Mixed-test method for performance evaluation of intelligent collaborative robotic systems

Miguel Da Silva, Maria Makarov, Remi Regnier, Didier Dumur

Mixed-test method for performance evaluation of intelligent collaborative robotic systems

Miguel Da Silva[1,2], Maria Makarov[2], Remi Regnier[1], Didier Dumur[2]

***Abstract*— Human-centered robotic systems open a large field of new applications, both in industry and service contexts. For their interaction with human beings, up to physical collaboration, they rely heavily on computer vision, and more recently on human motion tracking algorithms, which are examples of intelligent components. The complexity resulting from the variety of human behaviors and the combination of intelligent components with robotic collaborative tasks raises the problem of the performance evaluation of the overall system. To support experiment design for performance evaluation of intelligent collaborative robotic systems, we propose an approach combining real-world human motion recordings with numerical simulations of the dynamics of the robotic system with its controller. In this article, we illustrate this approach on the example of the handover task.**

I. INTRODUCTION

The integration of intelligent technologies into manufacturing processes has become essential for improving efficiency, flexibility, and overall productivity. Intelligent components, such as computer-vision-based human motion recognition, are central for the integration of intelligent collaborative robotic systems, where humans are part of the loop. Collaborative robots, or cobots, have emerged as a key element in this paradigm shift, redefining human-robot collaboration in industrial environments.

Evaluating the performance of cobots is essential to ensure that they operate safely, efficiently, and effectively alongside human workers, maximizing productivity while minimizing the risk of accidents and downtime. In [1], we presented the four types of testing methods: analytical methods, virtual tests, physical tests, and mixed tests. *Analytical methods* are useful to evaluate the elementary functionalities of cobots using simplified models or to prove the stability of the cobot in a given test configuration. *Virtual tests* allow the prediction of cobot behavior in complex scenarios. They allow us to numerically solve problems that are analytically difficult or even impossible to solve including those that use artificial intelligence. However, these tests rely on numerical models that have been built based on assumptions, and which may therefore neglect certain aspects of the task and lack realism in certain situations. *Physical tests* can overcome this issue but the number of use scenarios that can be assessed is limited because of the costs and the test time. *Mixed tests* are a trade-off between *physical tests* and *virtual tests*. For example, [2] create a real-time interaction between the human operator and a virtual cobotic task, using a body tracking system and a virtual reality headset. The *mixed tests* method we propose for evaluating the system's performance is based on a camera- and human-in-the-loop approach.

Despite these different testing possibilities, as stated by [3], no standardized evaluation methods guaranteeing the performance of intelligent human-robot collaborative systems exist. The ISO:15066 standard, analyzed in [4], focuses on safety but not performance evaluation. One of the main challenges results from non-deterministic intelligent components (e.g. cameras with human tracking algorithms) which can lead to unpredictable behaviors of the collaborative task.

To address this issue, we have developed a mixed-test methodology for evaluating the performance of a collaborative robotic system including intelligent components (e.g. cameras with human tracking algorithms). The objective is to foresee corner cases and to narrow down the experimental tests to be conducted on the whole system with humans. The handover task is selected for illustrative purposes since it is integrated into a large number of applications ranging from pick and place to collaborative assembly [5]. In that sense, the developed approach can be adapted for other applications.

In this paper, the considered AI system is a body-tracking algorithm. The reason of this choice is that 3D (three-dimensional) cameras with human motion tracking algorithms are increasingly being used for collaborative robotics tasks [6]. For example, [7] uses a depth camera together with an advanced hand-tracking algorithm to solve grasping problems to perform a handover (e.g. object transfer) task.

The proposed mixed-tests approach uses data from physical sensors and human interaction as input to a numerical simulation of the task, to predict its performance. It has the advantage of eliminating the need to model the unpredictable elements of the task (e.g. output of the 3D camera's wrist-tracking AI algorithm). In this way, the task can be simulated, as the data for the most uncertain elements of the task are measured. However, this approach requires the creation of a database of prepared signals for upstream numerical simulation. Creating this database requires defining the test conditions in advance. In other words, we need to know what types of task parameters can be controlled and select those relevant to the actual use scenarios.

In this study, we present a non-real-time mixed tests evaluation approach applied to a handover task. Our contributions are threefold. First, we introduce a model of the collaborative handover task (Fig. 1) that can be adapted for other collabora-

[1]Laboratoire National de métrologie et d'Essais (LNE), 78197 Trappes, France miguel.dasilva@centralesupelec.fr; remi.regnier@lne.fr

[2]Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France maria.makarov@centralesupelec.fr; didier.dumur@centralesupelec.fr

tive tasks by applying the following steps: scenario selection, database creation, data pre-treatment, task simulation, and performance assessment. Second, we illustrate this evaluation method by applying it to a handover task and following the previously enumerated steps. Finally, we analyze the evaluation results and conclude on the validity of this method and the perspectives for its improvement.

