The International Journal of Advanced Manufacturing Technology (2024) 134:57–73
https://doi.org/10.1007/s00170-024-14104-7

ORIGINAL ARTICLE

Experimental implementation of skeleton tracking for collision avoidance in collaborative robotics

Matteo Forlini[1] · Federico Neri[1] · Marianna Ciccarelli[1] · Giacomo Palmieri[1] · Massimo Callegari[1]

Received: 6 March 2024 / Accepted: 3 July 2024 / Published online: 15 July 2024

Abstract
Collaborative robotic manipulators can stop in case of a collision, according to the ISO/TS 15066 and ISO 10218-1 standards. However, in a human-robot collaboration scenario where the robot and the human share the workspace, a better solution with concerns to production and operator safety would be to perceive in advance the presence of an obstacle and be able to avoid it, thereby completing the task without halting the robot. In this paper, an obstacle avoidance algorithm is tested using a sensor system for real-time human detection; the operator represents a potential dynamic obstacle that can interfere with the robot motion. The sensor system consists of three RGB-D cameras. A custom software framework has been developed in Python exploiting machine learning tools for human skeleton detection. The coordinates of the human body joints relative to the manipulator base are used as input to the obstacle avoidance algorithm. The use of multiple sensors makes it possible to limit the occlusion problem; in addition, the choice of non-wearable sensors goes in the direction of better operator comfort. A series of experimental tests were performed to verify the accuracy of skeleton detection and the ability of the system to avoid obstacles in real time. The human motion caption system, in particular, was validated through a comparison with a commercial system based on wearable IMU sensors widely used and validated in motion capture. There is an NRMSE of 3.23% for the RGB-D camera-based skeleton detection system, against an NRMSE of 12.23% for the IMU wearable sensor system. Test results confirm that the system is able to avoid collisions with the human body under various conditions, static or dynamic, ensuring a minimum safety distance to any part of the manipulator. In 44 tests in which the operator moves around the robot with possible collisions (at a speed typical of manufacturing operations), the minimum operator-robot distance averaged 208 mm, being 200 mm the limit safety distance set by algorithm.

Keywords Collaborative robotics · Human robot interaction · Obstacle avoidance · Computer vision · Skeleton tracking

1 Introduction

Collaborative robotics is a new paradigm of robotics falling under the nine points of Industry 4.0 [22] that allows humans to work symbiotically with the robot [26]. Contrary to classical industrial robotics, human and robots can interact by sharing the work space or working simultaneously to the same task [2, 6]. Collaborative robots are typically characterized by a rounded shape, low payloads and low speeds (with respect to industrial robots). These features are necessary to intrinsically ensure operator's safety [3]. In addition, sensor systems can be used to sense any external contact, preventing violent impacts with humans. The result of the collaboration is the improving of the physical and mental well-being of the operator [21]: humans are given the tasks that require more flexibility and ingenuity, while repetitive, tedious and tiring tasks are assigned to robots.

1.1 Motivation

Collaborative robots can halt if a collision is detected, according to ISO/TS 15066 and ISO 10218-1 standards [1, 2]. In terms of production and operator safety, a preferable solution would be to perceive the presence of an obstacle in advance and be capable of avoiding it, thereby completing the task, without halting the robot and thus production. The human body itself can be considered an obstacle during the collaboration between robot and operator, so it is important to

✉ Matteo Forlini
m.forlini@pm.univpm.it

[1] DIISM—Department of Industrial Engineering and Mathematical Sciences, Polytechnic University of Marche, Via Brecce Bianche, Ancona 60131, Italy

discover the presence of a human and implement an obstacle avoidance technique to update the motion of the robot. With this technique, the interaction with the robot will be more efficient and usable. [24] claimed that, to completely ensure no contact with the robot, the distance between each part of the human body and each part of the robot structure must be taken into account. Although there are numerous technical insights in the literature on this field, the need felt by the authors is to create a comprehensive framework that is simple, robust, based on open source libraries, and capable of handling multiple sensors. The purpose of this framework is to identify the presence of humans in the robot's workspace, allowing real-time replanning of the trajectory in order to prevent collisions. The architecture of the system along with all the codes is provided at https://github.com/matteoforlini/human_obstacle_avoidance.git.

