Experimental Real-Time Setup for Vision Driven Hand-Over with a Collaborative Robot

Experimental Real-Time Setup for Vision Driven Hand-Over with a Collaborative Robot

II. Collaborative Workspace and Task Design

Fig. 1 shows the layout of the collaborative space. The human operator is standing in front of the robot, which is mounted on the workbench. They are displaced so that the intersection between the human and the robot workspace produces a collaborative volume which corresponds to almost 40% of the robot workspace. This choice opens to a wide range of hand-over positions (meeting points) and gives the human the possibility to meet the robot casually inside the collaborative volume. Two Microsoft Kinect sensors are displaced behind the robot, with their principal axis forming a 70 degrees angle and intersecting approximately at the center of the human workspace, which is measured considering the extended upper limbs as the radius of two sphere centered on the shoulders. The human joints are tracked from two points of view and multiple-Kinect algorithm is used to output an optimized skeleton at sample rate of 30 Hz [11]. The human upper body position is then interfaced with the UR3 robot in the World frame, which is built using reference points and the calibration procedure described in [4]. Fig. 2 illustrates the schematics of the task of human and robot, whose goal is to hand-over parts to be assembled starting and finishing from their relative resting pose. Without loss of generality, both human and robot can be giver or receiver, since the aim is to accomplish the collaborative task without discussing the human perception during the hand-over stages, because many contributes already investigated this field. In this work, the human is the giver and starts his task pointing towards the exchanging area. When the hand enters in the collaborative volume V_c, the robot reacts and drives its end effector towards the target. Once the hand and the robot gripper are at the meeting distance, the robot stops and receives the object. In this phase, the human can adjust his hand pose for the hand-over within a meeting volume V_m, in order to reach the interaction without that the UR3 moves. The meeting volume is a sphere centered on the robot TCP. Finally, the human retracts his upper limb and the robot goes to its home position as the hand exits from the meeting volume. In Fig. 2 the main task sequence is labeled with circular boxes, while the giver and receiver sub-tasks are introduced on the side of each brench; in the middle, the trigger events are also shown.

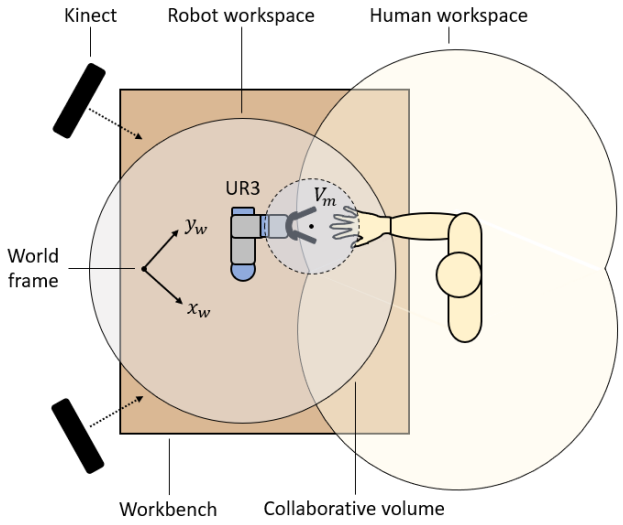

Fig. 1. Collaborative set-up for human-robot hand-over.

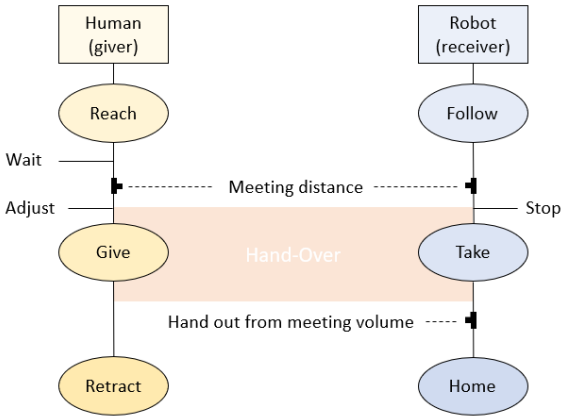

Fig. 2. Structure of the hand-over task for human and robot.

It is important to notice that the task of human and robot are connected in time and space: the relative position of hand and end-effector triggers the hand-over stages in order to execute the task synchronously, within the limit of collaborative application. In fact, the maximum robot TCP cartesian velocity is set to 0.25 m/s for collaborative purpose, resulting in a waiting phase for the operator, who's usually reaching faster the meeting point. In general, this will not affect the hand-over position if the separation of direction and velocity method is used for motion planning [4].