II. EVALUATION METHOD

This section details the main components of the proposed camera- and human-in-the loop evaluation method (Fig. 1), which are the human movement database creation (II-A), data pre-treatment (II-B), robot simulator (II-C) and performance metrics definition (II-D).

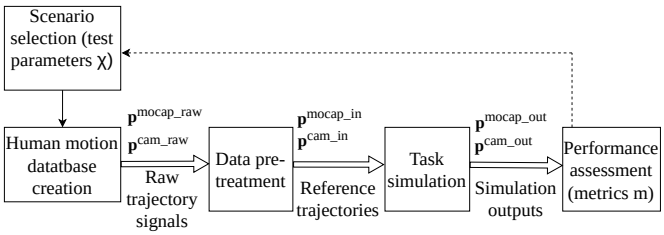

Fig. 1. Mixed-test workflow: acquisition of wrist trajectories for handover movements and data pre-treatment to build mirror trajectories used as a reference for the robot controller to simulate the approach phase of a handover task (Fig. 2,b).

A. Creation of a database of human movement

In this section, the creation of a database of human wrist movement is exemplified in a mock-up of a handover task.

The handover task [5] consists of three phases (Fig. 2): a/ *the user's intention detection phase* which triggers the start of the task when the user's wrist enters the object exchange zone; b/ *the approach phase* during which the human directs her/his hand towards the robot to receive or give an object; c/ *the physical object exchange phase* during which the human takes the object from the robot gripper or gives an object to the robot. This paper exclusively focuses on the approach phase, as it contains artificial intelligence bricks as well as indirect collaboration (i.e. without physical contact) with the human. This subtask is a minimal task that contains elements of a stochastic nature sufficient to highlight the difficulties of evaluating an intelligent collaborative system. Various strategies are possible for this approach phase. The one we have chosen is the "mirror" approach similar to [8], which consists of using as robot reference trajectory the symmetrical trajectory of the human wrist with respect to the object's exchange plane.

To achieve this task, we use a 3D camera with a body tracking algorithm to acquire the position of the human's wrist. The experiments were conducted within a motion capture arena (Fig. 3), in a mock-up of the handover task where the human participant executed reaching movements from initial positions to several targets, in multiple experimental conditions detailed below. Four main scenario parameters were selected in accordance with possible applications of the handover task (e.g. industrial assembly): human body

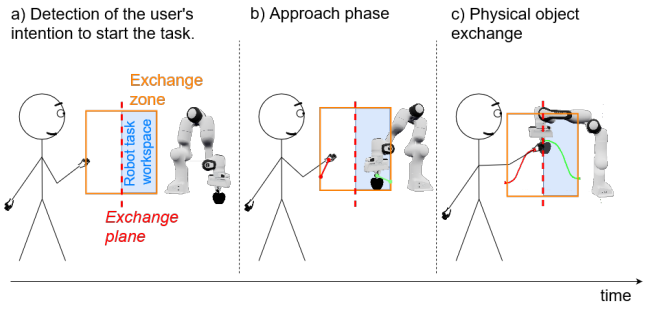

Fig. 2. Description of the different phases of a handover task

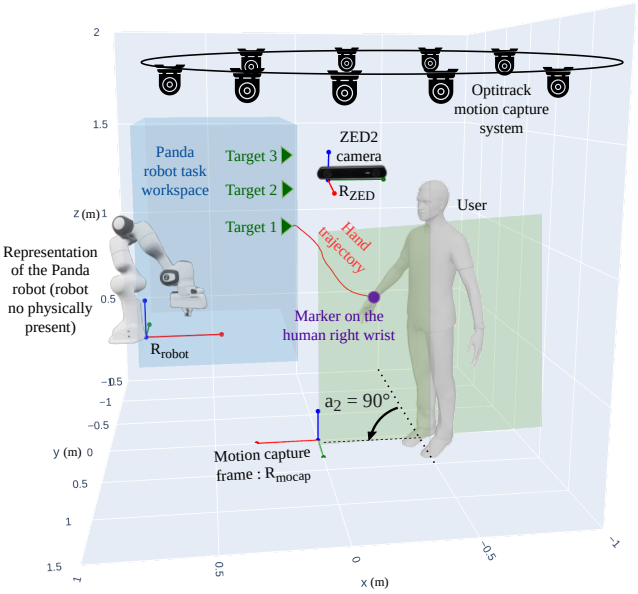

Fig. 3. Experimental setup for recording a handover trajectory. One participant performed a total of 420 handover gestures with 3 different target heights and in different scenarios described in Table II. These trajectories were recorded using a 3D camera and a motion capture system used as ground truth. The trajectories recorded by the motion capture system and the 3D camera were recorded in the frames associated with these respective elements, R_{mocap} and R_{ZED}. The position of the fictitious robot was measured by the motion capture system and used to determine the position of the R_{robot} frame. The axis orientation of each frame follows the standard RGB convention (red: $\boldsymbol{x}$, green: $\boldsymbol{y}$, blue: $\boldsymbol{z}$).

orientation with respect to the camera, walking speed, human posture, and robot position (see Table I). According to [9], the orientation of the human with respect to the camera is responsible for approximately 20% of wrist tracking errors. Therefore, in addition to being consistent with the variability of handover task applications, the choice of this parameter is important because it has a significant impact on the performance of human body tracking. Other important parameters may influence real-world operating conditions and induce variability in human tracking by depth cameras and associated algorithms (e.g. lighting, individual movement strategies, individual morphology etc.).

