2021 20th International Conference on Advanced Robotics (ICAR)
December 6-10, 2021. Ljubljana, Slovenia

2021 20th International Conference on Advanced Robotics (ICAR) | 978-1-6654-3684-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICAR53236.2021.9659446

Vision based robot-to-robot object handover

Monica Sileo[†], Michelangelo Nigro[†], Domenico D. Bloisi[⋆], Francesco Pierri[†]

Abstract— **This paper presents an autonomous robot-to-robot object handover in the presence of uncertainties and in the absence of explicit communication. Both the giver and receiver robots are equipped with an eye-in-hand depth camera. The object to handle is roughly positioned in the field of view of the giver robot's camera and a deep learning based approach is adopted for detecting the object. The physical exchange is performed by recurring to an estimate of the contact forces and an impedance control, which allows the receiver robot to perceive the presence of the object and the giver one to recognize that the handover is complete. Experimental results, conducted on a couple of collaborative 7 DoF manipulators in a partially structured environment, demonstrate the effectiveness of the proposed approach.**

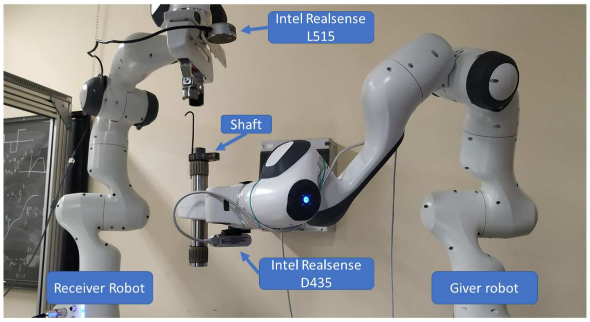


Fig. 1. Experimental setup. Two Panda Emika Franka robots equipped with depth cameras are used for handing over a shaft.

I. Introduction

The fourth industrial revolution, often referred to as Industry 4.0, is focused on a next generation smart factories with highly flexible and reconfigurable facilities, characterized by more autonomous, safe, and effective robotic systems. Thus, robots are required to cope with complex tasks in unstructured environments by leveraging on learning capabilities. The object handover is a typical industrial task involving cooperative robots, e.g. in logistic applications, where robots are widely adopted in picking operations. Nowadays, collaborative robots are often used in handover with humans [1], and, in quasi-autonomous production plants, even with robots, e.g., a logistic robot and an assembler one. Object handover can be partitioned in two phases, the pre-handover and the physical exchange phase. The pre-handover phase includes the object detection, in which the giver must recognize the presence of the object to hand over in its workspace, the object grasping, the transportation and the synchronization, i.e., finding an agreement about the exchange location and timing [2]. In recent years, interest in object detection has burst due to the rapid development of deep learning techniques [3], largely applied to robot vision. One of the most common deep neural networks is the Convolutional Neural Network (CNN), that represent the best trade-off among accuracy and the detection speed. Regarding the synchronization phase, it can require the giver robot to explicitly or implicitly communicate to the receiver. Explicit communication implies to share both sensory data and control signals among the robots, while implicit communication occurs when information is acquired only by sensors attached on the robot, e.g. via a force/torque sensor measuring the interaction forces, tactile sensors and/or visual sensors [4]. Explicit communication has largely been adopted for cooperative robots, since it allows to easily handle synchronization issues. However, in the presence of many and heterogeneous agents, the communication network complexity and load can increase. Even in an industrial scenario with only few manipulators involved, the communication channel, due to other devices connected, can experience packet loss and delays, which are detrimental to performance and can even cause production scraps. The use of an implicit communication, even if the control scheme becomes usually more complex and the performance worse, improves the flexibility and the scalability of the system.

The physical exchange phase starts at the instant of the first contact between the receiver robot and the object grasped by the giver robot and ends when the giver fully releases the object to the receiver [1]. The physical exchange requires cooperation among the giver and receiver robots, thus vision and force feedback can be adopted by the giver in order to understand if the receiver has grasped the object. Only when the grasping is safe the giver can start to release the object and allow the transition to the receiver.