1.2 Related works

Two categories of exteroceptive sensors can be used to get in real time the position of the human body: wearable and non-wearable sensors [10]. Among wearable sensors, the most widely used are inertial measurement units (IMU) or stereophotogrammetric motion capture system (SP). In recent years, efficient and accurate algorithms have been developed that compute human posture data by means of low-cost miniaturized inertial and magnetic sensors [19]; their efficiency and wide usage are reported in [16]. In [12], it is presented the integration of a LRF (laser range finder) sensor that tracks the human legs with a wearable IMU system used to capture the human motion during gait, thereby improving natural mobile robot-human following. Furthermore, a safe human-robot interaction is implemented in [34] using IMU sensors. Although IMU sensors are accurate and have a high frequency rate, they must be worn by operators creating comfort problems and are subject to drift problems that require frequent calibration.

Among unwearable sensors, depth camera sensors (RGB-D) are the most popular, such as Microsot Kinect, Intel RealSense, and Asus Xtion. In this case, the use of multiple cameras is essential to prevent occlusion problems. Usually, several cameras have to be set in the space in order to cover the principle viewpoints; then, a data fusion algorithm is needed [25, 33].

Kinect cameras are used in [30, 31] to detect the human hand and then implement an obstacle avoidance strategy based on kinetostatic receptive field. An algorithm based on controller barrier function around each link of the manipulator to avoid contact with the operator is presented in [15]. Two RGB-D cameras are used to know the position of the human, while Kalman filters are used to extract velocities and accelerations.

In [17, 18], a method to rapidly merge data from multiple RGB-D cameras and calculate the distance between two points of interest (robot and human link) is proposed, whereas [35] present a collision avoidance algorithm based on the distance between human joints (identified via two kinect sensors) and the robot links. Similarly, [27] present a strategy where human positions, speeds, and accelerations are known through the use of 4 kinect cameras. The robot's trajectory in the near future is simulated, so the robot stops its movement whenever an upcoming collision with humans is sensed.

Nowadays, machine learning (ML) techniques are spreading in work environments where human-robot interaction is taken into account [28, 39]. A control system based on multiple sensors for safe human-robot collaboration based on RGB-D cameras and mixed reality interface is proposed in [23]. In [14, 41], a strategy for obstacle avoidance based on operator identification using RGB-D camera is presented; the skeleton detection software is based on a deep learning tool. However, the problem of occlusions in a working environment is not addressed. An innovative collision avoidance approach working with one single camera is presented in [29].

Among the various path and trajectory planning algorithms [20], in the field of collision avoidance, the potential fields method is commonly used [38]. Scoccia et al. [36] developed algorithms based on potential fields and real-time velocity control where a repulsive velocity inversely proportional to the distance from the obstacle is assigned to the nearest control point of the kinematic chain of the manipulator. In [11], a similar obstacle avoidance strategy is implemented on a UR5e robot. The manipulator is enveloped along its kinematic chain by capsule-shaped safety regions of varying radius depending on the speed of the obstacle. Three different modes of obstacle avoidance are presented depending on the desired end-effector perturbation in reaction to a possible collision.

1.3 Aim of the research

The purpose of this paper is to present the integration and implementation, on a real human-robot interaction scenario, of an obstacle avoidance strategy with a vision system for real-time obstacle detection developed by the authors and validated with a different measuring system based on inertial sensors. The authors want to provide and validate a Python software framework based on machine learning tools for human skeleton detection (Mediapipe) and integrate it with an obstacle avoidance strategy used to control a collaborative manipulator. Mediapipe is widely used in the field of human motion detection, and its reliability is solidly proven [9, 37]. The framework allows an easy and robust implementation and integration of RGB-D cameras of any brand and han-

dles the occlusion problem by using several cameras to cover the entire workspace by performing an efficient data fusion. The setup used for tests consists of three Intel Realsense D455 RGB-D cameras and a UR5e cobot; the choice of non-wearable sensors goes in the direction of a better comfort for the operator. The coordinates of the human body joints relative to the robot base are then used as input to the obstacle avoidance algorithm.

A series of experimental tests were performed to verify the accuracy of the skeleton detection and the ability of the system to avoid obstacles in real time. In particular, several tests were performed with the aim of replicating the interaction and cooperation between the cobot and the operator, using also different types of robot mobility in order to study the manipulator's response to different working scenarios. The human motion caption system was validated through a comparison with the Xsens MVN suite [5]. This commercial system provides the position and orientation of body landmarks based on wearable IMU sensors data and represents a widely used and validated system in the motion capture field [40].

