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

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

Workspace 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.

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 (c.f. e.g.[4]). 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 have been addressed by employing different sensing techniques. For instance, the work in [7] 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 low cost real-time obstacle detection. In this line, Ebert et al. [8] proposed an emergency-stop approach based on an ad-hoc developed vision chip; Kuhn et al. [9] used multiple cameras to detect obstacles in the workspace; Iossifidis [11] uses 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 [12]. In these methods, repulsive forces are generated in the robot operational space. Erdmann [13] introduced the idea of using forces in configuration-space (C-Space), generated by C-Space obstacles (C-Obstacle). Rimon, in his Ph.D thesis [14] 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.
translated in a control paradigm described as “stiff and slow, fast and soft”. In [6], a computationally efficient, suboptimal approximation of the optimal solution of the “safe brachistochrone” problem underpinning this paradigm was proposed, which is used here to control the robot arm.

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 C-Obstacle in real time from depth map information. Section III describes how repulsive forces are generated on the basis of the 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 V and section VI report simulation and experimental results obtained by implementing the proposed methods.

II. 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 to be imposed on the 3D shapes (cf. e.g. [15]).

Our approach considers the camera Image Plane (IP) to detect the representation of obstacles in the space of configurations (C-Space); that representation is called C-Obstacle. The novelty of this approach is to reduce the collision detection to a comparison between matrix elements.

Let us consider the projection on the IP of manipulator in a configuration \( \tilde{q} \) and detected obstacles. A necessary condition to detect if the manipulator collide against the obstacles ([10]) 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 \( x, y \) 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 \( x, y \). The depth map of obstacles \( (O_{dm}) \) can be obtained real-time by using depth sensors such as PMD cameras, laser range finder, and stereo video systems.

The points of the manipulator in its own reference frame can be computed by using direct kinematics for the configuration \( q \). These coordinates can be transformed to the camera frame, in which the \( z \)-coordinate represents the depth, by using the extrinsic camera parameters as proposed by the pinhole camera model ([16]). Finally the points can be projected in the IP, by using the intrinsic camera parameters, thus obtaining the depth map of the manipulator for the configuration \( q \) \( (M_{dm}(q)) \). For this purpose a manipulator model is used, which can be obtained by using 3D CAD programs. This does not affect the system performance, because the \( M_{dm} \) data set can be computed once 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.

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

\[
M_{dm}(q)(x, y) \geq O_{dm}(x, y) - \epsilon,
\]

where \( \epsilon \) is a safety-margin parameter. A discretization of the C-Space is obtained considering a discrete set for the configuration of the manipulator. Due to this discretization the C-Space can be seen as composed by elementary cells. A cell is included in the C-Obstacle if the correspondent configuration verify the condition 1. On Fig. 3 an example of C-Obstacle is presented. As shown in [9], [3] the minimum distance, e.g. \( \min_{(x,y) \in IP} O_{dm}(x, y) - M_{dm}(q)(x, y) \), can be used to limit the maximum velocity or reduce the acceptable injury risk level. This measure have also the effect of increasing the human feeling of safety.

![Fig. 2. By projecting the manipulator points in the IP the collision is evaluated comparing the obstacle and the manipulator depth maps.](image)

![Fig. 3. Example of a C-Obstacle for a 3dof manipulator: each C-Obstacle cell correspond to a manipulator configuration which a collision with the obstacle occurs. Fig represent the C-Force obtained in this example if the actual configuration is q.](image)

III. Active Compliance: Repulsive C-Force

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

Basic ally the method is based on considering a repulsive force/torque generated by each elementary cell of the C-Obstacle. This force/torque has a module related to the distance between the actual configuration $q$ and the considered C-Obstacle cell, and a direction given by the straight line passing through them. To reduce the computational requirements the repulsive force can be computed only for C-Obstacle cells near to $q$ because of the contribution produced by far cells is negligible (Fig. 3). This allows the collision avoidance method to rely only on local information, thus reducing the sensing and computational capabilities needed.

The repulsive force can be expressed as

$$ F_{RS}(q) = K_s \sum_{i=1}^{N_C} \frac{1}{d(C_i, q)} \vec{U}_i = K_s \sum_{j=1}^{N} F_{RS,j} \vec{I}_j, $$

where $N_C$ is the number of the considered cells, $d()$ a function of the distance, $\vec{U}_i = \frac{q-C_i}{\|q-C_i\|}$ 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 $i$-th joint ($\tau_i$) can be expressed as $\tau_i = \tau_{C_i} + K_{dim_i} F_{RS,j}$, where $\tau_{C_i}$ is the torque generated by the free motion controller and $K_{dim_i}$ is a dimensional adjustment needed to have forces or torques.