III. Hand-over Control Strategy

The hand-over control strategy is based on two different algorithms: a duplex-Kinect algorithm and a path-planning algorithm, as in [4]. The duplex-Kinect algorithm is developed according to [11], instead the path-planning algorithm has been modified. The improvements in the path-planning are presented in this Section. In [4] the robot, in order to perform a hand-over task, considers as target point the right hand of the human operator. In this work the robot target is a virtual point, called *virtual hand*, that is an extension of the hand on the x_h- y_h plane as shown in Fig. 3. The origin of the hand frame moves rigid to the hand, while its orientation is fixed with the axis parallel to the world frame. This solution allows the robot to stop its motion in front of the human hand and to achieve the hand-over properly. In fact, considering the human hand as target point, the robot could place its TCP above or below the hand of the operator. Furthermore, using the virtual hand, it is not necessary to consider a security sphere around the hand of the human operator, because the robot stops away from the hand.

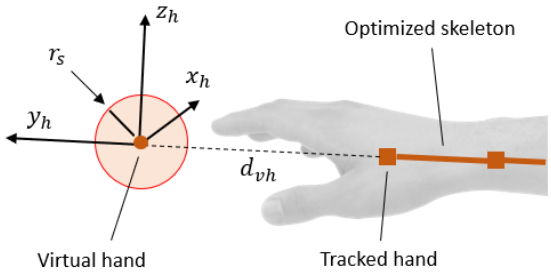

Fig. 3. The virtual hand uses as target point for the hand-over task. d_{vh} is the distance between the tracked hand and the virtual one.

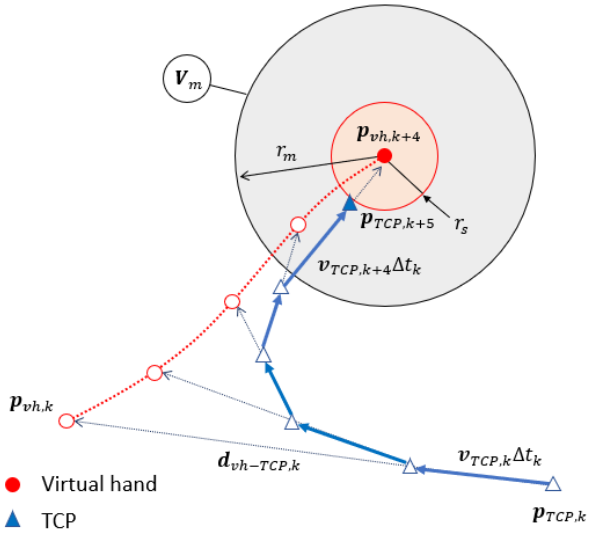

Fig. 4. Motion planning resulting from the hand-following algorithm for successive samples k, where Δt_k is the sampling rate of the vision system.

For control purposes, two spheres have been considered in this work, as can be seen in Fig. 4. The internal sphere with radius r_s is called *stopping volume* V_S, since the robot stops moving when the TCP reaches this volume around the target. The external sphere with radius r_m is the *meeting volume*: after the TCP reached the stopping volume, the robot can move again only when the distance between the virtual point and the TCP is bigger than r_m, therefore when the operator retracts his hand after the hand-over task has been accomplished. Using the meeting volume, small displacements of the hand from the TCP are permitted without involving robot motion. This facilitates the fulfillment of the hand-over task. The position vector of the virtual hand $\boldsymbol{p_{vh}}$ is used to generate the TCP linear velocity vector $\boldsymbol{v_{TCP}}$ to drive the robot towards the target. In [4], the velocity $\boldsymbol{v_{TCP}}$ is obtained from the distance vector $\boldsymbol{d_{vh-TCP}}$ between the target point position vector and the TCP position vector, through a constant diagonal matrix K, as in (2):

