

Integration of Active and Passive Compliance Control for Safe Human-Robot Coexistence

Riccardo Schiavi, Fabrizio Flacco, and Antonio Bicchi

Abstract—In this paper we discuss the integration of active and passive approaches to robotic safety in an overall scheme for real-time manipulator control. The active control approach is based on the use of a supervisory visual system, which detects the presence and position of humans in the vicinity of the robot arm, and generates motion references. The passive control approach uses variable joint impedance which combines with velocity control to guarantee safety in worst-case conditions, i.e. unforeseen impacts. The implementation of these techniques in a 3-dof, variable impedance arm is described, and the effectiveness of their functional integration is demonstrated through experiments.

I. INTRODUCTION

Work-space sharing between human operators and robotic manipulators is often necessary during the calibration or repairing processes of industrial manipulators, and is the usual working mode for service robots. As a consequence, the real time control of robots to guarantee safe physical coexistence of human operators and robotic manipulators has become a very active research field in recent years (c.f. [4]).

To guarantee safety of humans, and avoid damage to robots, unexpected collisions should be avoided whenever possible. However, since avoiding all chances of a collision is hardly possible, and very costly in terms of performance, all measures should also be taken so that only acceptable damage can result from impacts. For different applications, the best trade-off should be studied between a certain degree of tolerable risk of impact, and the minimization of ensuing danger.

Current practice in industrial robotics is to use proximity sensors (e.g. laser beams) to detect the presence of an operator in a vicinity of the robot, and to stop the task execution (ISO 10218 [1]). Considerable work has been devoted by the robotics community to overcome this quite conservative approach, and to allow degrees of coexistence and interaction between humans and robots. To do so, active collision avoidance policies have been advocated, which are based on i) real-time detection and localization of humans in the robot workspace, and ii) reactive planning algorithms to avoid collision.

Detection and localization has been addressed by employing different sensing techniques. For instance, the work in [7], [8] uses capacitance proximity sensors to identify presence of obstacles and avoid impacts with the whole arm. More recently, advances in artificial vision have enabled

effective and economic real-time obstacle detection. In this line, Ebert et al. [9] proposed an emergency-stop approach based on an ad-hoc developed vision chip; Kuhn et al. ([10]) used multiple cameras to detect obstacles in the workspace; Iossifidis [12] use a stereo camera system able to detect and avoid obstacles on white tables.

Many reactive planning methods are based on the idea of repulsive potential fields originated in [13]. In these methods, repulsive forces are generated in the robot operational space. Erdmann [14] introduced the idea of using repulsive forces in configuration-space (\mathcal{C} -Space), generated in the vicinity of \mathcal{C} -space obstacles (\mathcal{C} -Obstacle). Rimon, in his Ph.D thesis [15] proposed a method to obtain directly the torque for obstacle avoidance in a single step adopting the gradient of a navigation function. Unfortunately, most obstacle avoidance methods are often unable to accommodate for time varying environments, because of the relatively high computational requirements.

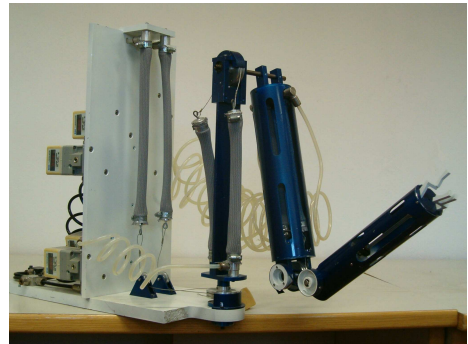


Fig. 1. Unipi SofArm: manipulator used to perform the experimental results.

The literature on passive safety is also rather extensive. The straightforward approach to obtain a safe robot consists in designing lightweight arm structures and controlling them to move slowly enough. However, these requirements often conflict with requirements on accuracy and promptness of response. Several researchers have proposed to use robots with variable compliance at their joints, so as to adapt to different tasks and deliver performance while guaranteeing safety. Previous work in our own group [2] has championed the design of variable stiffness actuators, and has shown that optimal performance under safety constraints can be translated in a control paradigm described as “stiff and slow, fast and soft”. In [6], a computationally efficient, suboptimal approximation to the optimal solution of the “safe brachistochrone” problem underpinning this paradigm was

This work was not supported by any organization
Interdepartmental Research Center “E. Piaggio”, Faculty of Engineering, University of Pisa, via Diotisalvi 2, 56126 Pisa, Italy
riccardo.schiavi@ing.unipi.it

proposed, which can be expressed by the formula

$$f(\sigma, v) = K_\sigma, \quad (1)$$

where σ is the joint stiffness, v is the joint velocity, and K_σ is a constant depending on the link and rotor inertias, the arm configuration and parameters, and expressing the acceptable level of injury risk.