In this paper, a fully autonomous cooperative robotic object handover strategy, able to cope with large errors on the object pose and without explicit communication, is presented. The focus is both on the object detection and on grip force modulation during the transition from the giver to the receiver. The object detection phase is handled by using a learning approach, in particular two different CNNs have been trained and compared on the same training and test data. Regarding the physical exchange phase, both visual sensors and a force estimator based on joint torque sensors are adopted in order to detect the exchange point and to modulate the grip force. More in detail, the first contact of the receiver with the object is detected by resorting to the force estimator, while the giver opens the gripper when the estimated force exerted by the receiver reaches a certain threshold. The proposed approach has been experimentally validated by considering two collaborative robot manipulators Franka Emika Panda, both equipped with a camera in eye-in-hand configuration. The camera on the giver robot

[†] School of Engineering, University of Basilicata, 85100 Potenza, Italy.
[⋆] Department of Mathematics, Computer Science, and Economics, University of Basilicata, 85100 Potenza, Italy.
Corresponding author: monica.sileo@unibas.it

This research has been supported by the project ICOSAF (Integrated collaborative systems for Smart Factory - ARS01_00861), funded by MIUR under PON R&I 2014-2020.

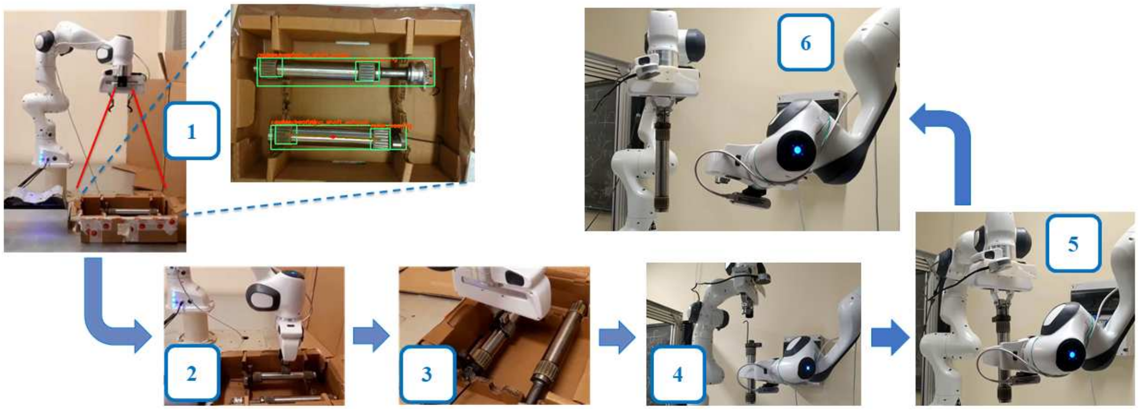


Fig. 2. Functional scheme. 1) Object detection phase. 2) Estimation of the grasping point. 3) Object grasping and motion to the exchange point. 4) Detection of the giver at the exchange position. 5) Object grasped by the receiver. 6) Object released by the giver.

is an Intel Realsense D435, while the one mounted on the receiver robot is an Intel Realsense L515 (see Fig. 1).

II. Related Work

The object detection phase concerns finding instances of objects of interest in a image and locating them with bounding boxes [3]. At the beginning of the last decade, Convolutional Neural Networks (CNNs) were firstly applied to object detection [5] and their use was boosted with the introduction of the Region-based CNNs (R-CNNs) [6]. A R-CNN is a *two-stage detector*: in the first stage a set of object proposals (object candidate boxes) is extracted, then, in the second stage, each proposal is fed to a CNN model to extract features and a classifier is used to detect the object and identify its class. The major drawback of R-CNN is the slow detection speed due to the redundant feature computations on the overlapped object proposals.

Faster R-CNN [7] was the first near-realtime deep learning detector thanks to the introduction of the Region Proposal Network (RPN), which enables nearly cost-free region proposals. Even if Faster R-CNN overcame most of the problems of R-CNN, a further improvement has been achieved with the first *one-stage detector*, named You Only Look Once (YOLO) [8]. Unlike Faster R-CNN, YOLO partitions the image into regions and predicts bounding boxes and probabilities for each region at the same time. The first version of YOLO was characterized by a lower localization accuracy compared with two-stage detectors. However, the new versions of YOLO present an improved detection accuracy still keeping a high detection speed [9].

Regarding the object handover, during the physical exchange phase, the object load is shared by the giver and the receiver and they must guarantee the object safety. Different studies have been conducted for the force exchanged by humans during handover operations. For example, in [10] it is found that the grip force of both giver and receiver is modulated during the object exchange, i.e., while the giver decreases its force, the receiver increases it until the load is transferred. Then, after the unloading the giver still applies a grasping force even though its sensed load is almost zero [11]. These results can be also applied to robot-to-robot handovers, where techniques for grip force modulation must be proposed. In [12], the sole communication mean between the two agents is provided by custom force/tactile sensors measuring the interaction force and moment. In their approach, the giver adopts a slipping detection algorithm that allows to foresee the possibility that the receiver cannot keep the object orientation and thus dangerous releases are avoided.