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 COG 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 noise, but because of the limited size of the neighbor cells near to the actual manipulator configuration, the movement of the COG can be computed approximately using only neighbor cells, instead of the entire set of C-Space elements.

The measured velocity for the COG of the $i$-th cluster can be defined as $V_i = \Delta B_i \Delta t$, where $\Delta B_i$ is the variation of the COG position for the $i$-th cluster, and $\Delta t$ is the time between two different measures.

The unit vector from $q$ to the actual COG position $B_i$ (Fig. 4) is $\vec{I}_i = \frac{q-B_i}{\|q-B_i\|}$. The component of $V_i$ who act on $q$ is the projection of $V_i$ on the line between $q$ and $B_i$, its module can be expressed as $|V_i| \cos \theta_i = \langle V_i, \vec{I}_i \rangle$, where $\theta_i$ is the angle formed by $V_i$ and $\vec{I}_i$ on the common plane, and $\vec{I}_i$ is the direction. The repulsive force $F_R$ to implement the collision avoidance maneuver is

$$ F_R(q) = F_{RS}(q) + K \sum_{i=1}^{N_{cl}} |V_i| \cos \theta_i \vec{I}_i, $$

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

B. Manipulator Movements

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

The relative velocities between manipulator and obstacles can be approximate with the relative velocities between $q$ and the C-Obstacle clusters as

$$ |V_{REL,i}| = \left| \frac{\Delta B_i}{\Delta t} \right| \cos \theta_i - \left| \frac{\Delta q}{\Delta t} \right| \cos \phi_i, $$

where $\phi_i$ is the angle between $q$ and $\vec{I}_i$ on the common plane (Fig. 5). Under these assumptions the total C-Force is

$$ F_R(q) = F_{RS}(q) + K \sum_{i=1}^{N_{cl}} |V_{REL,i}| \vec{I}_i. $$

Where $N_{cl}$ is the number of cluster. It should be noticed that the COG evaluation process may not be feasible real-time. 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 $|V_{REL,i}|$, and direction $\vec{I}_i$. If instead of the COG we consider the neighboring cells of the actual configuration $q$ 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

$$ \frac{\Delta d_M}{\Delta t} = \sum_{j=1}^{N} \frac{d_{M,j}(t) - d_{M,j}(t+\Delta t)}{\Delta t} q_j, $$

where $d_{M,j}$ is the distance from manipulator joint $j$.
where \( d_{M,j}(t) \) is the mean distance between \( q \) and the \( \mathcal{C} \)-\( \text{Obstacle} \) cells on the \( j \)-th component of the \( \mathcal{C}-\text{Space} \). Under these assumptions the \( \mathcal{C}-\text{Force} \) can be expressed as

\[
\mathbf{F}_R(q) = \mathbf{F}_{RS}(q) + K \left( \frac{\Delta d_{M,j}}{\Delta t}, \dot{V} \right) \ddot{V}.
\]

We call this method \( MD \) \( \mathcal{C}-\text{Force} \) (\( \text{Mean Distance \( \mathcal{C}-\text{Force} \)\)}).

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

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

We will refer to this method as \( PD \) \( \mathcal{C}-\text{Force} \) (\( \text{Proportional Derivative \( \mathcal{C}-\text{Force} \)\)}).

The \( PD \) \( \mathcal{C}-\text{Force} \) method relies on the derivation of a value dependent to the inverse of a distance and produces a control characterized by a high bandwidth. Instead, the \( MD \) \( \mathcal{C}-\text{Force} \) 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.

The computational load grows with the DOF of the manipulator. To allow the real-time specific the dimension of the \( \mathcal{C}-\text{Space} \) cells and the \( N_C \) parameter can be adapted.

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 \( \text{Safe Brachistochrone} \), whose solution represents the minimum time needed for the manipulator to reach the final position under safety constraints.

To enable real time planning, instead of solving the optimum control problem, the suboptimal solution of [6] is adopted, which can be expressed by the formula

\[
f(\sigma, v) = K_\sigma,
\]

where \( \sigma \) is the joint stiffness, \( v \) is the joint velocity, and \( K_\sigma \) is a function depending on the link and rotor inertias, the arm configuration and parameters, and expressing the acceptable level of injury risk. As in our previous work on variable stiffness actuator control ([6]), the value of \( K_\sigma \) can be set to a constant value \( K_1 \), depending on the task at hand.