The remainder of the manuscript is organized as follows: Sect. 2 presents the skeleton detection algorithm and its implementation; Sect. 3 briefly describes the obstacle avoidance algorithm; experimental tests are described in Sect. 4, while results are shown in Sect. 5; a final discussion and some insights to future developments are given in Sect. 6.

2 Skeleton detection algorithm

The purpose of this section is to describe the hardware configuration and the architecture and implementation of the skeleton tracking software, the output of which is the coordinates of human landmarks relative to the robot base.

2.1 Hardware and software setup

Human motion tracking is performed through three Intel Realsense D455 cameras. As shown in Fig. 1, two of them are placed at a distance $d \simeq 1.2$ m from the operator's workplace, symmetrically disposed with a 120° angle between them, at a height of about 0.8 m from the robot base. The third is placed in front of the operator, at a height of about $d \simeq 1.2$ m from the robot base, and a distance $d_1 \simeq 1.6$ m from the operator. This makes it possible to see the scene from different viewpoints, limiting the occlusion problem: at some stages of the robot's motion, the viewpoint of one camera may not detect the human body because it is covered by the robot's arm, in which case information is provided by the other cameras. The rear view of the operator is not useful for this type of application, so it may be disregarded. The robot,

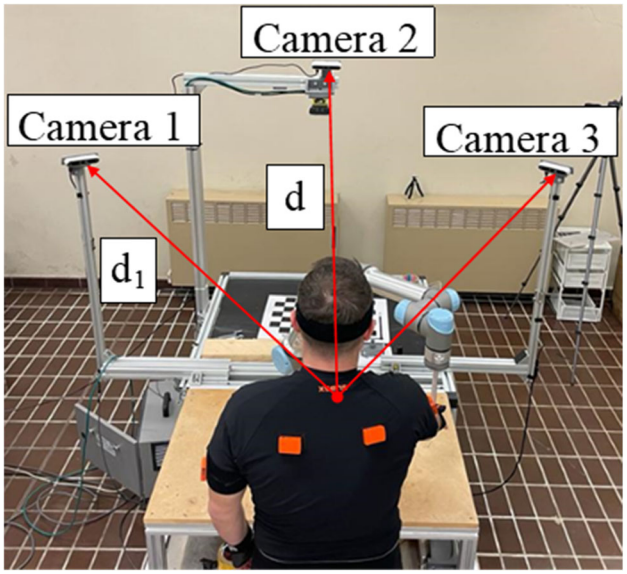

Fig. 1 Hardware setup: cobot installation and positions of the three RGB-D cameras with respect to the operator

a Universal Robot 5e cobot, is mounted on the workbench, in front of the operator.

Figure 2 shows the development environment used for the implementation of the skeleton tracking software. The cameras are connected to a standard PC running skeleton tracking software developed in Python. The cameras are configured using the Intel Realsense library for Python. RGB images and depth images are set to a resolution of 424×240 pixel, which is a compromise between accuracy and computational effort of image processing. The alignment of the two frames is done by interpolation, so that a pixel at coordinates (u_i, v_i) in the depth plane matches the pixel at the same coordinates in the RGB plane. Cameras are managed by three parallel threads, each of which outputs x, y, z coordinates of human body joints: images acquired by each RGB sensor are managed with tools from OpenCv computer vision library; then the skeleton identification is realized by means of the Mediapipe library tools, developed by Google, based on artificial intelligence algorithms. Two Matlab functions are called within the Python software. The first function is used to transform the coordinates of body joints from the camera reference frame to the robot reference frame; this is possible once the intrinsic and extrinsic parameters of the cameras have been estimated through calibration (Sect. 2.3). The second function is intended for data fusion between the three cameras: a specific index of visibility, provided by Mediapipe, is used to score the cameras based on the quality of their acquisition. For each body joint, the coordinates coming from the camera with the highest score are considered as output. In this way, the erroneous values provided by the low-scoring data do not affect the accuracy of the coordinate estima-