III. Proposed strategy

The considered task is the autonomous robot-to-robot object handover, in the absence of any explicit communication. The manipulated objects are a couple of counter-rotating shafts of different lengths and shapes. It is assumed that the objects are placed in a box roughly positioned in the field of view of the giver's camera. Due to the position uncertainties, the robot motion cannot be offline planned and a vision system is adopted in order to detect the presence, identify the class, and compute the pose of the objects.

The following strategy (shown in Fig. 2) is proposed:

1) When the box containing the objects is completely in the camera field of view of the giver robot, a CNN detects the presence, the orientation and the type of the shafts. The CNN recognizes also the roller bearings, that are used to determine the shaft axes and estimate the grasping pose.
2) The position of the grasping point is estimated by computing the center of the roller bearing bounding boxes and the shaft axis.
3) The giver grasps the shaft and moves it to the exchange point, assumed within the field of view of the receiver's camera.
4) The receiver robot recognizes, by using an eye-in-hand camera and a marker attached to the giver gripper, the giver pose and, thus, the object exchange point.
5) The receiver aligns its gripper to the shaft axis and moves toward the object until a contact is detected, then it closes the gripper.
6) The receiver moves the grasped shaft while the giver compliantly follows its motion. When the force exerted on the giver exceeds a threshold, the gripper is open and the shaft is released.

IV. Shaft detection

To carry out the shaft detector, two CNN-based approaches, namely Faster R-CNN and YOLOv4, have been investigated to compare their performance and select the best architecture.

A. Faster R-CNN

Faster R-CNN [7] is composed by two modules:

1) The Regional Proposal Network (RPN), which is a CNN that outputs the region proposals with the highest probability of object presence.
2) The Fast R-CNN detector. The regions generated by the RPN are fed to the Fast R-CNN detector in order to refine them and to determine the class membership of the object.

In this work, the Inception v2[1] model has been used since, thanks to the use of 1×1 convolution, the Inception network generates a reduced model size, which, in turn, can help to reduce the overfitting problem.

B. YOLOv4

YOLO [8] is a popular object detector based on Darknet, which is an open source neural network framework written in C and CUDA. The main feature of YOLO is its capability of making predictions considering object detection as a single regression problem. YOLOv4 is the latest Darknet based implementation currently available (February 2021) and it improves the previous YOLOv3's average precision of about 10%. YOLOv4 [9] consists of:

1) Backbone: CSPDarknet53 [13], which augments the learning capacity of CNN;
2) Neck: Spatial Pyramid Pooling additional module [14], PANet path-aggregation [15];
3) Head: YOLOv3 [16].

C. Dataset and Training

In the considered application, three object classes have been defined (see Fig. 3(a)):

- Counter-rotating shaft exhaust (CSE);
- Counter-rotating shaft intake (CSI);
- Roller bearing (RB).

Detecting CSE and CSI is useful to identify the shaft's type, while RB is used to determine the shaft axes and estimate the grasping point. A dataset made of 342 images, of size 640×480, has been built by considering two shafts, captured at different distances and with different orientation and background (see Fig. 3(b)).

The dataset has been manually annotated (using the LabelImg tool) and then split in two non-overlapping sets, namely the training (245 images) and test (97 images) sets. A data augmentation step has been carried out to create a larger training set. In particular, for each image, horizontal flipping, cropping, and zooming have been performed. The augmented training dataset, made of 980 images, has been used to train both the Faster R-CNN and YOLOv4 networks.

The annotated training and test data can be downloaded at the following URL: `https://tinyurl.com/36rjp2qy`

An Intel Xeon 3.7 GHz CPU 32 GB RAM with a NVIDIA Quadro P4000 8GB GPU has been used to carry out the training phase, which required about 8 hours for the Faster R-CNN and 14 hours for the YOLOv4.

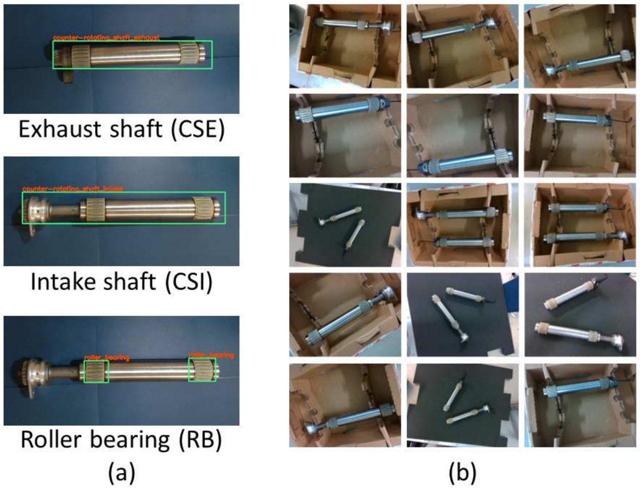


Fig. 3. Our image dataset. a) Object classes of interest. b) Samples with different orientation and background).