In Table II, bullets materialize the tested conditions. Some conditions were intentionally not tested because of resulting movements deemed non-natural or not realistic for the handover task. For example, condition RR-U2-a4 would result

TABLE I

HANDOVER TASK SCENARIO PARAMETERS

Parameter description	**Notations and values**
Orientation of human's sagittal plane	$a_1 = 135°, a_2 = 90°$, $a_3 = 0°, a_4 = -45°$, $a_5 = -90°$
Walking speed	**U1**: static ($0\ m/s$) **U2**: walking ($\neq 0\ m/s$)
Human's posture	**S**: sitting **U**: upright (U1 or U2 depending on walking speed)
Robot position from the human's perspective when he faces the 3D camera	**RR**: right **RL**: left

TABLE II

SELECTED SCENARIOS FOR THE HUMAN MOTION RECORDINGS

		a_1	a_2	a_3	a_4	a_5
RL	S	•	•			
	U1	•	•			
	U2	•	•			
RR	S			•	•	•
	U1			•	•	•
	U2				•	•

in a lateral walk pattern that is not natural for humans.

The duration of the trajectories is in the range of $[0.93; 3.2]$ seconds with a mean of 1.85 seconds. The mean maximum cartesian speed of the trajectories is 0.73 m/s. They were not accelerated or slowed down on purpose, our goal was to create these trajectories as close to natural handover movements as possible. The amplitude (e.g. 3D distance between the first and the last point of the trajectory) of the trajectories is between 0.25m and 1.37m, it varies according to the type of scenario described in Table I.

For each repetition, we recorded $\boldsymbol{p}^{\text{mocap}}$ the wrist mocap 3D position at 120Hz (ground truth of human movement), and $\boldsymbol{p}^{\text{cam}}$ the wrist 3D position estimated by the depth camera and human body tracking algorithm at 15Hz, with a ROS1 interface [10]. The motion capture equipment used was an Optitrack system with 16 PrimeX cameras that capture the position of the reflective markers attached to a bracelet placed around the human's wrist, to fake robot and to the 3D camera. The chosen 3D camera is the Stereolabs ZED2 with its dedicated body tracking algorithm.

B. Transforming the raw trajectories into valid robot inputs

Fig. 4 shows examples of the recorded mocap trajectories in the scenario RL-U1-a2. These raw trajectories, together with their equivalents acquired by the camera, are transformed into position reference inputs compatible with the robot (Fig. 1). As previously stated, we exemplify our evaluation approach using the mirrored strategy for the handover approach phase [8]. Main transformations imply frame transformations, mirroring (the robot is supposed to replicate the wrist trajectory symmetrically to the object's exchange plane), and robot workspace compatibility. The

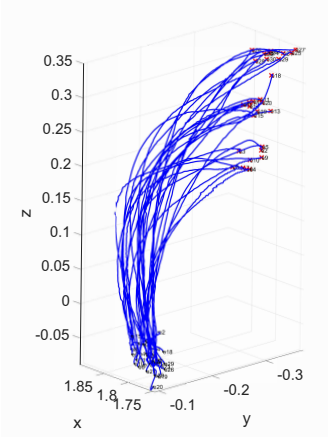

Fig. 4. Handover trajectories recorded with the motion capture system for the scenario RL-U1-a2. There are 30 trajectories, 10 for each target height.

handover task only begins when the human wrist position is inside the exchange zone, and x and y components of the trajectories are bounded to respect the constraints of the robot workspace. Note that by construction of the experimental setup, this situation never happens with the mocap trajectories, and only concerns camera trajectories in case of large depth errors.

C. Impedance-controlled Panda robot simulation

The preprocessed trajectories are used as reference position inputs within a robot simulator to assess the resulting performance. We do not intend to assess the performance of the robot or its control itself, but only the degradation of the performance induced by the use of black-box human motion tracking algorithms based on depth camera measurements. We select a general control framework based on impedance control, well suited for human-robot interaction and largely studied. This type of control law has many variations, such as those using variable gains to improve the compromise between robot end-effector trajectory accuracy and robot flexibility in the event of a collision. In our experiment, the gains are chosen to be constant. We use in this study a simulator of the 7 DOF Franka Emika Panda robot implemented using [11] with the dynamic model identified by [12].