The aim of this work is to implement a control method able to integrate active and passive control policies to improve safety for robots that must coexist with humans. The idea is to combine active methods for human detection and localization, with the possibility of setting different levels of passive safety, so as to make the robot aware of the situation and maximize the overall performance by actively adapting to it. The paper is organized as follows: in section II we describe a computationally efficient method to construct the \mathcal{C} -Obstacle in real time from depth map information. Section III we illustrate how repulsive forces are generated on the basis of the \mathcal{C} -Obstacle corresponding to detected human presence in the workspace. Section IV introduces the integration of information on human presence and localization, with the control of variable stiffness actuators in the robot arm. Finally, section and section reports simulation and experimental results obtained by implementing the proposed methods.

II. \mathcal{C} -Obstacle MAPPING

In this section the method adopted to detect the collision configurations of the manipulator is presented. Traditional collision detector works in a three dimensional Euclidean space using a 3D model for manipulator and obstacles. The 3D geometrical techniques adopted to detect the collision configurations require high computational capabilities. To be able to work in real time usually rather rough approximations have to be imposed on the 3D shapes (cf. e.g. [16]).

Our approach consider the camera Image Plane (IP) to detect the obstacles. The novel of this approach is to reduce the collision detection to a comparison between matrix elements.

A. Obstacle and Manipulator Depth Map

Consider the projection on the Image Plane (IP) of manipulator and detected obstacles. To detect a collision configuration $\hat{\mathbf{q}}$ a necessary condition ([11]) is to have an intersection of the two projections. A more restrictive condition can be obtained considering the depth maps of the current obstacles and manipulator configuration as shown in Fig. 2.

A Depth Map is a 2.5 coordinate system (x, y, d) where the coordinate x, y identify a pixel on the image plane, and the value d is the positive depth of the object on the projection ray of the pixel x, y . Using stereo vision methods or special depth sensors such as PMD cameras (PMD[vision] Camera), 2D scan lasers (e.g. SICK Laser Rangefinder with laser pant/tilt unit) or integrated stereo vision systems (Videre design STOC, TXYZ DeepSea, ...) the object depth map (O_{dm}) can be obtained in real time.

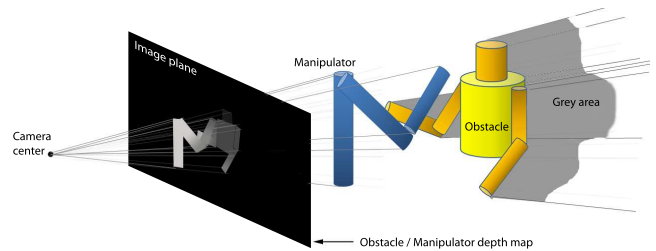


Fig. 2. By projecting the manipulator points in the IP the collision is evaluated comparing the obstacle and the manipulator depth maps.

The points of the manipulator in its own reference frame can be easily computed by using direct kinematics. To transform these coordinates on the camera frame, the extrinsic camera parameters can be used. On the camera frame, the z -coordinate of a point represents its depth. Using the intrinsic camera parameters the points can be projected in the IP obtaining the manipulator depth map ($M_{dm}(\mathbf{q})$). For this purpose a manipulator model is used, which can be obtained by using 3D CAD programs or by approximations combining elementary shapes such as cylinders and spheres. This does not affect the system performance, because the M_{dm} data set can be computed off-line whenever the relative position between manipulator and camera is fixed during the task.

It should be noticed that any other depth map on a different plane with respect to the IP will not consider all the grey points as a part of the obstacle.

An elementary cells representation of the \mathcal{C} -Space is obtained considering a discrete set for the configuration of the manipulator. A cell is included in the \mathcal{C} -Obstacle if the correspondent configuration verify the condition 2.

A manipulator configuration \mathbf{q} is on the \mathcal{C} -Obstacle if there exists a pixel (x, y) in the IP that satisfies the relation

$$M_{dm}(x, y) \geq O_{dm}(x, y) - \epsilon, \quad (2)$$

where ϵ is a safety-margin parameter.

An estimation of the minimum distance between obstacles and the manipulator is

$$\min_{(x,y) \in IP} O_{dm}(x, y) - M_{dm}(\mathbf{q})(x, y).$$

On Fig. 3 an example of \mathcal{C} -Obstacle is presented. As shown in [10], [3], the minimum distance can be used to tune the robot behaviour by limiting the maximum velocity or reducing the acceptable injury risk K_s in (1). This measures have also the effect of increasing the human feeling of safety.