D. Detection Results

To evaluate the detection performance, the mean average precision (mAP) metric has been considered [17]. The Average Precision (AP) for each class represents the integral of the precision-recall curve, measured for a certain value of the Intersection over Union (IoU). The Intersection over Union measures the overlap between the predicted bounding box and the labelled one. The mAP is the AP averaged over all classes. Table I shows detection results on the test set for a value of IoU of 0.5.

The detector is implemented in C++ and it runs on 640×480 images coming from the sensor mounted on the end-effector of the giver robot via the `librealsense2` library. The detection process takes on GPU an average time per image of 58 ms for the Faster R-CNN and 44 ms for the YOLOv4. On the basis of these results and performance, the YOLOv4 architecture has been chosen to implement the shaft detector.

TABLE I
AVERAGE PRECISION FOR IOU VALUE OF 0.5

Network Architecture	CSE	CSI	RB	mAP
YOLOv4	**0.952**	**0.966**	**1.000**	**0.973**
Faster R-CNN	0.928	0.855	**1.000**	0.914

V. ESTIMATION OF THE GRASPING POINT

Once the shaft has been detected in the image, it is necessary to compute the position of the grasping point and the shaft orientation (see Fig. 4). The latter is crucial to compute the gripper orientation since the shaft must be grasped by the receiver always from the cogwheel. In order to identify the orientation, the distance, δ (see Fig. 4), between the top-left corner of the CSI bounding box and the edge of the left RB bounding box has been computed.

It is assumed that, for both the shafts, a suitable grasping point should be located along the longitudinal axis. More in detail, the best grasping point for the CSE is equidistant from the center of the two roller bearings, while for the CSI, it is closer to the cogwheel. A two step procedure

[1] https://github.com/Khaivdo/How-to-train-an-Object-Detector-using-Tensorflow-API-on-Ubuntu-16.04-GPU

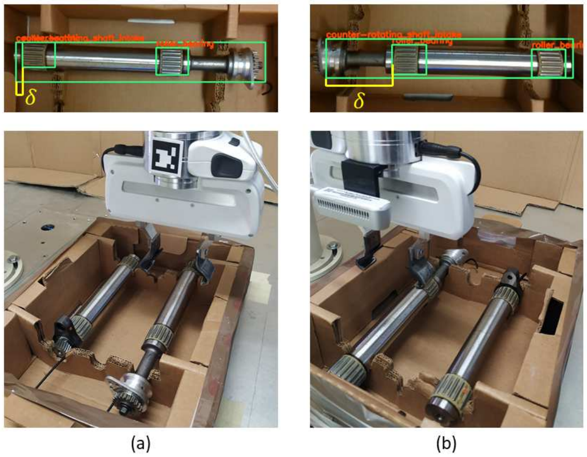


Fig. 4. Two possible configurations of the shafts. The distance δ is computed in order to detect the shaft orientation.

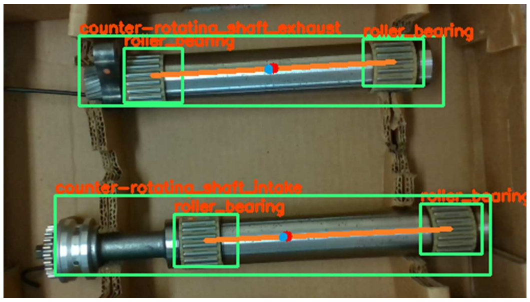


Fig. 5. Shafts axes and grasping points: actual ones in blue and estimated ones in red.

is used to estimate the position of the grasping points: 1) the shaft axis is estimated by connecting the centers of the two RB bounding boxes, 2) the grasping point is computed along this axis (see Fig. 5). It is worth noticing that, due to possible variations in the detected bounding boxes size, the grasping point estimation could be not perfect. However, such uncertainties will be managed in the physical exchange phase by ensuring a suitable compliance to the robots.

The grasping point is detected in the image frame and, in order to perform the grasp, must be transformed in a 3D reference position, expressed in the robot base frame. To this aim, the RGB camera has been calibrated with a Direct-Linear-Transformation (DLT) method [18] using a 3D target.