The simulated robot dynamics are:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{v}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} + \boldsymbol{\tau_{ext}} \qquad (1)$$

with $\boldsymbol{q} \in \mathbb{R}^7$ the vector of joint angles and $\boldsymbol{\tau} \in \mathbb{R}^7$ the joint torques, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{7\times7}$ the robot's inertia matrix, $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^7$ the vector of Coriolis and centrifugal effects, $\boldsymbol{g} \in \mathbb{R}^7$ the gravity vector, and $\boldsymbol{v} \in \mathbb{R}^7$ the viscous friction torque. During the approach phase, we consider that no additional external efforts $\boldsymbol{F_{ext}}$ are applied to the robot so that $\boldsymbol{\tau_{ext}} = \boldsymbol{J}(\boldsymbol{q})^{\boldsymbol{T}}\boldsymbol{F_{ext}} = \boldsymbol{0}$ with $\boldsymbol{J}(\boldsymbol{q})$ the Jacobian matrix.

In this study, the impedance control [13] is selected for its easily interpretable trade-off between safety and performance [14] and its suitability for collaborative tasks. The control objective is generally formulated as the following task space dynamics (case without inertia shaping [15]):

$$\boldsymbol{\Lambda}(\boldsymbol{q})\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D_d}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K_d}\tilde{\boldsymbol{x}} = 0 \qquad (2)$$

with $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x_d}$ where $\boldsymbol{x}$ is the robot's end-effector pose in task coordinates, $\boldsymbol{x}_d$ the trajectory to follow, $\boldsymbol{\Lambda(q)} = \boldsymbol{J(q)^{-T}M(q)J(q)^{-1}}$ is the task-space inertia matrix, and $\boldsymbol{K_d} \in \mathbb{R}^{6\times6}$, $\boldsymbol{D_d} \in \mathbb{R}^{6\times6}$ positive definite desired matrices of task-space stiffness and damping if $\boldsymbol{x}$ has a minimal representation (3 rotations and 3 translations). When expressed in the joint coordinates, and considering no anticipation terms, i.e. $\ddot{\boldsymbol{x}}_d = \dot{\boldsymbol{x}}_d = \boldsymbol{0}$, the required joint torque becomes:

$$\boldsymbol{\tau} = \boldsymbol{J(q)}^T(-\boldsymbol{D_d}\dot{\tilde{\boldsymbol{x}}} - \boldsymbol{K_d}\tilde{\boldsymbol{x}}) + \boldsymbol{g(q)} + \boldsymbol{c(q)} \quad (3)$$

Among the numerous variants for implementing impedance control, the one in the joint space was selected for this simulator, similar to [16]:

$$\boldsymbol{\tau} = \boldsymbol{K}(\boldsymbol{q_d} - \boldsymbol{q}) + \boldsymbol{D}(\dot{\boldsymbol{q}}_{\boldsymbol{d}} - \dot{\boldsymbol{q}}) + \boldsymbol{g(q)} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \quad (4)$$

with $\boldsymbol{K} \in \mathbb{R}^{7\times7}$ the matrix of the desired stiffness for each robot joint, $\boldsymbol{D} \in \mathbb{R}^{7\times7}$ the matrix of the desired damping for each robot joint, $\boldsymbol{q_d} \in \mathbb{R}^7$ the vector of desired joint positions and $\dot{\boldsymbol{q}}_{\boldsymbol{d}} \in \mathbb{R}^7$ the vector of desired joint velocities. In our case, we used $\boldsymbol{K} = \text{diag}(100, 100, 100, 100, 100, 100, 8)$ and $\boldsymbol{D} = 2\xi\boldsymbol{K}^{1/2}$ with $\xi = 0.71$ which are the values used by [17] that are suited to applications involving human-robot interactions. Note that the choice of constant joint gains implies position-dependent task-space stiffness and damping, which in turn materializes as position-dependent task-space tracking performance. This serves as an example of non-linear effects in the robot controller and should be kept in mind in the forthcoming analysis. Peter Corke's Python robotics toolbox [11] was used to solve the inverse kinematic (IK) model allowing the transformation of the input trajectories $\boldsymbol{p}_{R_{robot}}^{mocap\_in}$ and $\boldsymbol{p}_{R_{robot}}^{cam\_in}$ from Cartesian space to joint space.

To solve the robot dynamic equation (1) controlled with the control law (4) we used an LSODA Python solver from the SciPy library with a time-step of 1ms. An example of the simulation output is shown in the Fig. 5.