Adopting a bilinear approximation for the expression in (3), i.e. \( f(\sigma, v) = \sigma v \), a reference stiffness for the \( j \)-th joint could be obtained as

\[
\sigma_j \propto P_{t_j} = \text{sat} \left( K_1 \frac{1}{q_j} \right),
\]

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

When used together with active safety control and sensing, a situation-adaptive approach can be used instead, consisting in controlling the maximum injury risk level proportionally to an estimation of the risk of collision, so as to decrease the stiffness only when necessary. As the risk of collision depends on manipulator/obstacles distances and relative velocities, an estimate of the risk of collision can be given by the repulsive \( \mathcal{C}-\text{Force} \) value, i.e. we can set

\[
\sigma_j = K_2 \frac{1}{F_{R_t}(q)},
\]

where \( F_{R_t} \) is the repulsive \( \mathcal{C}-\text{Force} \) applied to the \( j \)-th joint.

Intuitively, this method (CF-VST) achieves better performance in the absence of humans in the workspace, and it provides adequate safety margins whenever a human is detected. However, it does rely critically on sensing, and might result unsafe if the detection and localization system fails. The two approaches can be combined in a time-varying approximation of the optimal solution by setting \( K_\sigma = K_1 - \gamma F_{R_t}(q) \). Under these assumptions we obtain

\[
P_{t_j} = \left( K_1 - \gamma F_{R_t}(q) \right) \frac{1}{q_j},
\]

where \( \gamma \) is a parameter related to the vision subsystem reliability. This solution is reminiscent of the human approach to control arm stiffness, which is set to a low value when moving at high velocities, as well as in conditions where visual feedback is impaired.

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 \( \mathcal{C}-\text{Space} \) is simulated until the target configuration is reached or a collision occurs. The \( \mathcal{C}-\text{Obstacle} \) is simulated with a static or moving ellipse; the simulator considers that a static or moving ellipse; the simulator considers that the obstacle avoidance algorithm knows the real position of the \( \mathcal{C}-\text{Obstacle} \) only every \( \Delta t \) seconds. Whenever the position of the \( \mathcal{C}-\text{Obstacle} \) is updated it is represented dashed. Start and target configuration, manipulator mobility, and \( \mathcal{C}-\text{Obstacle} \) features are completely user-customized. With these simulated 2D representations of \( \mathcal{C}-\text{Space} \) environment the results obtained with moving \( \mathcal{C}-\text{Obstacle} \) are shown.

It should be noticed that the static repulsive \( \mathcal{C}-\text{Force} \) method (solid line) collide with the \( \mathcal{C}-\text{Obstacle} \) because of the \( \mathcal{C}-\text{Obstacle} \) is too fast; The \( \mathcal{C}-\text{Obstacle} \) can be avoided if its velocity is less than a threshold that depends on
the $\Delta t$ value. The MD $C$-Force (dot-dashed line) and the PD $C$-Force (dashed line) avoid the obstacle with an evasive maneuver.

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 UNIPI 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 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 collision free task for different VST methods are shown in Fig. 7 (left). Performance can be evaluated comparing the step response characteristic.

In Fig. 7 (right) are shown the results of collisions with a clamped force sensor.

In table VI-A are shown the numeric results thus obtained. The performance of the methods are evaluated considering the overshoot and the mean configuration error (Me) with respect to the reference. The safety is characterized by the peak force (Pf) produced by an impact and the impulse ($I_3$) computed as $I_3 = \int_{t^*}^{t^*+3} F(t)dt$, where $F(t)$ is the contact force and $t^*$ is the impact time. The $I_3$ value considers the clamping forces that is an estimate of the human injury when is clamped by the robot. The different VST methods in Table VI-A are: MAX with 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 whenever an obstacle is detected safety is privileged with respect to performance.

B. Simulated C-Space

In this test set the C-Obstacle is simulated with various positions, dimensions and shapes. Fig. 8 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.

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 this case the operator safety is preserved by the passive compliance system. Fig. 9 shows 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 obstacles approximations. Future work will also concern the detection of humans, by employing a model of the human body.

VIII. ACKNOWLEDGMENTS

Authors would like to acknowledge the helpful comments by 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

[8] 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