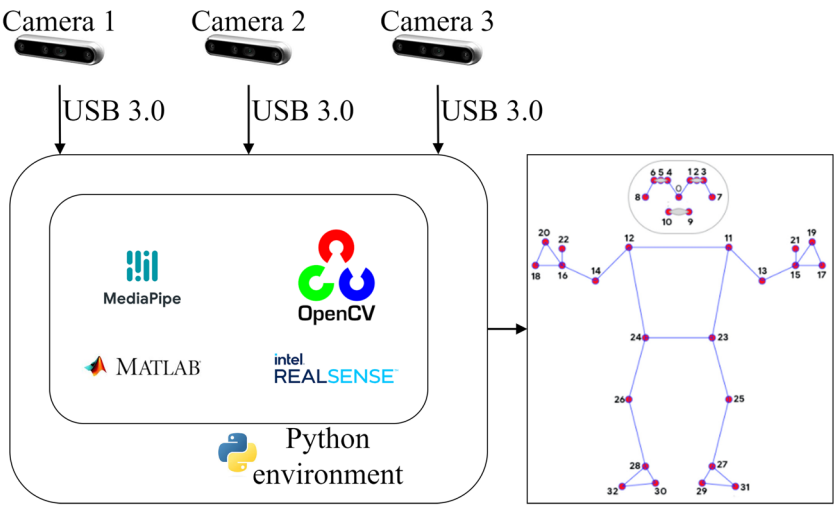

Fig. 2 Development environment used for the implementation of the skeleton tracking software

tion; alternatively, the average of the data provided by all sensors with sufficient confidence can be considered; the authors did not observe any substantial differences between the two approaches. These functions could be implemented using Python; however, since calling Matlab functions from Python scripts does not significantly slow down the process, the authors decided to keep this approach.

2.2 AI tools for skeleton detection

Mediapipe is an artificial intelligence-based framework which is able to obtain information from temporal data such as video and audio. MediaPipe Pose is a ML solution for body pose detection. It detects 33 3D landmarks and performs full body background segmentation mask from RGB video frames [4]. A deep learning detector named BlazePose [8] is first used to identify the region of interest (ROI) within which the person is positioned, then to identify within this region the location of landmarks. In the case of video, the ROI is identified only when necessary, that is, in the first frame and when the tracking algorithm can no longer identify the presence of the body pose in the previous frame. For the other frames, the ROI is derived from the landmarks of the previous frame.

For each landmark, identified by the index ID_i, the u_i, v_i pixel coordinates normalized to with respect to the image width and height respectively are provided. The depth value corresponding to the u_i, v_i point on the depth sensor is considered to evaluate the z_i coordinate (expressed in [m]) of ID_i. Among the various adjustable parameters of the Mediapipe software, two proved to be particularly influential for correct skeletal detection: they are the minimum detection confidence and minimum tracking confidence, both settable from 0 to 1. In the present application, a value of 0.9 was set for both parameters to improve the robustness of the algorithm. In addition, each landmark in the scene can be associated with a visibility index ranging from 0 to 1: visibility indicates the probability that a landmark is not occluded in the scene. In the case of occlusion, the Mediapipe software can still estimate the location of the occluded landmark by obtaining information from other detected parts of the skeleton model, but the assigned visibility index will be low. If, on the other hand, the landmark is clearly visible, the index will tend to 1. Therefore, to give robustness to the system, only landmarks with a visibility index greater than 0.95 are taken into account in the rest of the calculation. Furthermore, for a given landmark, the camera that detects it with the greater visibility index is selected to retrieve the coordinates to be input to the obstacle avoidance algorithm.

2.3 Camera calibration

The intrinsic and extrinsic parameters of the RGB sensor were obtained with the camera calibration procedure described below. First, a standard calibration was performed with the dedicated Matlab toolbox using images of a randomly placed checkerboard within the field of view; one of the images was obtained by placing the checkerboard at a known position relative to the robot base in order to find an initial guess of the relative robot-camera pose.

Once the intrinsic parameters and the estimated relative pose between robot and camera are known from the preliminary calibration, parameter refinement was performed through further hand-eye calibration: a custom tool with a spherical tip was mounted on the end-effector of the robot and moved to 12 different positions framed by the camera; the 3D coordinates of the spherical tip are known from the kinematic model of the manipulator, with a measurement uncertainty corresponding to the nominal accuracy of the robot. The 3D metric coordinates of the tip are then projected into the RGB plane using the estimated intrinsic and extrinsic parameters, resulting in pixel coordinates that can