III. ACTIVE COMPLIANCE: REPULSIVE \mathcal{C} -FORCE

To avoid collision against obstacles we adopt a so called \mathcal{C} -Force method. The idea is to add to torques generates by the free motion controller (about which we make no particular assumption here), a torque related to \mathcal{C} -Space distances.

Let us consider the \mathcal{C} -Obstacle as composed by elementary cells, because of the discretization done on the joint positions. Each elementary cell produces a force with module related to the distance between the actual configuration \mathbf{q}

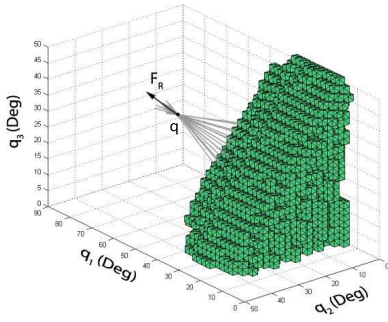


Fig. 3. Detected C -Obstacle: each C -Obstacle cell correspond to a manipulator configuration which a collision with the obstacle occurs. F_R represent the C -Force obtained in this example if the actual configuration is \mathbf{q} .

and direction given by the straight line joining the cell and \mathbf{q} . To reduce the computational requirements the repulsive force can be computed only for C -Obstacle cells near to \mathbf{q} because of the contribution produced by far cells is negligible (Fig. 3). This allows to rely only on local information, thus reducing the sensing and computational capabilities needed.

The repulsive force can be expressed as

$$\mathbf{F}_{RS}(\mathbf{q}) = K_S \sum_{i=1}^{N_C} \frac{1}{d(\mathbf{C}_i, \mathbf{q})} \vec{U}_i = K_S \sum_{j=1}^N F_{RS_j} \vec{q}_j,$$

where N_C is the number of the considered cells, $d()$ a function of the distance, $\vec{U}_i = \frac{\mathbf{q} - \mathbf{C}_i}{\|\mathbf{q} - \mathbf{C}_i\|}$ the direction between the i -th cell and the current configuration, K_S a proportional factor, and N the dimension of the C -Space. Because of the definition of C -Space the j -th component of the C -Force represents the force/torque to be applied on the j -th joint to perform the collision avoidance maneuver. The torque applied to the controlled joint (τ) can be expressed as

$$\tau = \tau_C + \mathbf{F}_{RS}(\mathbf{q}),$$

where τ_C is the torque generated by the free motion controller.

A. Obstacle Movements

To extend the C -Force approach to moving obstacles, a measure of the obstacle velocity must be provided to the system. Each object can be mapped into the C -Space with one or more *clusters*, composed by neighbor cells, whose center of gravity (COG) can be computed. A moving barycenter can be detected also for non moving objects because of noise on sensing devices. It is reasonable to assume that in this case the COG of the agglomerates is subjected to small movements only (Fig. 4-A).

A real obstacle motion will change at least one between size, shape and location of the C -Obstacle, consequently varying the position of the COG (Fig. 4-B).

The measured velocity for the COG of the i -th cluster can be defined as

$$\mathbf{V}_i = \frac{\Delta \mathbf{B}_i}{\Delta t},$$

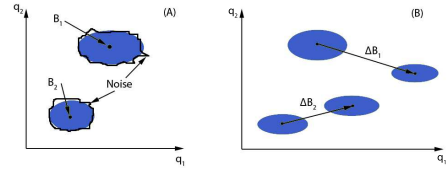


Fig. 4. COG variation: the C -Obstacle computation of a static object is subject to noise, then the COG is quasi-static (A). If the obstacle is moving the COG position change (B).

where $\Delta \mathbf{B}_i$ (Fig. 4-B) is the variation of the COG position for the i -th cluster, and Δt is the time between two different measures.

The unit vector from \mathbf{q} to the actual COG position \mathbf{B}_i (Fig. 5) is

$$\vec{I}_i = \frac{\mathbf{q} - \mathbf{B}_i}{\|\mathbf{q} - \mathbf{B}_i\|},$$

The component of \mathbf{V}_i who act on \mathbf{q} is the projection of \mathbf{V}_i

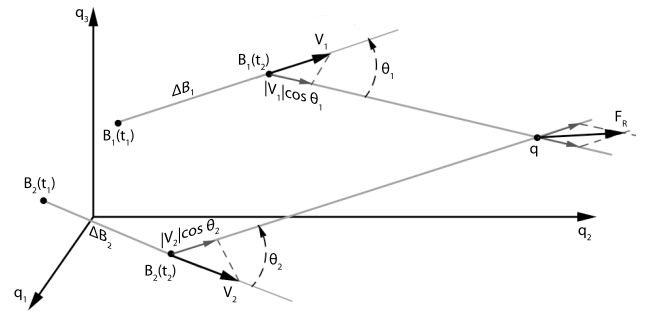


Fig. 5. Repulsive C -Force composition considering the variation of the C -Obstacle COG position.