D. Metrics

Fig. 6 lists the most commonly used metrics for intelligent collaborative robotic tasks involving object transfer. This figure aims at analyzing the influence of the individual performance of different subsystems on the complete task. It also shows the different possibilities for error propagation. As we are mainly interested in identifying the effect of errors related to the use of computer vision algorithms in our use case, only those metrics circled in dotted line were considered. To evaluate the performance of the approach phase of a handover task, three different metrics were used: the temporal error, the spatial error, and the robot final pose error. The temporal error is the norm of the difference between the input trajectory and the output trajectory (end-effector trajectory): $\epsilon_{temporal}(t) = \|\boldsymbol{x_d}(t) - \boldsymbol{x}(t)\|$, with $t \in [0; t_{end}]$. It is a common metric to compare two trajectories, for example, [18], [19] used it to compare several human keypoint trajectories from depth camera with ground-truth trajectories from motion capture systems. However, this metric takes into account the potential delay between the

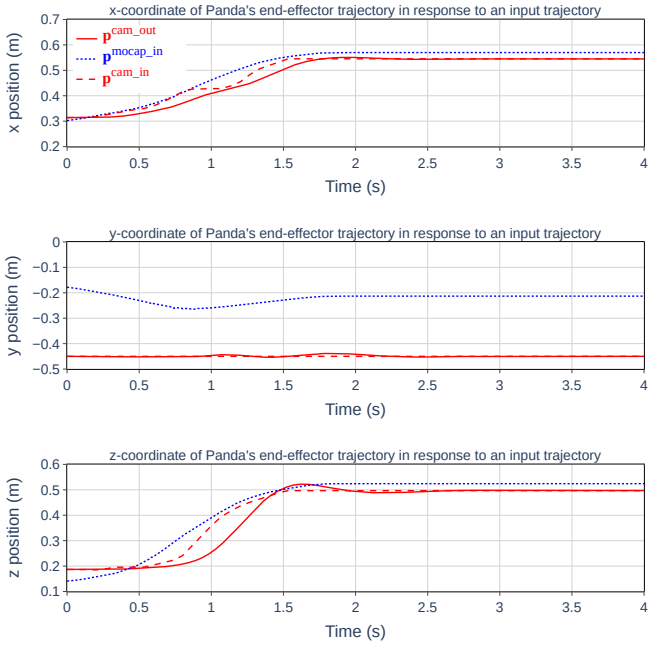

Fig. 5. Simulation of the Panda robot arm with an input trajectory of the scenario RR-U1-a4. Each figure represents one dimension of the input trajectory that was recorded with the ZED2 camera $\boldsymbol{p}_{R_{robot}}^{cam\_in}$, the ground truth input trajectory recorded with the mocap $\boldsymbol{p}_{R_{robot}}^{mocap\_in}$ and the robot end-effector output trajectory $\boldsymbol{p}_{R_{robot}}^{cam\_out}$ in response to the $\boldsymbol{p}_{R_{robot}}^{cam\_in}$ reference trajectory. These three trajectories are expressed in the cartesian space in the R_{robot} frame (see Fig. 1 for the trajectory notations).

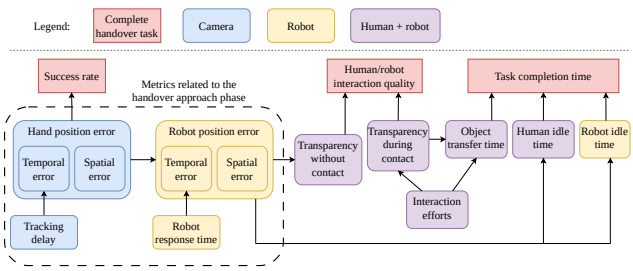

Fig. 6. Metrics for the evaluation of an intelligent collaborative robotic system and its subsystems, in a handover task.

compared trajectories, which means that if the trajectories are similar but not synchronized, the temporal error will be non-zero. The spatial error (Fig.7) is a metric that does not depend on this delay. It is calculated for each point on the robot end-effector trajectory (output trajectory) as the distance between the k-th point of this trajectory and the closest point on the target trajectory: $\epsilon_{spatial}(k) = min(\|\boldsymbol{x}_{\boldsymbol{d}}^i - \boldsymbol{x}^k\|)$ for $i \in [\![k - 100; k + 100]\!]$, with $k \in [\![1; n]\!]$ being the number of the output trajectory point and n being the last one. These two metrics provide information on the robot's behavior during the approach phase of the handover task. If the reference trajectory is too distorted, the human may have difficulty predicting it, and this may have repercussions on the success of the overall task. The last metric we select is the robot final pose error: $\epsilon_{final\_pose} = \|\boldsymbol{x_d}(t_{end}) - \boldsymbol{x}(t_{end})\|$. This metric is important for determining whether the robot's position when exchanging the object is far from that expected

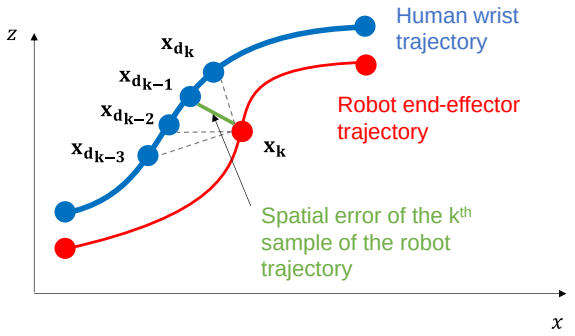

Fig. 7. The spatial error is computed as the shortest distance between the robot end-effector position at time t_k, and the human wrist position from t_{k-100} to t_{k+100}.

by the human.