be compared with the point corresponding to the centroid of the tip on the image. The error between the projected and the framed centroid, cumulated over all the different 12 tip positions, is considered as an objective function of a minimization procedure (fminsearch algorithm, Matlab) that optimizes the extrinsic parameters of the RGB sensor. Then, representing the optimized relative camera-robot pose by the transformation matrix ${}^{r}_{cam}\mathbf{T}$ and given the coordinates ${}^{cam}\mathbf{P}$ of a point with respect to the camera reference system (Fig. 3), the coordinates of the same point expressed with respect to the robot's base frame can be obtained as

$$ {}^{r}\mathbf{P} = {}^{r}_{cam}\mathbf{T}\,{}^{cam}\mathbf{P} \tag{1} $$

The quality of calibration can be expressed in terms of reprojection error, that is, in our case lower than 0.8 pixel for all the cameras. In terms of real world X, Y, Z coordinates, given the pixels coordinates u, v of a point and reading its distance form the camera by the depth sensor (890 mm on average), the 3D position of that point can be estimated with an error lower than 3 mm. The error is calculated as the Euclidean distance between the position of the point measured by the robot controller and the position estimated by the optical system. The order of magnitude of the error can be considered acceptable for two reasons: firstly, the anatomical dimensions of landmarks, such as the elbow or wrist, are much larger than the measurement error of their position, so the inaccuracy is mostly related to their schematizing as a point, neglecting their 3D shape; secondly, the area of influence of the obstacle around the manipulator links is about 200 mm, again much larger than the error in estimating the position of the landmark.

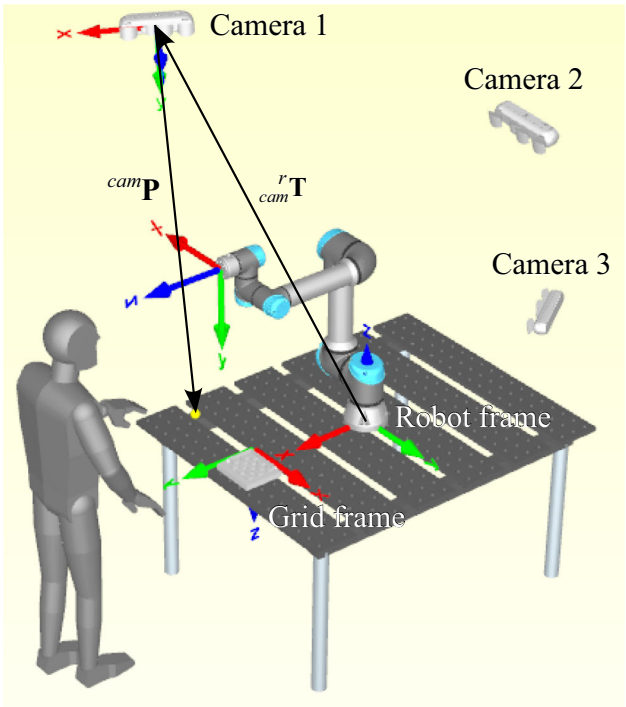

Fig. 3 Coordinates transformation from camera reference system to robot base reference system

Once intrinsic and extrinsic parameters of each camera are known from calibration, 3D coordinates of each landmark can be estimated with the following steps: given by the depth sensor the z_i coordinate of a landmark, and given its u_i, v_i pixel coordinates on the camera sensor, the classic pinhole problem can be solved for x_i, y_i metric coordinates. Then, Eq. 1 allows to change coordinates of the landmark from the camera frame to the robot base frame.

3 Obstacle avoidance strategy

In this section, the obstacle avoidance strategy is illustrated in brief. The algorithm can be adopted for a generic manipulator with a number of actuators n greater or equal to 6. The vector of joints position is defined as $\mathbf{q} = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix}^T$, while the pose of the end-effector in the Cartesian space is defined by the vector $\mathbf{x} = \begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^T$, where the three angles of rotation follow the Euler angle ZYZ convention. The velocity forward kinematics can be written as

$$ \dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_p \\ \mathbf{J}_r \end{bmatrix} \dot{\mathbf{q}} \tag{2} $$

where $\mathbf{J}$ is the $6 \times n$ arm Jacobian ($n \geq 6$), $\mathbf{J}_p$ is the $3 \times n$ part relative to translations, and $\mathbf{J}_r$ the $3 \times n$ part relative to rotations.