VI. Shaft handover

Once the grasping point position and the shaft orientation have been detected, the giver robot can be commanded to grasp the shaft in such a way to align the x_e axis of its end-effector reference frame (see Fig. 6) with the shaft axis, and move it to the exchange point. The motion is performed by using a closed-loop inverse kinematics algorithm [19]. The exchange point is off-line planned on the basis of work-cell configuration in such a way the marker attached on the giver's gripper is in the field of view of the receiver's camera. It is worth noticing that the planned point is related to the end-effector reference frame, while the shaft tip position is unknown, due to the uncertainties of the grasping point estimation and the different length of the two shafts. Moreover, due to the absence of communications, the exchange point is unknown to the receiver, thus its motion cannot be planned off-line and performed via a pure positional control.

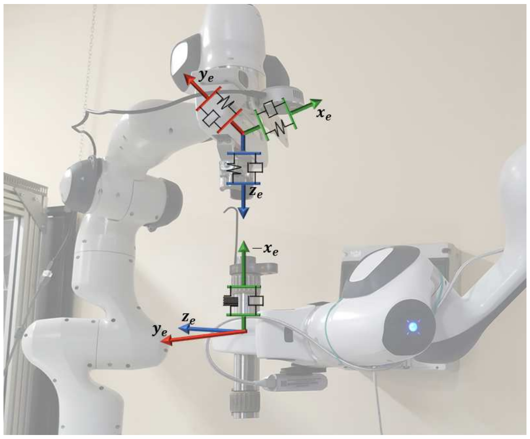


Fig. 6. Reference frames for the robots' end-effectors.

The receiver needs an exteroceptive sensor to detect the presence of the shaft and to align its gripper to the shaft axis. To this aim, it is equipped with a Intel Realsense L515 camera, while an Aruco marker [20] is positioned on the giver robot (Fig. 7). In order to avoid collisions, the receiver can start its motion only when the giver reaches the exchange point. Therefore, the camera detects the position of the center of the marker, by using the `OpenCV` library, in two consecutive frames: if the difference between the detected positions is below a certain threshold for at least 0.5 s, the robot assumes that the giver has reached the exchange position and starts its motion. Firstly, the receiver aligns the x_e and y_e axis of its end-effector reference frame (see Fig. 6) to the marker frame, then, since the shaft position with respect to the marker is fixed and assumed known, moves the end-effector in order to align the z_e axis with the shaft axis. Finally, it moves along the shaft axis and stops when a contact is detected. Since wrist mounted force-torque sensors are not present on the Franka Emika Panda robots, the contact detection is provided by a momentum-based observer [21].

A. Contact estimation

The wrench acting on the end-effector is estimated via an observer, based on the generalized momentum

$$\boldsymbol{\nu} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}\,, \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the robot inertia matrix and $\boldsymbol{q}$ ($\dot{\boldsymbol{q}}$) is the vector of joint positions (velocities). By exploiting the robot dynamic model and the property of the inertia matrix [19]

$$\dot{\boldsymbol{M}}(\boldsymbol{q}) = \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{q},\dot{\boldsymbol{q}}), \tag{2}$$

where $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is the matrix collecting the Coriolis and centrifugal terms, the time-derivative of (1) can be expressed as

$$\dot{\boldsymbol{\nu}} = \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{F}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{\tau} + \boldsymbol{\tau}_e, \tag{3}$$

where $\boldsymbol{g}(\boldsymbol{q})$ is the vector of gravity terms, $\boldsymbol{F}(\boldsymbol{q})$ is the matrix of the viscous friction terms, $\boldsymbol{\tau}$ is the vector of joint torques and $\boldsymbol{\tau}_e$ are the torques induced at the joints by the contact

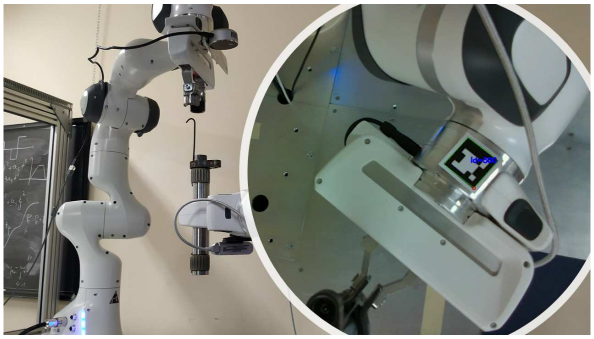


Fig. 7. Marker detection performed by the receiver robot to detect the presence of the giver one.