III. RESULTS

In this section, the proposed approach is analyzed using the acquired dataset of human movement. Fig. 1 shows that the trajectories recorded by the mocap and 3D camera are transformed several times before they can be used as robot input. To investigate how the robot's input error is propagated and reflected in its output, we compare the mocap and ZED trajectories truncated before saturation (Section III-A), as well as the robot end-effector trajectories with the truncated mocap trajectories (Section III-B).

A. Analysis of input trajectories

By analyzing Fig. 8, we can identify 2 distinct scenarios, RR_U1_a4 and RR_U1_a5 that present a higher temporal error ($>$ 10 cm in average). To understand the origin of this error, we visualized these trajectories in 3D (Fig. 9).

We observe that the error is principally caused by a depth estimation error from the camera. It is high in these scenarios because they are the ones in which the amplitude of trajectories in the direction of depth in the 3D camera frame is the highest. Thus, the trajectory orientation relative to the camera is a major influence factor that affects the performance of all the collaborative tasks that rely on AI tracking algorithms from depth cameras.

B. Analysis of simulator outputs

Fig. 10 shows the errors of the output trajectories of the Panda robot's end-effector. When we compare it to Fig.8, we observe that the contribution of the the ZED2 camera tracking algorithm to the temporal error is significant. The contribution of the control algorithm to the temporal error is about 1 cm or 2 cm during the movement. It is a consequence of the robot response time that is slow because of the chosen gains in the robot control which was designed to keep the human-robot interaction safe. The last metric we computed is the final pose error, this one needs to be as low as possible to make the handover task efficient. For example, some industrial handover applications are designed to protect against musculoskeletal disorders, but if the final pose error is too high, the ergonomics of the task may deteriorate. If we exclude the scenario RR_a4_U1 that has very high error

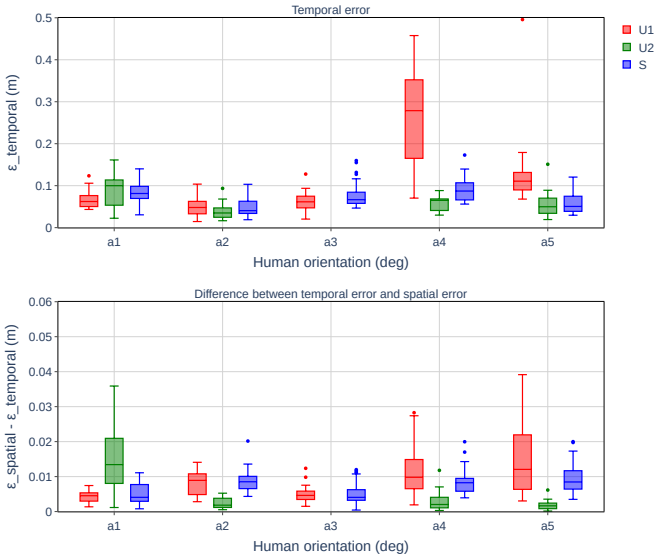

Fig. 8. Temporal and spatial errors between the trajectories $\boldsymbol{p}_{R_{robot}}^{mocap\_in}$ and $\boldsymbol{p}_{R_{robot}}^{cam\_in}$ (see Fig. 1 and Table I for notations).

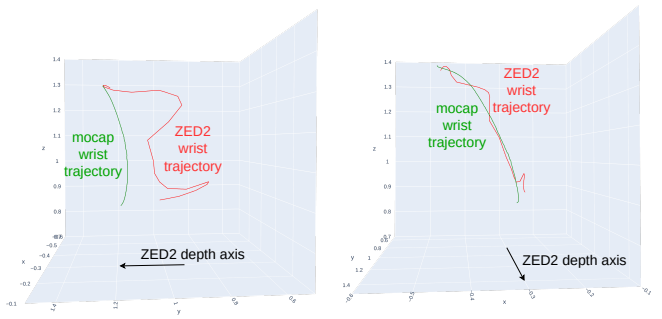

Fig. 9. 3D representation of the ground truth trajectory $\boldsymbol{p}_{R_{mocap}}^{mocap\_raw}$ (green) and the 3D camera trajectory $\boldsymbol{p}_{R_{mocap}}^{cam\_raw}$ (red) for one trajectory of the RR_U1_a4 scenario (see Fig. 1 and Table I for notations).