In order to detect the risk of collision between an obstacle and the manipulator's chain, at each time step of the algorithm, the distance of each obstacle from each link of the robot is calculated. Being $\mathbf{P}_o$ the position of the obstacle, its distance $\mathbf{d}_o$ from a link is calculated according to the diagrams of Fig. 4, in which $\mathbf{P}_r$ is the point of the link closest to the obstacle. The Jacobian $\mathbf{J}_0$ related to the velocity of $\mathbf{P}_r$ is expressed as

$$ \mathbf{J}_0 = \begin{bmatrix} \mathbf{J}_{0p} \\ \mathbf{J}_{0r} \end{bmatrix} \tag{3} $$

where $\mathbf{J}_{0p}$ and $\mathbf{J}_{0r}$ are the translation and rotation parts, respectively. The anti-collision control is activated when the distance between the obstacle and the robot is lower than a settable threshold r. Thus, the safety volume of each link can be viewed as a cylinder with two hemispheres at its extremities, all of radius r, and to avoid the collision, a repulsive velocity $\dot{\mathbf{x}}_0$ is applied to $\mathbf{P}_r$ in the opposite direction of $\mathbf{d}_o$ if $d_o < r$.

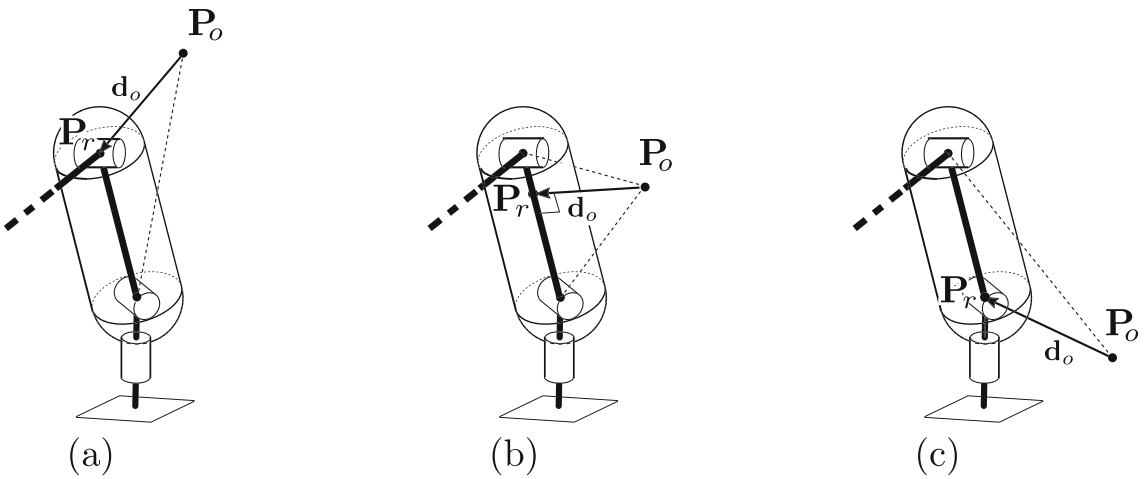

Fig. 4 Different cases for obstacle-link distance calculation depending on the position of the obstacle with respect to the link: **a** the point $\mathbf{P}_r$ of the link closest to the obstacle $\mathbf{P}_o$ corresponds to distal extremity of the link; **b** $\mathbf{P}_r$ is found as the orthogonal projection of $\mathbf{P}_o$ over the link; **c** $\mathbf{P}_r$ corresponds to the proximal extremity of the link

As described in Chiriatti et al., three different modes are implemented to execute the repulsive velocity $\dot{\mathbf{x}}_0$:

3.1 Mode I: 6-DOF perturbation

Mode I gives the manipulator the possibility to avoid the obstacle changing both orientation and position. The control law assumes the form:

$$\dot{\mathbf{q}} = \mathbf{J}^* (\dot{\mathbf{x}} + \mathbf{K}\mathbf{e}) + \mathbf{J}_I^* \dot{\mathbf{x}}_0 \tag{4}$$

where $\dot{\mathbf{x}}$ is the vector of planned velocities, $\dot{\mathbf{x}}_0$ is the repulsive velocity, $\mathbf{J}^*$ is the damped least square (DLS) inverse of the Jacobian [13], $\mathbf{J}_I = \mathbf{J}_{0p}$ and $\mathbf{K}$ is a diagonal gain matrix acting on the Cartesian position error vector $\mathbf{e}$.