$$\boldsymbol{d_{vh-TCP}} = \boldsymbol{p_{vh}} - \boldsymbol{p_{TCP}} \tag{1}$$

$$\boldsymbol{v_{TCP}} = K\boldsymbol{d_{vh-TCP}} \tag{2}$$

In this work, a smooth profile for the TCP linear velocity is adopted, to obtain no sudden movements of the robot. In Fig. 5, the magnitude of the linear velocity v_{TCP} versus the distance between the target point and the TCP is shown. For what concerns the orientation of the TCP, this is represented in terms of unit quaternion and the angular velocity vector $\boldsymbol{\omega_{TCP}}$ is calculated as in [4]. The joint velocities vector $\dot{\boldsymbol{q}}$, that is the result of the path-planning algorithm, is obtained inverting the Jacobian J

$$\dot{\boldsymbol{q}} = J^{-1} \cdot [\boldsymbol{v_{TCP}}\ ;\ \boldsymbol{\omega_{TCP}}] \tag{4}$$

To avoid any unintentional contact between the links of the robot and the human operator, a collision avoidance algorithm, based on the works presented in [10, 12-13], is considered.

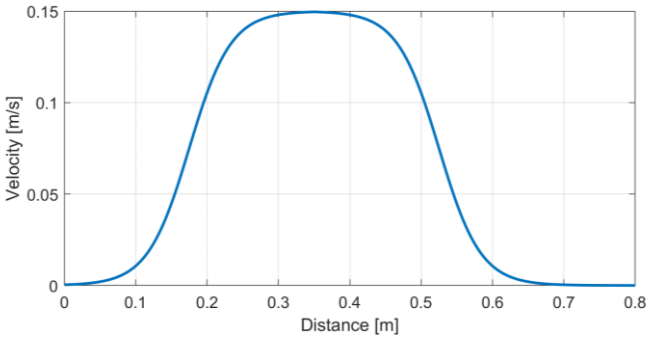

Fig. 5. Profile of the magnitude TCP linear velocity versus the distance between the virtual hand and the TCP.

The results of the collision avoidance algorithm, expressed as a joints velocity vector $\dot{\boldsymbol{q}}_{ca}$, and the ones of the path-planning algorithm are summed up in order to avoid collisions while the robot try to reach the target point.

IV. Experimental Tests

To validate the effectiveness of the hand-over control strategy, experimental tests have been carried out. In this section, the experimental setup and the results of significant tests are described.

A. Experimental setup

In Fig. 6 a scheme of the experimental hardware is shown. The experimental setup consists of three PCs, two Microsoft Kinect v2, a UR3 robot from Universal Robots and a router. Each Kinect is connected to a PC. The PCs have i7-6700 processor and 32 GB RAM and the algorithms run in the Matlab environment. The router has a transfer speed of 300 Mbps. In the following, the two PCs connected to the Kinect will be called "Master" and "Slave" and the third PC "External robot controller". The Master is in charge of the acquisition process from the two Kinect. In fact, the Master handles the start/stop triggers for the acquisition of the sensors to synchronize the samples. In this way, the Master and the Slave receive the data from the Kinect and locate them in the World reference frame, obtaining at the sample time t_k the skeletons $\boldsymbol{S}_{M,k}$ and $\boldsymbol{S}_{S,k}$. Then the Slave sends the skeleton $\boldsymbol{S}_{S,k}$ to the Master, that uses the skeleton data of the PCs to generate the optimized skeleton $\boldsymbol{S}_{opt,k}$, that is the result of the duplex-Kinect algorithm.

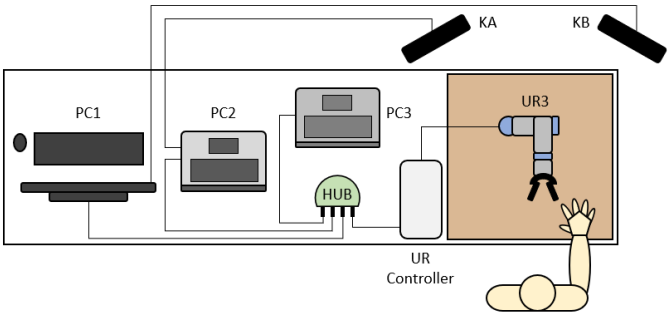

Fig. 6. Experimental hardware layout.