wrench $\boldsymbol{h}$. An estimate of $\boldsymbol{\tau}_e$ can be computed as

$$\hat{\boldsymbol{\tau}}_e = \boldsymbol{K}_o \Bigg[(\boldsymbol{\nu}(t) - \boldsymbol{\nu}(t_0)) + \qquad (4)$$
$$- \int_{t_0}^{t} (\boldsymbol{C}^{\mathrm{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{F}\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau} + \hat{\boldsymbol{\tau}}_e) d\varsigma \Bigg],$$

where t and t_0 are the current and initial time instant respectively, and $\boldsymbol{K}_o \in \mathbb{R}^{n \times n}$ is a positive definite gain matrix. In (4), the dynamic parameters identified for the Franka Emika Panda in [22] have been used, suitably modified to take into consideration the contribution to the inertia and gravity terms of the gripper and, only for the giver robot, of the shafts. The estimation dynamics can be easily derived as

$$\dot{\hat{\boldsymbol{\tau}}}_e + \boldsymbol{K}_o \hat{\boldsymbol{\tau}}_e = \boldsymbol{K}_o \boldsymbol{\tau}_e, \qquad (5)$$

which is a first-order low-pass dynamic system. Under the assumption of constant or slowly variant contact wrench, $\hat{\boldsymbol{\tau}}_e \rightarrow \boldsymbol{\tau}_e$ when $t \rightarrow \infty$ for any positive definite gain matrix $\boldsymbol{K}_o$. Finally, an estimate of the external wrench can be obtained as [23]

$$\hat{\boldsymbol{h}} = \boldsymbol{J}^{\dagger \mathrm{T}}(\boldsymbol{q})\hat{\boldsymbol{\tau}}_e, \qquad (6)$$

where $\boldsymbol{J}^{\dagger}$ is the right pseudo-inverse of the robot Jacobian matrix. Finally, in order to suppress non-existent small force and torque estimations owing to unmodeled dynamics and sensor noise, a *dead zone* has been implemented, i.e., any value of force component below 3 N and any value of moment below 1 Nm estimated by the observer are neglected. Moreover, to achieve a continuous wrench signal, the same thresholds have been subtracted from higher estimations.

B. Physical exchange phase

During the physical exchange phase, different behaviors for the giver and the receiver are required. When the receiver hits the object, a force along the shaft axis is perceived by both robots. In this phase, in order to successfully perform the handover, the giver has to keep constant its position and orientation, crucial to make possible the grasping of the receiver, while the receiver must be compliant enough to avoid large contact forces and mechanical stresses on the shaft. Then, after the receiver grasps the object, it moves upward and the giver has to compliantly follow it.

To enforce the desired behaviors to the robots, an admittance control [24] has been implemented.

By defining the planned trajectory as $\boldsymbol{x}_d = [\boldsymbol{p}_d,\ \boldsymbol{\phi}_d]^{\mathrm{T}}$, where $\boldsymbol{p}_d$ and $\boldsymbol{\phi}_d$ are the planned position and orientation, expressed in Euler angles, of the end-effector, a reference trajectory $\boldsymbol{x}_r = [\boldsymbol{p}_r,\ \boldsymbol{\phi}_r]^{\mathrm{T}}$, to be fed to the low-level motion controller, can be computed via the following

$$\boldsymbol{M}_a \boldsymbol{\Delta}\ddot{\boldsymbol{x}}^e + \boldsymbol{D}_a \boldsymbol{\Delta}\dot{\boldsymbol{x}}^e + \boldsymbol{K}_a \boldsymbol{\Delta}\boldsymbol{x}^e = \boldsymbol{S}\boldsymbol{T}_A^{\mathrm{T}}(\boldsymbol{\phi})\hat{\boldsymbol{h}}^e, \qquad (7)$$

where $\boldsymbol{M}_a$, $\boldsymbol{D}_a$ and $\boldsymbol{K}_a$ are, respectively, the virtual inertia, damping and stiffness matrices imposed to the end-effector, $\hat{\boldsymbol{h}}^e$ is the estimated contact wrench (6) expressed in the end-effector frame. $\boldsymbol{\Delta}\boldsymbol{x}^e$ is the difference between the desired and reference pose expressed in the end-effector frame, i.e.