on the line between \mathbf{q} and \mathbf{B}_i , its module can be expressed as

$$|\mathbf{V}_i| \cos \theta_i = \langle \mathbf{V}_i, \vec{I}_i \rangle. \quad (3)$$

where θ_i is the angle formed by \mathbf{V}_i and \vec{I}_i on the common plane, and \vec{I}_i is the direction. The repulsive force \mathbf{F}_R to implement the collision avoidance maneuver is

$$\mathbf{F}_R(\mathbf{q}) = \mathbf{F}_{RS}(\mathbf{q}) + K \sum_{i=1}^{N_{cl}} |\mathbf{V}_i| \cos \theta_i \vec{I}_i,$$

where N_{cl} is the number of the considered clusters.

B. Manipulator Movements

To improve safety of the C -Force approach also information about $\dot{\mathbf{q}} = \frac{\Delta \mathbf{q}}{\Delta t}$ can be used.

The relative velocities between manipulator and obstacles can be approximate with the relative velocities between \mathbf{q} and the C -Obstacle clusters as

$$|\mathbf{V}_{REL_i}| = \left| \frac{\Delta \mathbf{B}_i}{\Delta t} \right| \cos \theta_i - \left| \frac{\Delta \mathbf{q}}{\Delta t} \right| \cos \varphi_i, \quad (4)$$

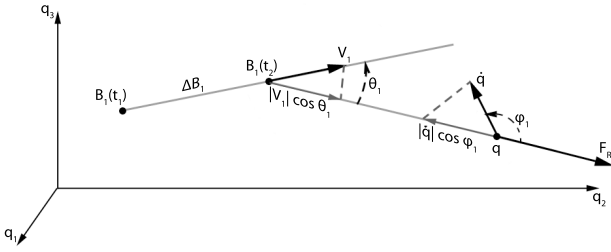


Fig. 6. Repulsive C -Force composition considering the manipulator movements.

where φ_i is the angle between $\dot{\mathbf{q}}$ and \vec{I}_i on the common plane (Fig. 6). Under these assumptions the total C -Force is

$$\mathbf{F}_R(\mathbf{q}) = \mathbf{F}_{RS}(\mathbf{q}) + K \sum_{i=1}^{N_{cl}} |\mathbf{V}_{REL_i}| \vec{I}_i$$

It should be noticed that the COG evaluation process should be not feasible runtime in a real-time application. To simplify the overall system only movements of cells of the C -Obstacle near to the actual manipulator configuration can be considered.

C. Evasive Maneuver without COG Computation

In the previous section the repulsive force was considered as composed by module $|\mathbf{V}_{REL_i}|$, and direction \vec{I}_i . If instead of the COG we consider the neighboring cells of the actual configuration \mathbf{q} we can assume as direction the unit vector $\vec{V} = \frac{1}{N_C} \sum_{i=1}^{N_C} \vec{U}_i$, and as relative velocity the variation of the mean distance

$$\frac{\Delta \mathbf{d}_M}{\Delta t} = \sum_{j=1}^N \frac{d_{M,j}(t) - d_{M,j}(t + \Delta t)}{\Delta t} \hat{\mathbf{q}}_j.$$

where $d_{M,j}(t)$ is the mean distance between \mathbf{q} and the C -Obstacle cells on the j -th component of the C -Space.

Under these assumptions the C -Force can be expressed as

$$\mathbf{F}_R(\mathbf{q}) = \mathbf{F}_{RS}(\mathbf{q}) + K \left\langle \frac{\Delta \mathbf{d}_M}{\Delta t}, \vec{V} \right\rangle \vec{V},$$

We call this method *MD C-Force (Mean Distance C-Force)*.

An alternative approach is to consider the repulsive force due to the distance between the current configuration and the C -Obstacle ($\mathbf{F}_{RS}(\mathbf{q}, t)$) and its time derivative, yielding as C -Force

$$\mathbf{F}_R(\mathbf{q}, t) = \mathbf{F}_{RS}(\mathbf{q}, t) + K \frac{\mathbf{F}_{RS}(\mathbf{q}, t) - \mathbf{F}_{RS}(\mathbf{q}, t - \Delta t)}{\Delta t}.$$

We will refer to this method as *PD C-Force (Proportional Derivative C-Force)*.