1. *for* $k \in \{kinect\ samples\}$
2. $\boldsymbol{S}_{A,k}, \boldsymbol{S}_{B,k} \longrightarrow Master$
3. $\boldsymbol{S}_{opt,k} \longrightarrow \boldsymbol{p}_{joints,k}$
4. $\boldsymbol{p}_{joints,k} \longrightarrow \boldsymbol{p}_{vh,k}$
4. $fb\ data, \boldsymbol{p}_{vh,k} \longrightarrow Ext.robot\ controller$
5. *if* $\boldsymbol{p}_{vh,k} \in V_{CV}$ && $\boldsymbol{p}_{TCP,k}, \boldsymbol{p}_{vh,k} \notin V_{SV}$
6. $Hand\ following \longrightarrow \boldsymbol{v}_{TCP,k}$
7. $Collision\ Avoidance \longrightarrow \dot{\boldsymbol{q}}_{ca,k}$
8. $\dot{\boldsymbol{q}}_k = \boldsymbol{J}_k^{-1}\boldsymbol{v}_{TCP,k} + \dot{\boldsymbol{q}}_{ca,k} \longrightarrow UR3\ control.$
12. *else* $Hand\ following = false$
13. *end*
14. *End*

Fig. 7. Pseudocode of the hand-over strategy.

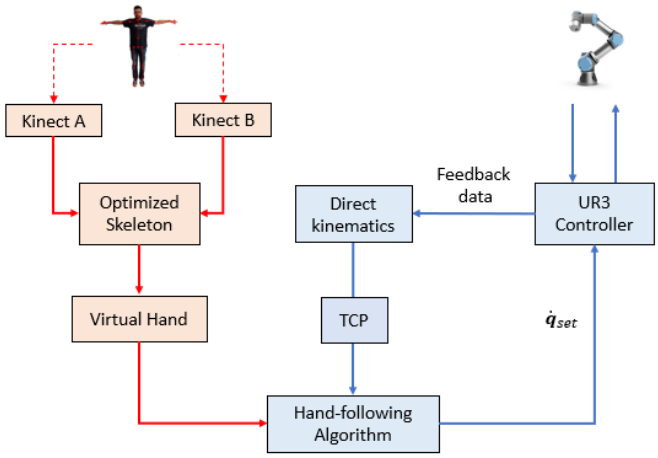

Fig. 8. In red the loop of the virtual hand position calculation that runs with a 30Hz frequency, in blue the UR3 control loop that update the $\dot{\boldsymbol{q}}_{set}$ with 60Hz frequency.

The optimized skeleton data (the joint positions $\boldsymbol{p}_{joints,k}$) are then sent from the Master to the External robot controller, that calculates the position of the virtual hand $\boldsymbol{p}_{vh,k}$. The External robot controller receive also feedback data from the controller of the robot. The feedback data and the target point position are the input of the path-planning algorithm, that generates the joints velocity vector $\dot{\boldsymbol{q}}_k$. Finally, the External robot controller send the vector $\dot{\boldsymbol{q}}_k$ to the controller of the robot through UR3 customized APIs, built by the authors to interface the robot with Matlab. A pseudocode of the Kinect acquisition process and the calculation of the set velocity vector $\dot{\boldsymbol{q}}$ for the controller is reported in Fig. 7, while the flowchart of the hand-over strategy can be seen in Fig. 8. The virtual hand position information is updated with a 30Hz frequency, instead the vector $\dot{\boldsymbol{q}}$ is calculates with 60Hz frequency. So, there are two control loops: an external one that calculates the target position every 0.033 ms, and an internal one that works with a frequency of 60Hz and generates the velocity set for the controller of the UR3.

B. Tests

Different movements towards casual meeting points into the collaborative volume were performed by the human. In all cases it was observed that the robot can meet the human hand with a final pose which is suitable for the hand-over, both in position and orientation. In Fig. 9 the frames of a hand-over characteristic test are showed, while Fig. 10 illustrates, for the same test, the relative distance between the human hand and the TCP of the robot versus the time. The 6 frames identify the main stages of the hand-over task seen in Fig. 2, which are also indicated on the distances curve with different marks. If the hand is outside the collaborative volume, the robot does not move; when the human approaches the collaborative volume, the hand-following algorithm drives the UR3 TCP, which starts pointing toward the dynamic target (*follow*). Due to the restricted robot velocity [5], the human reaches a meeting point (*reach*) and waits for the robot (*wait*). Once the relative distance assumes the stopping sphere radius value, the UR3 stops (*stop*) and the hand-over is accomplished (*hand-over*). Finally, the human exits the meeting volume (*retract*) and the robot takes the object in its home position (*home*). During the targeting phase, even if the robot represents an obstacle for the vision system, the optimization made on the skeleton solves the occlusions and produces a quite stable track of the human hand. Results in terms of trajectories are showed in Fig. 11, where the hand paths and the TCP paths are plotted, together with the stopping volume, for different meeting points. In Fig. 12 the module of the TCP linear velocity vector versus the time for a characteristic test is shown. The profile of the velocity is quite smooth, so there were no sudden movements of the robot. The norm of the TCP velocity is zero when the hand is outside the collaborative volume (t < 1s) and when the distance between the virtual hand and the TCP is less than (t > 5s).

