of freedom manipulator. A 7 elements **cs** vector is used in Fig. 2 for sake of generality. The values of the elements of **cs** depend on the current state of the state machine, which in turn depends on the danger level d, known to the state machine. Three danger thresholds are defined to determine the current state: *dLow*, *dMed*, *dHigh*. Fig. 2 represents the state machine structure, which is described in the following.

- **State i**
  The i-th state of the state machine corresponds to the i-th skill composing the task.
  - **Sub-state 1**
    Sub-state 1 corresponds to an interaction condition where danger is below *dHigh* and the skill can therefore be executed.
    - **Sub-sub-state 1** $d < dLow$ When danger is below the minimum threshold no constraint is relaxed and therefore no safety action is performed.
    - **Sub-sub-state 2** $dLow \leq d < dMed$ As danger increases over the first threshold, possible robot kinematic redundancy is exploited. Safety motions have thus no consequences on the end effector position and velocity.
    - **Sub-sub-state 3** $dMed \leq d < dHigh$ If danger is larger than the second threshold, soft constraints are relaxed.
  - **Sub-state 2**
    When danger is larger than *dHigh* skill execution is suspended. The state machine remains in sub-sub-state 2 until danger falls below *dHigh* and the robot can resume skill execution
    - **Sub-sub-state 1** $d \geq dHigh$ When danger is larger than *dHigh* skill constraints are relaxed.
    - **Sub-sub-state 2** $d < dHigh$ If danger falls below the high threshold the robot pose is recovered to resume skill execution.

IV. OVERALL CONTROL SYSTEM

Control of enforced constraints is supposed to be performed through modification of position setpoints sent by the industrial controller to the robot axes. Setpoints modifications are computed by the control system.

A. Evasive motions

1) Computation of evasive motions: In [12] a method is presented to compute the value of the danger field generated by the link of a robot. Moreover, a vector field is derived from the danger field gradient. $\mathbf{CDF}(\mathbf{r})$ is defined as a vector anchored in the obstacle position $\mathbf{r}$ pointing in the direction of the danger field gradient, with modulus the value of danger generated by the i-th link in that point. This vector quantity can be interpreted as a virtual repulsive force pushing the robot in the direction of maximum decrease of danger [13]. Given a generic sensor system able to detect n obstacles surrounding the robot, a cumulative virtual repulsive force for the obstacles with respect to the i-th robot link can be computed as:

$$\mathbf{CDF}_i = \sum_{j=1}^{n} CDF(\mathbf{r_j}) \frac{\nabla CDF(\mathbf{r_j})}{\|\nabla CDF(\mathbf{r_j})\|}$$

As the modulus of $\mathbf{CDF}_i$ is the mean of all the values of danger for the n obstacles, possible high-danger obstacles may have little influence on the overall virtual repulsive force. In order to ensure evasion even when a single obstacle has high related danger, a new virtual repulsive force can be defined as:

$$\mathbf{CDF}_i^* = \frac{\mathbf{CDF}_i}{\|\mathbf{CDF}_i\|} \cdot \max_j CDF(\mathbf{r_j})$$

The direction of the force is therefore the average of the directions of the n virtual forces related to the n obstacles weighted on their modulus, while the modulus of the force is the highest of all the moduli of the n contributions. The virtual force computed for the i-th link can be applied at the link endpoints and then easily transformed into joint torques $\mathbf{T}_i = \mathbf{J}^{\mathbf{T}}_{i-1,v}(\mathbf{q}) \cdot \mathbf{CDF}^*_i + \mathbf{J}^{\mathbf{T}}_{i,v}(\mathbf{q}) \cdot \mathbf{CDF}^*_i$ where $\mathbf{J}_{i,v}$ is the linear velocity Jacobian of the i-th Denavit Hartenberg frame. As the robot is not torque controlled, evasive joint velocities can be obtained by applying the torques to a mass-damper impedance filter.