The *PD C-Force* method relays on the derivation of a value dependent to the inverse of a distance producing a control characterized by an high-bandwidth. Instead, the *MD C-Force* relying on the variation of a mean distance causes a smoother control.

It should be noticed that our method does not guarantee to perform the optimal trajectory, in the sense of minimum time or distance, or to have a single stability configuration;

however it ensures the respect of a safety distance to the obstacle (human operator) even if the obstacle is moving.

IV. PASSIVE COMPLIANCE - VARIABLE STIFFNESS TRANSMISSION (VST) ACTUATORS

To ensure safety against undetected or fast moving obstacles, passive techniques such as shown in [5] can be combined with any of the previous methods.

The manipulator employed in our experiment is the UNIPI SoftArm, a 3DOF manipulator actuated by McKibben muscles on agonistic-antagonistic configuration. As demonstrated in [2] the stiffness can be controlled changing the total pressure P_t common to the antagonistic muscles. The joint stiffness could be limited in order to improve safety, but this will cause a limit on the maximum torque, and consequently a bound on performance. As exposed in [6] an optimal tradeoff between safety and performance could be obtained solving the *Safe Brachistochrone*, whose solution represents the minimum time needed for the manipulator to reach the final position under safety constraints.

For the sake of simplicity, instead of solving the optimum control problem (equation 1), a suboptimal solution can be considered. Adopting the linear approximation $f(\sigma, v) = \sigma v$ the stiffness reference for the j -th joint can be obtained by evaluating

$$\sigma_j \propto P_{t_j} = K_1 \frac{1}{\dot{q}_j},$$

where K_1 is a scale factor and \dot{q}_j is the angular velocity of the j -th joint. This method allows low impact forces but, the \dot{q}_j tends to zero during an impact or in clamped condition and this could increase the injuries on the human operator.

An alternative approach is to control the joint stiffness proportionally to an estimation of the risk of collision, in order to decrease the stiffness only when necessary. The risk of collision depends on manipulator/obstacles distances and relative velocities, then an estimate of the risk of collision is already given by the repulsive C -Force value.

The C -Force VST (*CF-VST*) of the j -th joint is obtained as

$$P_{t_j} = K_2 \frac{1}{F_{R_j}(\mathbf{q})},$$

where F_{R_j} is the repulsive C -Force applied to the j -th joint.

An appropriate control of the joint stiffness could reduce injuries, hence the choice of the VST control is extremely important. As shown in the experimental results the best tradeoff between safety and performance is obtained using the *CF-VST* control because it works with the maximum performance if no objects are detected, and have low impact and clamping forces. The reliability of the vision system represents the core of the system. Indeed safety is not guaranteed if the C -Obstacle is not correctly evaluated. Combining the *V-VST* and the *CF-VST*, a degree of redundancy can be obtained. An approach to combine the two VST control methods is to adopt a time varying approximation of the optimal solution (equation 1) assuming $K_\sigma = K_1 - \gamma F_{R_j}(\mathbf{q})$.

Under these assumptions we obtain

$$P_{t_j} = (K_1 - \gamma F_{R_j}(\mathbf{q})) \frac{1}{\dot{q}_j},$$

where γ is a parameter related on vision subsystem reliability. This solution follows the human approach to avoid obstacles when the visual feedback is not guaranteed. Indeed, if visual is limited, humans use low stiffness perceiving a collision risk, and usually they adopt low stiffness while moving at high velocities.

V. SIMULATION RESULTS

To demonstrate the effectiveness of our approach the obstacle avoidance algorithms are tested in a simulated environment. A point to point task in a two dimensional C -Space is simulated until the target configuration is reached or a collision occurs. The C -Obstacle is simulated with a static or moving ellipse; the simulator considers that the obstacle avoidance algorithm knows the real position of the C -Obstacle only every Δt seconds. Whenever the position of the C -Obstacle is updated it is represented dashed. Start and target configuration, manipulator mobility, and C -Obstacle features are completely user-customized.

With these simulated 2D representation of C -Space environment the results obtained with a static and moving C -Obstacle are shown.

1) *Static Obstacle Avoidance*: Fig. 7 shows the configurations trajectory (solid line) performed adopting the repulsive C -Force method with a static C -Obstacle. The repulsive force doesn't acting when the C -Obstacle is far from the actual configuration (start configuration to A), a safe distance is insured when the C -Obstacle is near (A to B), then the target is reached with a straight trajectory (B to target).