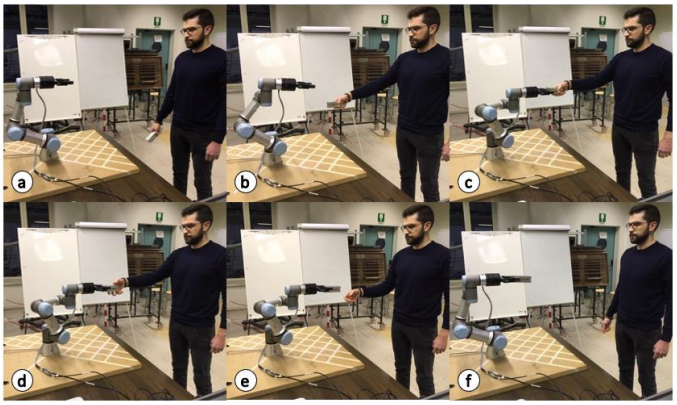

Fig. 9. Hand-over task, test frames: a) task start; b) follow; c) reach; d) hand-over; e) retract; f) home.

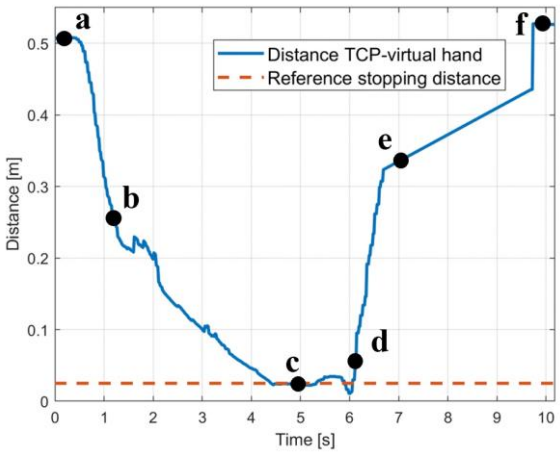

Fig. 10. Distance virtual hand-TCP in function of task time

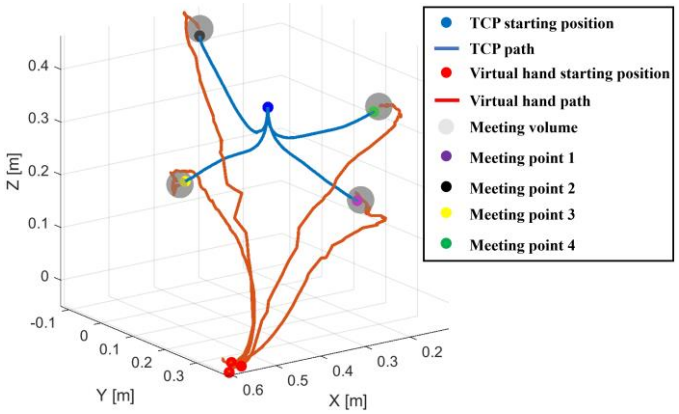

Fig. 11. Paths of the virtual hand and of the TCP of the robot for different meeting points.

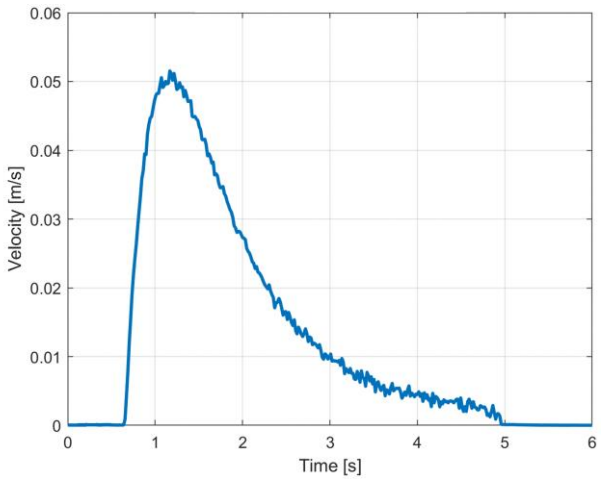

Fig. 12. Norm of the TCP linear velocity vector during the hand-over task.