$$\dot{q}_{ev} = (\mathbf{M}s + \mathbf{D})^{-1} \cdot \mathbf{T} \quad \text{where} \quad \mathbf{T} = \sum_{i=1}^{nlink} \mathbf{T}_i$$

2) Skill null-space projection: Evasive motions can be performed by exploiting relaxed instantaneous constraints. In fact using the proper null-space projector they can be projected in the null-space of constrained velocities. In the general case of a kinematically redundant manipulator, a n_j-by-n_j augmented Jacobian $\mathbf{J}$ is considered, under the hypothesis that suitable coordinates are identified for each degree of kinematic redundancy. The null-space projector is defined as $\mathbf{N}_{cs} = \mathbf{I} - \mathbf{J}^{\dagger}_{cs}\mathbf{J}_{cs}$ where $\mathbf{J}_{cs}$ is the matrix composed of the rows of $\mathbf{J}$ corresponding to **cs** vector elements equal to 1 and $\mathbf{J}^{\dagger}_{cs}$ is its Moore-Penrose pseudoinverse.

B. Recovery trajectories

If a constraint is relaxed and then enforced again, the corresponding Cartesian coordinate of the robot has to recover the robot controller setpoint. For this, two poses are computed through direct kinematics: the first is the one set by the robot controller, while the second one is the actual pose, differing from the first one by the offsets determined by safety actions. A recovery trajectory from the current value to the reference value is therefore computed. As the controller setpoint is a priori not constant, the trajectory has to be updated at each sample time to cope with a possibly changing goal position. For this purpose an algorithm computing a trapezoidal velocity profile from the current to the goal position, taking into account current speed has been implemented. The activation of a recovery trajectory is based on algorithm 1. As already said, evasive velocities are projected in the null-space of constrained velocities. However, a CLIK algorithm is used to compensate for the drift caused on such velocities by linearization approximations. The setpoint for the CLIK will alternatively be the controller reference or the recovery trajectory depending on whether such trajectory is being executed or not. This strategy is formalized in algorithm 2.

Algorithm 1

```
for i=1:n_j do
  if cs(i,t)=1 and cs(i,t-1)=0 then
    recovery state(i)=1
  else if cs(i,t)=0 and cs(i,t-1)=1 then
    recovery state(i)=0
  end if
  if recovery state(i)=1 then
    if recovery complete(i)=1 then
      recovery state(i)=0
    end if
  end if
end for
```

Algorithm 2

```
for i=1:n_j do
  if cs(i)=1 then
    if recovery state(i)=1 then
      CLIK reference(i)=recovery trajectory(i)
    else
      CLIK reference(i)=controller pose(i)
    end if
    CLIK loop(i)=close
  else
    CLIK loop(i)=open
  end if
end for
```

V. EXPERIMENTAL VALIDATION ON AN INDUSTRIAL ROBOT

The safety system presented has been tested on the ABB FRIDA prototype, a lightweight robot composed of two 7-dof arms. The control architecture presented in Section III and IV has been implemented using the External Control (from now on ExtCtrl) interface developed by Lund University [14]. This interface allows the modification of high-level joint references computed by the robot main controller by means of an external PC. Modified references are then sent to the axis controller which transforms them into motor references. Desired control laws can be implemented on the external PC using Simulink™environment, which allows also the integration of external sensors. Such an architecture allows therefore to combine the industrial controller RAPID programming language and trajectory planning functionalities for robot programming with the use of additional externally developed sensor-based functionalities.

A. Sensor system

The robot right arm is equipped with a distributed distance sensor based on Sharp GP2Y0A21YK infrared proximity sensors. This sensor range is 10-80 cm with an update frequency of approximately 25 Hz. The upper arm is covered with a shell housing 16 sensors while a single sensor is mounted on the robot elbow, as shown in Fig. 3. Sensors placement has been defined according to the danger minimization method presented in [15]. Sensors measurements are acquired by the external PC and used to compute the danger field for the arm according to Section IV.