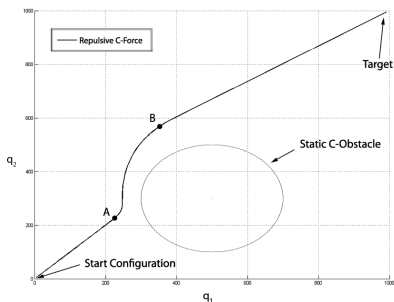


Fig. 7. Static obstacle avoidance simulation: the static obstacle is represented by an ellipse, the configurations trajectory (solid line) is all in the C -Free then the obstacle is avoided. The repulsive C -Force doesn't acting when the C -Obstacle is far (Start to A). when the obstacle is near a safety distance is maintained (A to B), then the target is reached with a straight trajectory (B to target).

Using the other presented C -Force method the same trajectory of Fig. 7 is obtained.

2) *Moving Object Avoidance*: In the simulation displayed in Fig.8 a moving C -Obstacle is considered. It should be noticed that the static repulsive C -Force method (solid line) collide with the C -Obstacle because of the C -Obstacle is too fast; The C -Obstacle can be avoided if its velocity is less

than a threshold that depends on the Δt value. The MD C -Force (dot-dashed line) and the PD C -Force (dashed line) avoid the obstacle with an evasive maneuver.

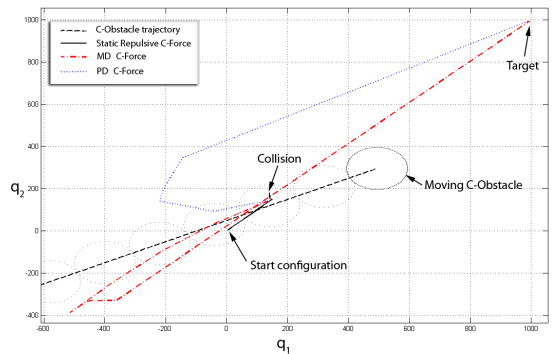


Fig. 8. Simulation with a moving C -Obstacle. Using only the $F_{RS}(q)$ a collision occur (solid). Both MDC -Force (dashed) and PDC -Force (dotted) avoids the obstacle. The PDC -Force method requiring a short time to reaching the target because of is characterized by a high bandwidth, instead MDC -Force is characterized by a more fluid trajectory.

Using the PD C -Force method the target is reached in shorter time than other method, due to its high bandwidth. The trajectory of the PD C -Force method is composed by fast changes of direction, that the human operator could feel this unsafe. The trajectory of the MD C -Force method is more fluid, such that the operator perceive an higher safety level.

VI. EXPERIMENTAL RESULTS

The presented methods are tested in real environment using the *UNIFI Soft-Arm* manipulator (Fig.1). In the first part the proposed VST methods are tested evaluating the impact forces to verify the results of section IV. In this case the obstacle avoidance module is not executed. In the second tests set a virtual obstacle is simulated, to verify the effectiveness of the obstacle avoidance module, while in the final part a real object in the task space is considered. An Arimoto controller is employed in all these tests to generate the nominal torques used in free motion.

A. Variable Stiffness Tests

The VST methods introduced in section IV are tested considering at first a free collision movement, and then crashes with a fixed obstacle equipped by a force sensor. The contact forces during the impact and after collisions are evaluated to compare the different VST methods. The free collision task for different VST methods are shown in Fig. 9. Performance can be evaluated comparing the step response characteristic.

In Fig. 10 are shown the results of collisions with a clamped force sensor.

In table I are shown the numeric results thus obtained. The method performance are evaluated considering the overshoot and the mean configuration error compared to the reference. The safety is characterized by the peak force at the impact

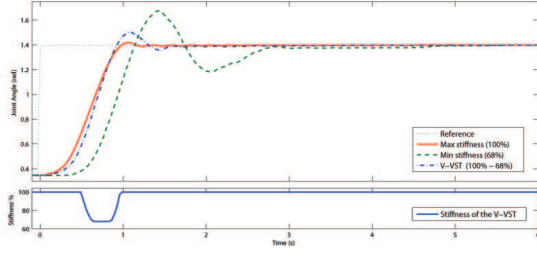


Fig. 9. Free collision tasks with different VST methods. The performance differences are clear considering the overshoot the delay and the setting time. Quantitative values are shown in Table I.