$$\boldsymbol{\Delta}\boldsymbol{x}^e = \begin{bmatrix} \boldsymbol{p}_d^e - \boldsymbol{p}_r^e \\ \boldsymbol{\phi}_{dr} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_e^{\mathrm{T}}(\boldsymbol{p}_d - \boldsymbol{p}_r) \\ \boldsymbol{\phi}_{dr} \end{bmatrix}, \qquad (8)$$

where $\boldsymbol{R}_e$ is the rotation matrix expressing the orientation of the end-effector in the robot base frame, and $\boldsymbol{\phi}_{dr}$ is the vector of Euler angles extracted from the matrix $\boldsymbol{R}_d^{\mathrm{T}}\boldsymbol{R}_r$, where $\boldsymbol{R}_d$ and $\boldsymbol{R}_r$ are the rotation matrices expressing the planned and reference orientation, respectively. The matrix $\boldsymbol{T}_A^{\mathrm{T}}(\boldsymbol{\phi})$ in (7) is defined as

$$\boldsymbol{T}_A^{\mathrm{T}}(\boldsymbol{\phi}) = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{T}(\boldsymbol{\phi}) \end{bmatrix},$$

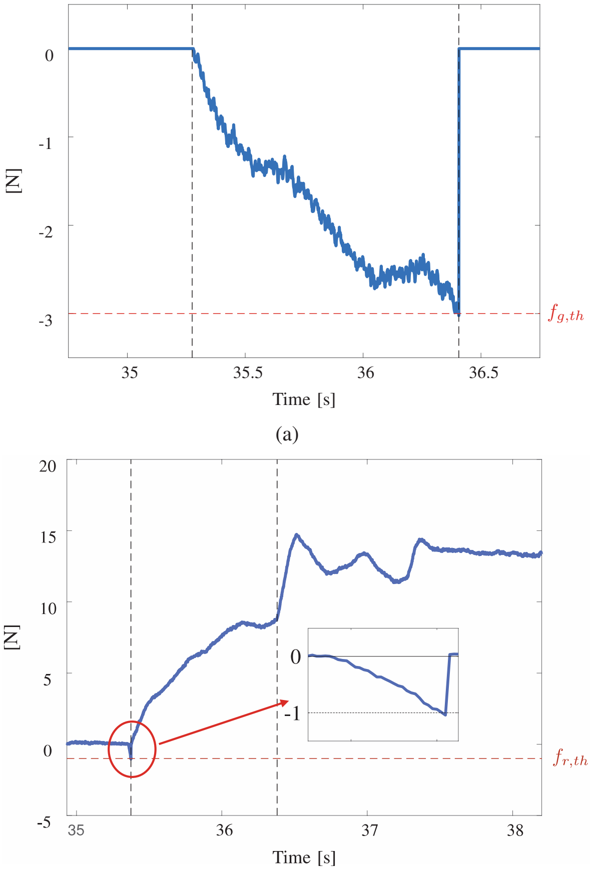
where $\boldsymbol{T}(\boldsymbol{\phi})$ is the matrix that maps the time derivative of the Euler angles $\boldsymbol{\phi}$ of the end-effector to the angular velocity [19], $\boldsymbol{I}_3$ and $\boldsymbol{O}_3$ are the 3×3 identity and null matrices, respectively. Finally, $\boldsymbol{S}$ is a (6×6) diagonal selection matrix of ones and zeros, whose (i, i) element is 0 (1) if the robot must be rigid (compliant) with respect to the i-th component of the wrench $\hat{\boldsymbol{h}}^e$. For the receiver robot, the matrix $\boldsymbol{S}$ has been set as $\boldsymbol{S}_r = \boldsymbol{I}_6$, while, for the giver, in order to enforce the above described behavior, it has been set as a matrix with all zeros except for the element (1,1) given by

$$\boldsymbol{S}_g(1,1) = \frac{1 - \mathrm{sgn}(\hat{f}_{g,x}^e)}{2}, \qquad (9)$$

where $\mathrm{sgn}(\cdot)$ is the sign function and $\hat{f}_{g,x}^e$ is the estimated force acting on the giver end-effector along the axis x_e expressed in the robot end-effector frame. In other words, such a condition means that the giver robot is commanded to be compliant with respect to force along the axis $-x_e$ and rigid with respect to other forces and moments (see Fig. 6).

The other matrices in (7) have been set as $\boldsymbol{M}_a = \mathrm{diag}\{15\boldsymbol{I}_3, 0.5\boldsymbol{I}_3\}$, $\boldsymbol{D}_a = \mathrm{diag}\{30\boldsymbol{I}_3, \boldsymbol{I}_3\}$ and $\boldsymbol{K}_a = \mathrm{diag}\{45\boldsymbol{I}_3, 1.5\boldsymbol{I}_3\}$.