B. RAPID programming of skills

Skills programmed with the industrial controller language do not include information on constraints according to the classification presented in Section II. Such information is

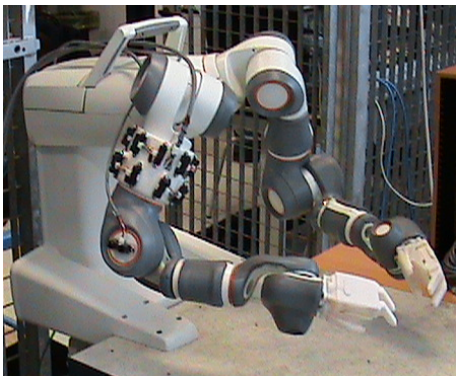


Fig. 3. The sensor system mounted on ABB Frida.

instead contained in the state machine, which has a state for each skill. In order to have the proper state machine state, say the i-th, running simultaneously with the robot instructions constituting the i-th skill, a synchronization protocol has been implemented. At the end of a program section constituting a skill, a RAPID function is called which triggers the activation of a recovery trajectory for all coordinates and the transition to the next state on the ExtCtrl side.

C. Skill suspension

As explained in Section III, when danger exceeds *dHigh*, skill execution is suspended. The value of danger is computed based on sensors measurements, which are available to the external PC. However, as the skill execution suspension is performed on the industrial controller side, the external PC is periodically asked if task suspension is needed, i.e. if danger is exceeding *dHigh*. In this case, a RAPID function performs the following actions:

1) triggers safety state machine transition to sub-state 2, sub-sub-state 1;
2) starts waiting for the acknowledgement that danger has fallen below *dHigh*.

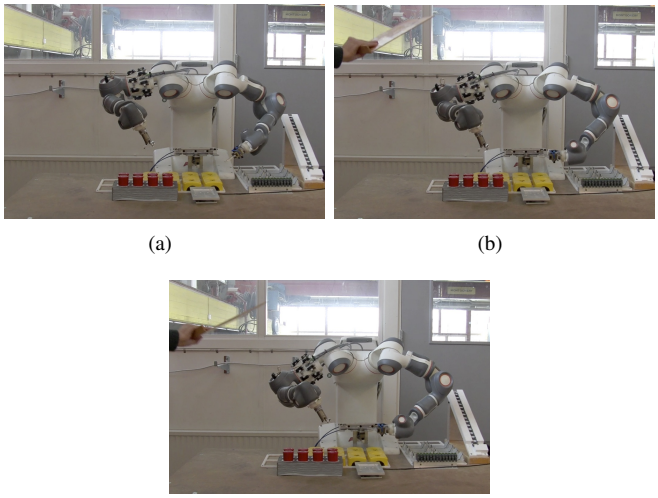
On ExtCtrl side, the drop of danger below *dHigh* triggers the transition of the safety state machine to sub-state 2, sub-sub-state 2 and consequently the execution of a recovery trajectory. When recovery has been completed the execution of the skill on the industrial controller side is resumed.

D. Assembly task

The assembly task developed in [16] has been used as the testbed of the safety system. During such task an emergency stop button is assembled through three main phases: button insertion in its case (a peg-in-hole skill), nut screwing on the button thread and finally the snap-fit assembly of the electronic component of the button. Position controlled and force controlled skills compose the task: the safety system has been integrated with the former ones. In the following, obstacle evasion during the execution of two of the skills composing the task is analyzed.

1) Free space movement: The first skill considered is the right arm movement to approach the emergency button. As this skill consists in changing the TCP position, the constraints applied to translational velocities are classified as skill constraints. Constraints applied to rotational velocities are instead classified as soft constraints. In this case, none of the constraints are classified as hard.