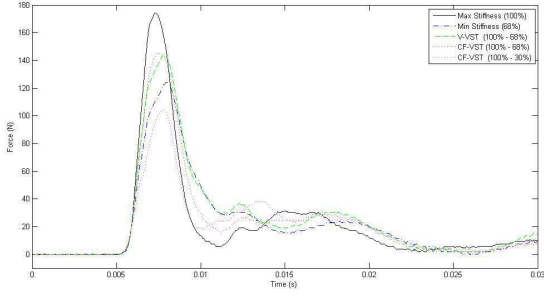


Fig. 10. Clamped collisions for different VST methods. Quantitative values are shown in Table I.

and the impulse computed as

$$\text{Impulse}_{3 \text{ sec}} = \int_{t^*}^{t^*+3 \text{ sec}} F(t)dt$$

where $F(t)$ is the contact force and t^* is the impact time. The impulse value consider the clamping forces that is an estimate of the human injury when is clamped by the robot. The different VST methods in Table I are: MAX with

TABLE I
VST TEST RESULTS

	MAX	MIN	V-VST	CF-VSTa	CF-VSTb
Overshoot %	1.48	19.17	7.47	1.48	1.48
Mean error (rad)	0.11	0.19	0.11	0.11	0.11
Peak force (N)	175.67	124.08	143.17	146.95	104.92
Impulse _{3 sec} (Ns)	20.54	12.19	20.78	10.61	0.99

constant stiffness at 100%; MIN with constant stiffness at 68%; V-VST and CF-VSTa with a variable stiffness between 100% and 68%, and CF-VSTb with a variable stiffness between 100% and 30%. If the stiffness is below 68% the arm workspace decreases, for this reason is used only with the CF-VST because of if an obstacle is detected safety is privileged with respect to performance.

B. Simulated C-Space

In this tests set the *C-Obstacle* is simulated with various positions, dimensions and shapes. Fig. 11 shows a simulated *C-Obstacle* and the configurations trajectory obtained. It should be noted that the configurations trajectory are on the *C-Free* (the subset of the *C-Space* where no collision

occur), and the obstacle is avoided. The repulsive force act on the manipulator even if the straight line between actual configuration and the target is included in the *C-Free*.

In all the tests the obstacle is avoided but two kind of undesired stability condition has been detected. The first happen when the repulsive force is equal to the force given by the free motion controller, while the second when the repulsive force impose to the manipulator to move to the workspace boundary. This happens when the desired trajectory is external to the manipulator workspace. In both these conditions the manipulator holds its position.

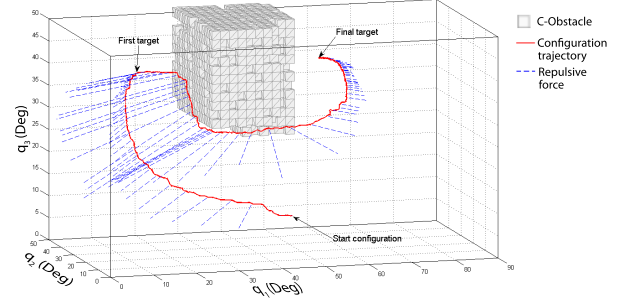


Fig. 11. Test result on a 3D *C-Space* with a simulated *C-Obstacle*, the manipulator trajectory (solid) is all in the *C-Free* then the obstacle is avoided. From the start configuration to the first target can be noted the non optimality of the repulsive *C-Force* method. A representation of repulsive *C-Forces* is displayed dashed.

C. Tests With Real Obstacles

In the second test set the obstacles (parts of a human body) are detected by using the Vision Subsystem and the *C-Obstacle* representation is computed using the collision configurations detector. A specific task is requested to the manipulator, and a human enters in the task space during the task execution. If the human obstacle is quasi-static its safety is guaranteed and the desired task is completed. A collision could occur if the human velocity is greater than a threshold related to the depth sensor bandwidth and characteristics. In Fig. 12 is displayed a test where the obstacle is a human hand.

VII. CONCLUSION AND FUTURE WORKS

In this paper a novel approach based on active and passive techniques to ensure human-robot coexistence is illustrated. The approach relies only on depth map information and Variable Stiffness, and is able to avoid moving obstacles guaranteeing safety even if the depth sensor fails.

This approach can be improved by using multiple depth sensors in order to integrate multiple obstacle depth maps that will reduce the grey areas: This is an important issue for the quality of the obstacle approximation. Future work will concern the detection of human, by employing a model of the human body.