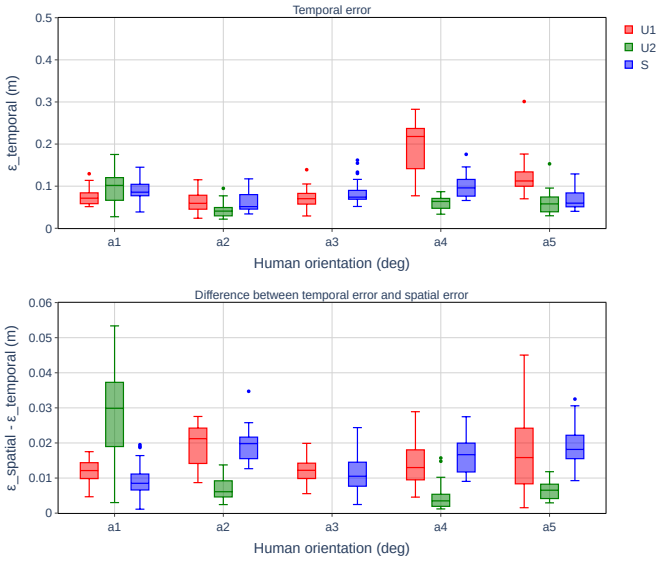

Fig. 10. Temporal and spatial errors between the trajectories $\boldsymbol{p}_{R_{robot}}^{mocap\_in}$ and $\boldsymbol{p}_{R_{robot}}^{cam\_out}$ (see Fig. 1 and Table I for notations).

because of the depths errors as discussed in the previous subsection, the median of the final pose error is between 3 cm and 8 cm which can be acceptable or not according to the chosen handover application and to the human characteristics (eg. height, strength, etc.).

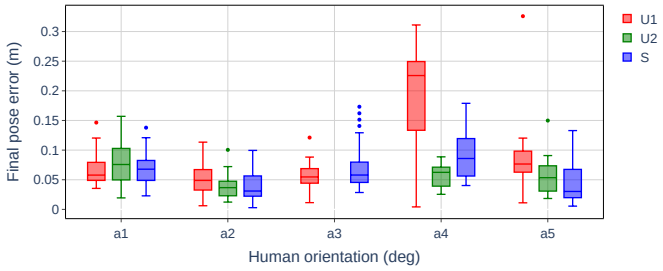

Fig. 11. Final pose error between the trajectories $\boldsymbol{p}_{R_{robot}}^{mocap\_in}$ and $\boldsymbol{p}_{R_{robot}}^{cam\_out}$ (see Fig. 1 for notations).

TABLE III

SUMMARY OF THE MEAN OUTPUT ERRORS

	Temporal error	Spatial error	Final pose error
a1	U1(◇) U2(◇) S(◇)	U1(◇) U2(◇) S(◇)	U1 U2(◇) S(◇)
a2	U1(⋆) U2(⋆) S(⋆)	U1(⋆) U2(⋆) S(⋆)	U1(⋆) U2(⋆) S(⋆)
a3	U1(◇) S(◇)	U1(⋆) S(◇)	U1(⋆) S(⋆)
a4	U1(△) U2(◇) S(◇)	U1(△) U2(◇) S(◇)	U1(△) U2(⋆) S(◇)
a5	U1(◇) U2(◇) S(◇)	U1(◇) U2(⋆) S(⋆)	U1(◇) U2(⋆) S(⋆)
	$\star$: $\bar{\epsilon}_{temporal} \in [0, 0.059]m$	$\star$: $\bar{\epsilon}_{spatial} \in [0, 0.058]m$	$\star$: $\bar{\epsilon}_{final\_pose} \in [0, 0.065]m$
	$\diamond$: $\bar{\epsilon}_{temporal} \in [0.06, 0.118]m$	$\diamond$: $\bar{\epsilon}_{spatial} \in [0.059, 0.115]m$	$\diamond$: $\bar{\epsilon}_{final\_pose} \in [0.066, 0.13]m$
	$\triangle$: $\bar{\epsilon}_{temporal} > 0.118m$	$\triangle$: $\bar{\epsilon}_{spatial} > 0.115m$	$\triangle$: $\bar{\epsilon}_{final\_pose} > 0.13m$

Table. III shows a global view of all the mean errors between the reference mocap input trajectories and the robot output trajectories. It highlights that the best-performing scenarios are those with the "a_2" human orientation. This 90° angle corresponds to a wrist trajectory in a plane of constant depth relative to the camera, confirming the impact of depth variation on task performance as discussed in Section III-A.

IV. CONCLUSIONS

In this paper, we introduce a mixed-tests evaluation methodology for intelligent collaborative robotic systems performing handover tasks. This approach aims to identify the most important scenario parameters (human orientation for the handover task), critical corner-case scenarios and minimize the number of real-world tests required to validate the system's performance. To adapt this approach to a different task, such as service robotics in healthcare, we need to define the potential scenarios, create a database of unpredictable data from AI systems or human interactions, and use this data and the models of the other to simulate the entire collaborative task. To the best of our knowledge, no existing methods or studies have proposed a similar approach. However, some papers such as [20] test the accuracy of 3D cameras using a motion capture system with a setup similar to ours (Fig. 3).