In Fig. 4 three snapshots of the evasion are shown. As the obstacle is detected by the sensor system, the robot evades exploiting its kinematic redundancy and therefore changing the elbow position. When danger increases, also TCP orientation is relaxed, allowing a more effective evasion. Fig. 5 shows the modification of the actual pose with respect to the target pose determined by the controller during evasion and the recovery trajectory: as danger falls below *dMed*, recovery trajectory is activated for orientation. Afterwards, as danger falls below *dLow*, recovery trajectory for elbow position is also activated.



(a) (b) (c)

Fig. 4. The three phases of collision avoidance during free space movement.

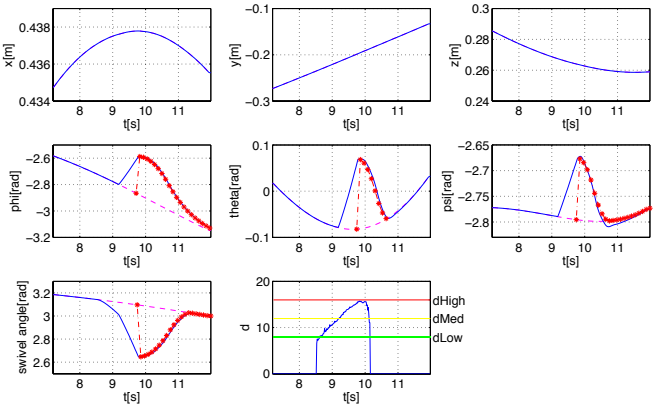
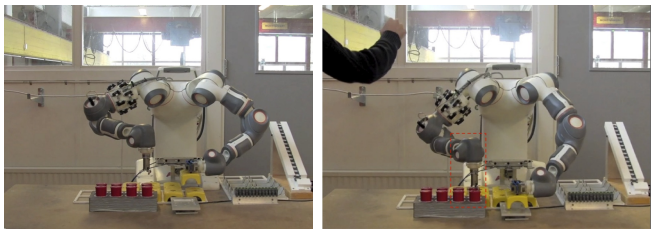
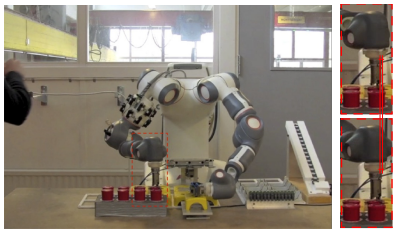


Fig. 5. Modification and recovery of robot controller setpoint during evasion. Top-right plot shows danger value.

2) Button picking: During the second skill the robot picks the emergency button from its cylindrical housing. The button and its housing compose a cylindrical pair: no physical constraint impede their relative rotation around the cylinder axis. The skill is therefore inherently redundant, as rotations around the cylinder axis have no effect on it. Constraints classification is therefore as follows: rotational velocity around the button housing axis is a soft constraint



(a) (b)



(c)

Fig. 6. The three phases of collision avoidance during button picking. Rotation in the button housing is highlighted on the right of snapshot (c).

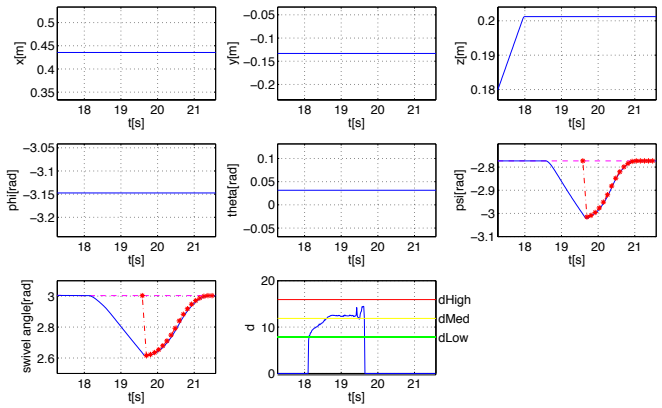


Fig. 7. Modification and recovery of robot controller setpoint during button picking. Top-right plot shows danger value.

while the other rotational and translational velocities are considered as hard constraints. In Fig. 7 constraints relaxation during button picking is shown. As in the previous skill at first the elbow deviates from the controller reference. Then also orientation around the button axis deviates from reference and, finally, a recovery trajectory is activated for orientation and elbow position. Fig. 6 shows the different phases of the evasion.