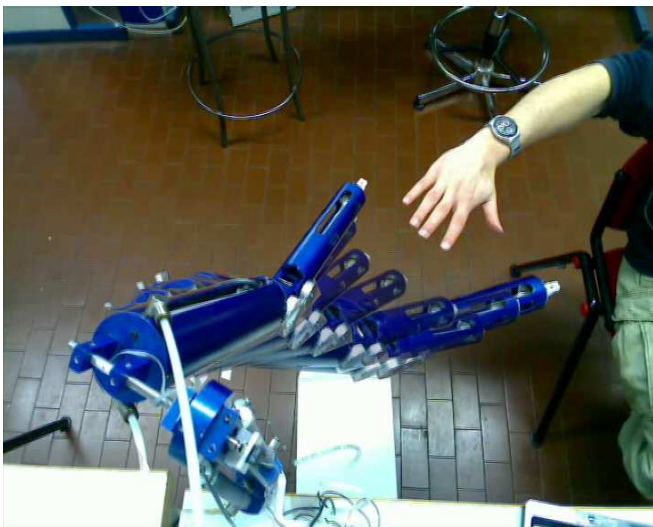


Fig. 12. Test result with real obstacle, the image is obtained superimposing some captured frames. The manipulator is commanded with a point to point task. Using the proposed method the target configuration is reached avoiding the operator hand.

VIII. ACKNOWLEDGMENTS

Authors would like to acknowledge the work done by Dr. Giovanni Tonietti. This work was supported by the PHRIENDS Specific Targeted Research Project, funded under the 6th Framework Programme of the European Community under Contract IST-045359. The authors are solely responsible for its content.

REFERENCES

- [1] Manipulating industrial robots - Safety, ISO 10218, 1992, Geneva.
- [2] A. Bicchi, G. Tonietti, Fast and Soft Arm Tactics: Dealing with the Safety-Performance Tradeoff in Robot Arms Design and Control, IEEE Robotics and Automation Magazine, Vol. 11, No. 2, June, 2004.
- [3] E.Yoshida, M.Poirier, J.P.Laumond, O.Kanoun: F.Lamiroux, R.Alami, and K.Yokoi, Whole-body motion planning for pivoting based manipulation by humanoids, IEEE International Conference on Robotics and Automation (ICRA 2008), Pasadena (USA), 19-23 Mai 2008, pp.3181-3186.
- [4] D. Kulic, E. Croft, Pre-collision safety strategies for human-robot interaction, Autonomous Robots archive Volume 22 , Issue 2, February 2007, pp.149-164.
- [5] R. Schiavi, G. Grioli, S. Sen, and A. Bicchi, VSA-II: A Novel Prototype of Variable Stiffness Actuator for Safe and Performing Robots Interacting with Humans, IEEE Int. Conf. on Robotics and Automation, 2008
- [6] G. Tonietti, R. Schiavi, and A. Bicchi, Design and Control of a Variable Stiffness Actuator for Safe and Fast Physical Human/Robot Interaction, IEEE Int. Conf. on Robotics and Automation, pages 528-533, 2005
- [7] Novak J.L., Feddema J.T., A Capacitance-Based Proximity Sensor for Whole Arm Obstacle Avoidance, IEEE Proceedings of the Intl. Conf. on Robotics and Automation, pp. 1307-1314, 1992.
- [8] Feddema J.T., Novak J.L., Whole Arm Obstacle Avoidance for Teleoperated Robots, IEEE Robotics and Automation Proceedings, pp.3303 - 3309, 1994.
- [9] D. Ebert, T. Komuro, A. Namiki, M. Ishicawa, Safe Human-Robot-Coexistence: Emergency-Stop Using a High-Speed Vision-Chip, Ishikawa Namiki Laboratory, Department of Information Physics and Computing, Graduate School of Information Science and Technology, University of Tokyo, Japan

- [10] Kuhn S., Gecks T. and Henrich D., Velocity control for safe robot guidance based on fused vision and force/torque data, IEEE Conference on Multisensor Fusion and Integration, September 3-6, 2006, Heidelberg, Germany
- [11] Gecks T., Henrich D., Human-robot cooperation: Safe Pick-and-Place Operations, IEEE International Workshop on Robots and Human Interactive Communication, 2005
- [12] Iossifidis I., Schöner G. , Dynamic System Approach for the Autonomous Avoidance of Obstacle and Joint-limit for an Redundant Robot Arm, in proceeding of the IEEE International Conference on Intelligent Robots and System, Beijing, China, 2006
- [13] O. Khatib, Real Time Obstacle Avoidance for Manipulators and Mobile Robots, The international journal of robotics research, 5(1):90-99, Spring 1986
- [14] M. Erdmann, Using Backprojections for Fine Motion Planning with Uncertainty, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, 1985
- [15] E. Rimon, Exact robot navigation using artificial potential functions, PhD Thesis, Yale University, 1990.
- [16] L. Balan and G. M. Bone, Real-time 3D Collision Avoidance Method for Safe Human and Robot Coexistence, International Conference on Intelligent Robots and Systems, October 9 - 15, 2006, Beijing, China