Fig. 8 reports the estimated forces acting on the two robots along the shaft axis. In detail, the receiver detects the first contact when the force $\hat{f}_{r,z}$, i.e. the estimated force acting on the receiver end-effector along the axis z_e, overcomes a threshold, $f_{r,th}$, that has been set as -1 N (see Fig. 8(b)). Once the contact is detected, the receiver closes the gripper and starts to move compliantly followed by the giver robot. During this phase, a force along the shaft axis, $\hat{f}_{g,x}$, is experienced on the giver's end-effector (Fig. 8(a)), it decreases until the threshold $f_{g,th} = -3$ N is reached. At the same time, the force $\hat{f}_{r,z}$ increases until reaching the gravitational force due to the shaft's mass (Fig. 8(b)). Once the threshold is reached, the giver opens the gripper and the object is fully released to the receiver. A video of the whole task can be



(a)

(b)

Fig. 8. Estimated contact forces along the shaft axis: (a) force $\hat{f}^{e}_{g,x}$ acting on giver robot, (b) force $\hat{f}^{e}_{r,z}$ acting on receiver robot. Vertical dashed lines delimit the physical exchange phase. Horizontal dashed red lines represent the thresholds.

found at: `https://youtu.be/bLfm3qG2ooE`.

The results for the CSE handover are not reported for the sake of brevity, since they are analogous to those showed above.

VII. Conclusions

An approach for achieving autonomous execution of robot-to-robot object handover task has been developed and experimentally validated in a partially structured environment and in the absence of explicit communication between the robots. The proposed approach requires only visual and joint torque sensors and can be easily extended to industrial scenarios for flexible production. More in detail, visual measures are adopted for detecting the presence of the object both in the giver and in the receiver workspace, while an observer which exploits the joint torque measurements is adopted for modulating the grip force of the two robots. The proposed strategy can be exploited for different objects by extending the classes detected by the CNN trained on a new dataset with other images and by defining different grasp poses. Future work will be devoted to involve a human operator in the task, and extend the approach to mobile manipulators.

References

[1] V. Ortenzi, A. Cosgun, T. Pardi, W. P. Chan, E. Croft, and D. Kulić, "Object handovers: a review for robotics," *IEEE Trans. on Robotics*, 2021.

[2] R. Koeppe, D. Engelhardt, A. Hagenauer, P. Heiligensetzer, B. Kneifel, A. Knipfer, and K. Stoddard, "Robot-robot and human-robot cooperation in commercial robotics applications," in *Robotics research. the eleventh international symposium*. Springer, 2005, pp. 202–216.

[3] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055*, 2019.

[4] A. Tsiamis, C. K. Verginis, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Cooperative manipulation exploiting only implicit communication," in *2015 IEEE/RSJ Int.Conf. on Intelligent Robots and Systems (IROS)*, 2015, pp. 864–869.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 142–158, 2015.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[10] A. H. Mason and C. L. MacKenzie, "Grip forces when passing an object to a partner," *Experimental brain research*, vol. 163, no. 2, pp. 173–187, 2005.

[11] W. P. Chan, C. A. Parker, H. M. Van der Loos, and E. A. Croft, "Grip forces and load forces in handovers: implications for designing human-robot handover controllers," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, 2012, pp. 9–16.

[12] M. Costanzo, G. De Maria, and C. Natale, "Handover control for human-robot and robot-robot collaboration," *Frontiers in Robotics and AI*, vol. 8, p. 132, 2021.

[13] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390–391.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[15] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[16] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[17] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41–65, 2018.

[18] Y. Abdel-Aziz, H. Karara, and M. Hauck, "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry," *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 2, pp. 103–107, 2015.

[19] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics – Modelling, Planning and Control*. London, UK: Springer, 2009.

[20] A. Babinec, L. Jurišica, P. Hubinskỳ, and F. Duchoň, "Visual localization of mobile robot using artificial markers," *Procedia Engineering*, vol. 96, pp. 1–9, 2014.

[21] A. De Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *2005 IEEE Int.Conf. on Robotics and Automation*, 2005, pp. 999–1004.

[22] C. Gaz, M. Cognetti, A. Oliva, P. R. Giordano, and A. De Luca, "Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4147–4154, 2019.

[23] F. Ficuciello, A. Romano, L. Villani, and B. Siciliano, "Cartesian impedance control of redundant manipulators for human-robot co-manipulation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2120–2125.

[24] L. Villani and J. De Schutter, "Force control," in *Handbook of Robotics*, Springer-Verlag, B. Siciliano, and O. Khatib, Eds., 2008.