Future work includes real robot experiments using the Panda 7DOF robot to validate our approach.

REFERENCES

[1] M. D. Silva, R. Regnier, M. Makarov, G. Avrin, and D. Dumur, "Evaluation of intelligent collaborative robots: a review," in *2023 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, pp. 1–7.

[2] M. Metzner, D. Utsch, M. Walter, C. Hofstetter, C. Ramer, A. Blank, and J. Franke, "A system for human-in-the-loop simulation of industrial collaborative robot applications*," in *2020 IEEE 16th Int. Conf. on Automat. Sci. and Eng. (CASE)*, Aug. 2020, pp. 1520–1525.

[3] A. H. Quispe, H. B. Amor, and H. I. Christensen, "A taxonomy of benchmark tasks for robot manipulation," in *Robotics Research*. Springer, 2018, pp. 405–421.

[4] M. J. Rosenstrauch and J. Krüger, "Safe human-robot-collaboration-introduction and experiment using iso/ts 15066," in *2017 3rd International conference on control, automation and robotics (ICCAR)*. IEEE, 2017, pp. 740–744.

[5] V. Ortenzi, A. Cosgun, T. Pardi, W. P. Chan, E. Croft, and D. Kulić, "Object handovers: A review for robotics," *IEEE Transactions on Robotics*, pp. 1–19.

[6] L. Villani, A. De Santis, V. Lippiello, and B. Siciliano, "Human-aware interaction control of robot manipulators based on force and vision," in *Robot Motion and Control 2009*. Springer, 2009, pp. 209–225.

[7] W. Yang, C. Paxton, A. Mousavian, Y.-W. Chao, M. Cakmak, and D. Fox, "Reactive Human-to-Robot Handovers of Arbitrary Objects," June 2021.

[8] A. Sidiropoulos, E. Psomopoulou, and Z. Doulgeri, "A human inspired handover policy using gaussian mixture models and haptic cues," *Autonomous Robots*, vol. 43, no. 6, pp. 1327–1342.

[9] M. Rietzler, F. Geiselhart, J. Thomas, and E. Rukzio, "FusionKit: a generic toolkit for skeleton, marker and rigid-body tracking," in *Proceedings of the 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. ACM, pp. 73–84.

[10] T. Aarsh, "Seamless integration of Optitrack motion capture with ROS," June 2022. [Online]. Available: https://hal.science/hal-04150950

[11] P. Corke and J. Haviland, "Not your grandmother's toolbox–the robotics toolbox reinvented for python," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 357–11 363.

[12] C. Gaz, M. Cognetti, A. Oliva, P. Robuffo Giordano, and A. De Luca, "Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4147–4154.

[13] N. Hogan, "Impedance control: An approach to manipulation: Part II—implementation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 8–16.

[14] F. Ficuciello, L. Villani, and B. Siciliano, "Redundancy resolution in human-robot co-manipulation with cartesian impedance control," in *Experimental robotics*. Springer, pp. 165–176.

[15] C. Ott, *Cartesian impedance control of redundant and flexible-joint robots*. Springer, 2008, ch. 3.

[16] C. Saldarriaga, N. Chakraborty, and I. Kao, "Damping selection for cartesian impedance control with dynamic response modulation," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1915–1924.

[17] T. Coleman, G. Franzese, and P. Borja, "Damping Design for Robot Manipulators," in *Human-Friendly Robotics 2022*, P. Borja, C. Della Santina, L. Peternel, and E. Torta, Eds. Cham: Springer International Publishing, 2023, vol. 26, pp. 74–89, series Title: Springer Proceedings in Advanced Robotics.

[18] F. Schlagenhauf, S. Sreeram, and W. Singhose, "Comparison of kinect and vicon motion capture of upper-body joint angle tracking," in *2018 IEEE 14th international conference on control and automation (ICCA)*. IEEE, 2018, pp. 674–679.

[19] S. Giancola, A. Corti, F. Molteni, and R. Sala, "Motion capture: an evaluation of kinect v2 body tracking for upper limb motion analysis," in *Wireless Mobile Communication and Healthcare: 6th International Conference, MobiHealth 2016, Milan, Italy, November 14-16, 2016, Proceedings 6*. Springer, 2017, pp. 302–309.

[20] M. Zago, M. Luzzago, T. Marangoni, M. De Cecco, M. Tarabini, and M. Galli, "3D Tracking of Human Motion Using Visual Skeletonization and Stereoscopic Vision," *Front. Bioeng. Biotechnol.*, vol. 8, p. 181, Mar. 2020.