V. Conclusions

In this work, a collaborative workspace and a hand-over strategy implemented on a UR3 to perform a collaborative task were presented and described. The collaborative set-up included optimized human skeleton tracking by means of two depth cameras and a local control strategy for the on-line motion planning of the robot. The collaborative task consisted in a 6-stage hand-over in which the giver (human) moved towards casual meeting point inside the robot (receiver) workspace. The proposed hardware architecture used a Master-Slave computer configuration to run the human tracking algorithm at 30 Hz. The path-planning algorithm was handled by a third computer, which sent data to the robot controller through UR3 customized APIs. Finally, the UR3 controller updates the robot velocity set at the frequency of 60 Hz. Tests were performed to verify that the strategy successfully accomplishes the hand-over between the human operator and the collaborative robot. Results validated the control algorithm and the robot motion planning strategy, showing the improvements coming from the introduction of a virtual target, which is an extension of the human hand. Future works will involve prediction of the movements of the human operator to refine the algorithm and different modelling of the TCP linear velocity to obtain human-like movements of the robot.

References

[1] J. Kruger, T.K. Lien, and A. Verl,"Cooperation of human and machines in assembly lines," in CIRP Annuals, Manufacturing Technology, vol.58, 2009, pp. 628-646.

[2] M. Cakmak, S. S. Srinivasa, M. K. Lee, S. Kiesler, and J. Forlizzi, "Using spatial and temporal contrast for fluent robot-human hand-overs," in Proc. 6th ACM/IEEE Int. Conf. Human-Robot Interaction, 2011, pp. 489-496.

[3] G. Michalos, S. Makris, P. Tsarouchi, T. Guasch, D. Kontovrakis, and G. Chryssolouris, "Design considerations for safe human-robot collaborative workplaces," in Proc. CIRP - Understanding the life applications of manufacturing, vol. 37, 2015, pp. 248-253.

[4] M.Melchiorre, L. Scimmi, S.Mauro, and S.Pastorelli "Influence of human limb motion speed in a collaborative hand over task," in Proc. of the 15th Int. Conf. on Informatics in Control, Automation and Robotics, vol.2, 2018, pp 349-356.

[5] https://www.iso.org/standard/62996.html

[6] A. Koene, A. Remazeilles, M. Prada, A. Garzo, M. Puerto, S. Endo, and A. M. Wing, "Relative importance of spatial and temporal precision for user satisfaction in human-robot object handover interaction," in Proc. New Frontiers in Human-Robot Interaction, 2014

[7] C. M. Huang, M. Cakmak, B. Mutlu, "Adaptive coordination strategies for human-robot handovers," in Robotics: Science and Systems, 2015, pp. 1-10.

[8] W. Wang, R. Li, Z. M. Diekel, Y. Chen, Z. Zhang, and Y. Jia, "Controlling object hand-over in human-robot collaboration via natural wearable sensing," in IEEE Transactions on Human-Machine Systems, vol. 49(1), 2019, pp. 59-71.

[9] https://www.universal-robots.com/products/ur3-robot/.

[10] L. S. Scimmi, M. Melchiorre, S. Mauro, and S. Pastorelli, "Multiple collision avoidance between human limbs and robot links algorithm in collaborative tasks," in Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics, vol.2, 2018.

[11] K. Y. Yeung, T. H. Kwok, and C. L. Wang, "Improved skeleton tracking by a duplex kinects: a practical approach for real-time applications," In Journal of Computing and Information Science in Engineering, 13(4), 2013.

[12] S. Mauro, S. Pastorelli, and L. S. Scimmi, "Collision Avoidance Systems for Collaborative Robotics," In Mechanisms and Machine Science, Vol.49, pp. 344-352, 2018.

[13] S. Mauro, L. S. Scimmi, and S. Pastorelli, "Collision Avoidance Algorithm for Collaborative Robotics," In Int. J. Automation Technol., Vol.11, No.3, pp. 481-489, 2017.