3.2 Mode II: 4-DOF perturbation

In case only a 4-DOF Schoenflies-like perturbation is admitted (3-DOF translation and a rotation about the vertical axis), the Eq. 4 is modified in

$$\dot{\mathbf{q}} = \mathbf{J}^* (\dot{\mathbf{x}} + \mathbf{K}\mathbf{e}) + \mathbf{J}_{II}^* \begin{bmatrix} \dot{\mathbf{x}}_0 \\ \mathbf{0}_{2\times 1} \end{bmatrix} \tag{5}$$

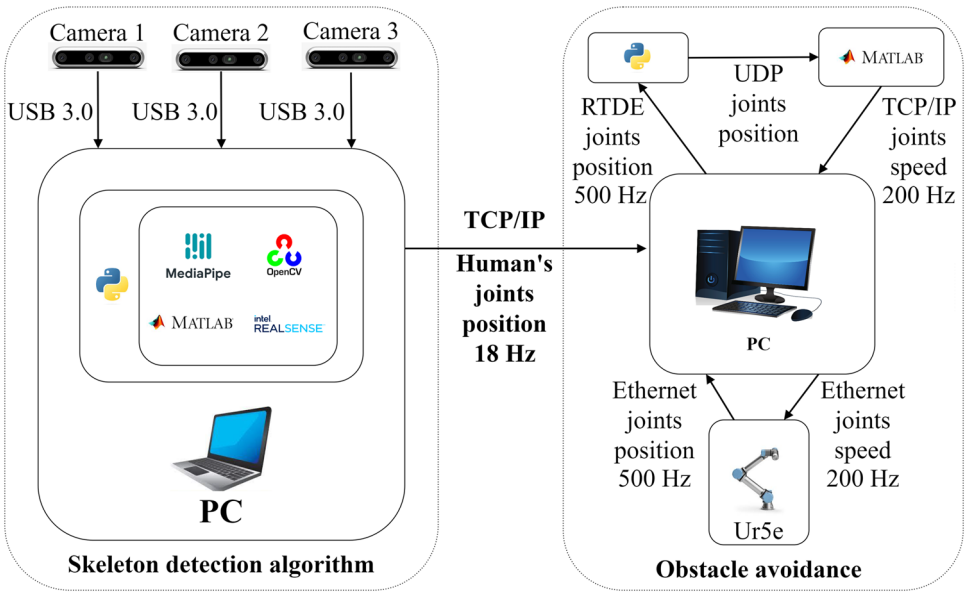

Fig. 5 Overall communication architecture between the two sub-systems: skeleton detection system and obstacle avoidance controller

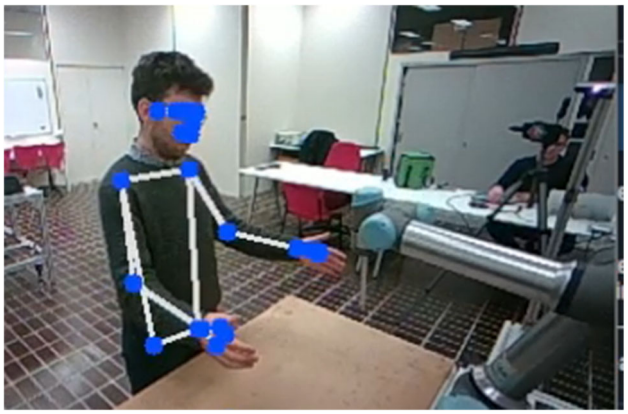

Fig. 6 Graphic output of the skeleton detection system

Fig. 7 Pose of the operator during the stationary test for skeleton detection systems

The Jacobian matrix J_{II}, of dimension $(5 \times n)$, is defined as

$$\mathbf{J}_{II} = \begin{bmatrix} \mathbf{J}_{0p} \\ \mathbf{J}_{r4} \\ \mathbf{J}_{r5} \end{bmatrix} \tag{6}$$

being $\mathbf{J}_{r4}$ and $\mathbf{J}_{r5}$ the first and second rows of the orientation Jacobian matrix $\mathbf{J}_r$ of the end-effector.