VI. CONCLUSIONS

This paper presented a task-consistent safety strategy for human-robot interaction. A three-categories constraints classification based on task relevance has been proposed. Instantaneous constraints over-constraining the skill, constraints necessary for task completion that can be relaxed during task suspension, and constraints that cannot be relaxed without causing skill disruption are identified. A state machine template has been proposed for the definition of the task-consistent safety strategy: once skills constraints are classified, the state machine can be automatically programmed, thus contributing to a tighter integration of safety strategy design and task definition. A control system for the execution of a task-consistent collision avoidance strategy has been proposed, based on interaction danger assessment proposed in [12]. The control system allows adding evasive motions to position setpoints computed by an industrial controller guaranteeing task completion together with safe interaction. The system has been experimentally validated on a dual-arm robot performing an assembly task and task consistency has been confirmed. Future work will focus on the integration of non-instantaneous and inequality constraints in the definition of the task-consistent safety strategy and on the definition of task-consistent safety strategies for dual-arm skills.

REFERENCES

[1] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimbock, and G. Hirzinger, "The dlr lightweight robot: design and control concepts for robots in human environments," *Industrial Robot: An International Journal*, vol. 34, no. 5, pp. 376–385, 2007.

[2] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Safety analysis for a human-friendly manipulator," *I. J. Social Robotics*, vol. 2, no. 3, pp. 235–252, 2010.

[3] A. Bicchi, A. Tonietti, M. Bavaro, and M. Piccigallo, "Variable stiffness actuators for fast and safe motion control," in *Robotics Research, The Eleventh International Symposium, ISRR*, 2003, pp. 527–536.

[4] M. Zinn, B. Roth, O. Khatib, and K. Salisbury, "A new actuation approach for human friendly robot design," *The International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 379–398, 2004.

[5] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, Spring 1986.

[6] D. Kulic and E. A. Croft, "Real-time safety for human-robot interaction," *Robotics and Autonomous Systems*, vol. 54, no. 1, pp. 1–12, 2006.

[7] O. Brock and O. Khatib, "Elastic strips," *The International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031–1052, 2002.

[8] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, Jul 1987.

[9] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, June, pp. 1211–1216 vol.2.

[10] F. Flacco, A. De Luca, and O. Khatib, "Prioritized multi-task motion control of redundant robots under hard joint constraints," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Algarve, Portugal, October 2012, pp. 3970–3977.

[11] M. T. Mason, "Compliance and force control for computer controlled manipulators," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 11, no. 6, pp. 418–432, June.

[12] B. Lacevic and P. Rocco, "Kinetostatic danger field - a novel safety assessment for human-robot interaction," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Oct., pp. 2169–2174.

[13] ——, "Safety-oriented control of robotic manipulators - a kinematic approach," in *18th IFAC World Congress (IFAC 2011)*, Aug./Sep. 2011.

[14] A. Blomdell, G. Bolmsjo, T. Brogardh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K. Nilsson, M. Olsson, T. Olsson, A. Robertsson, and J. W., "Extending an industrial robot controller: implementation and applications of a fast open sensor interface," *Robotics Automation Magazine, IEEE*, vol. 12, no. 3, pp. 85–94, Sept.

[15] N. M. Ceriani, G. Buizza Avanzini, A. M. Zanchettin, L. Bascetta, and P. Rocco, "Optimal placement of spots in distributed proximity sensors for safe human-robot interaction," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013.

[16] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson, "Robotic assembly using a singularity-free orientation representation based on quaternions," in *10th International IFAC Symposium on Robot Control*, Sep. 2012.