3.3 Mode III: 3-DOF perturbation

In this case, only a perturbation with constant orientation of the end-effector is allowed. The control law becomes

$$\dot{\mathbf{q}} = \mathbf{J}^* \left(\dot{\mathbf{x}} + \mathbf{K}\mathbf{e}\right) + \mathbf{J}^*_{III} \begin{bmatrix} \dot{\mathbf{x}}_0 \\ \mathbf{0}_{3\times 1} \end{bmatrix} \tag{7}$$

The Jacobian matrix J_{III}, of dimension $(6 \times n)$, is made up of the translation part of $\mathbf{J}_0$ and the orientation part of $\mathbf{J}$:

$$\mathbf{J}_{III} = \begin{bmatrix} \mathbf{J}_{0p} \\ \mathbf{J}_r \end{bmatrix} \tag{8}$$

Table 1 Main parameters of obstacle avoidance algorithm: r_0 is the distance of activation of the avoidance control, r is the safety radius from the manipulator, $|\dot{x}_0|$ is the magnitude of the repulsive velocity, k_{ep} and k_{er} are the gain values for the error compensation term for translation and orientation respectively, $\dot{\theta}_{max}$ is the joint speed of saturation, and T is the total time of the trajectory

r_0 [mm]	r [mm]	$\lvert\dot{\mathbf{x}}_\mathbf{0}\rvert$ [m/s]	k_{ep}	k_{er}	$\dot{\theta}_{max}$ [rad/s]	T [s]
260	200	0.40	1.1	10	$\pi/2$	6

4 Experimental setup

The experimental tests have been designed to recreate a scenario of interaction between a human and a collaborative robot. The operator stands in front of the robot as shown in Fig. 3. An end-effector with a length of 205 mm is mounted on the robot's flange to simulate the encumbrance of a generic tool. Human joints coordinates obtained by the skeleton detection system are sent to the obstacle avoidance algorithm at a frequency of 18 Hz. The reference signal generated by the external controller is sent to the robot controller by TCP communication. The overall communication architecture is shown in Fig. 5. The collision avoidance algorithm runs in Matlab at a frequency of 200 Hz. A strategy is needed to match the different execution frequencies between the skele-

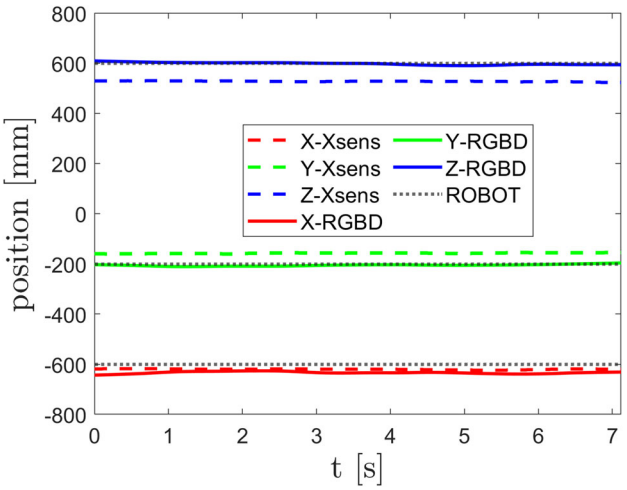

Fig. 8 Data of the skeleton detection systems comparison - stationary test

Table 2 Statistical parameters for the RGB-D cameras and Xsens measuring systems in the stationary test

Ground truth	[mm]	*X* -600	*Y* -200	*Z* 600	*X* -600	*Y* -200	*Z* 600
		RGB-D cameras			Xsens		
Mean	[mm]	−633.76	−205.21	598.05	−620.31	−157.71	527.25
Std	[mm]	4.14	3.44	4.93	1.68	1.57	1.56
RMSE	[mm]	34.01	6.24	5.30	20.38	42.31	72.75
NRMSE		5.67%	3.12%	0.09%	3.40%	21.16%	12.13%

ton detection algorithm and the collision avoidance motion control: the newest output data-set form the skeleton detection software is stored in memory and given as input to the motion controller until a new data-set is available. Moreover, it should be mentioned that although the frequency of the skeletal sensing system (about 18 Hz) is considerably lower than that of the motion controller, it is still sufficient to accurately sense the voluntary movement of a human being, which is typically less than about 10 Hz (98% of the FFT amplitude is contained below 10 Hz [7, 32]).

Figure 6 shows a typical graphic output of the skeleton detection system. In terms of data, a matrix of values with five rows (Landmark ID, Visibility index and Cartesian coordinates) and 33 columns (number of anatomical landmarks) is sent to the obstacle avoidance planner.

Landmarks belonging to the lower part of the human body were not considered in the tested application because the robot's workspace makes collisions possible only with the upper part of the body.

Joint limits of the robot and mounting table limitations can be commonly managed by specific tools of the control

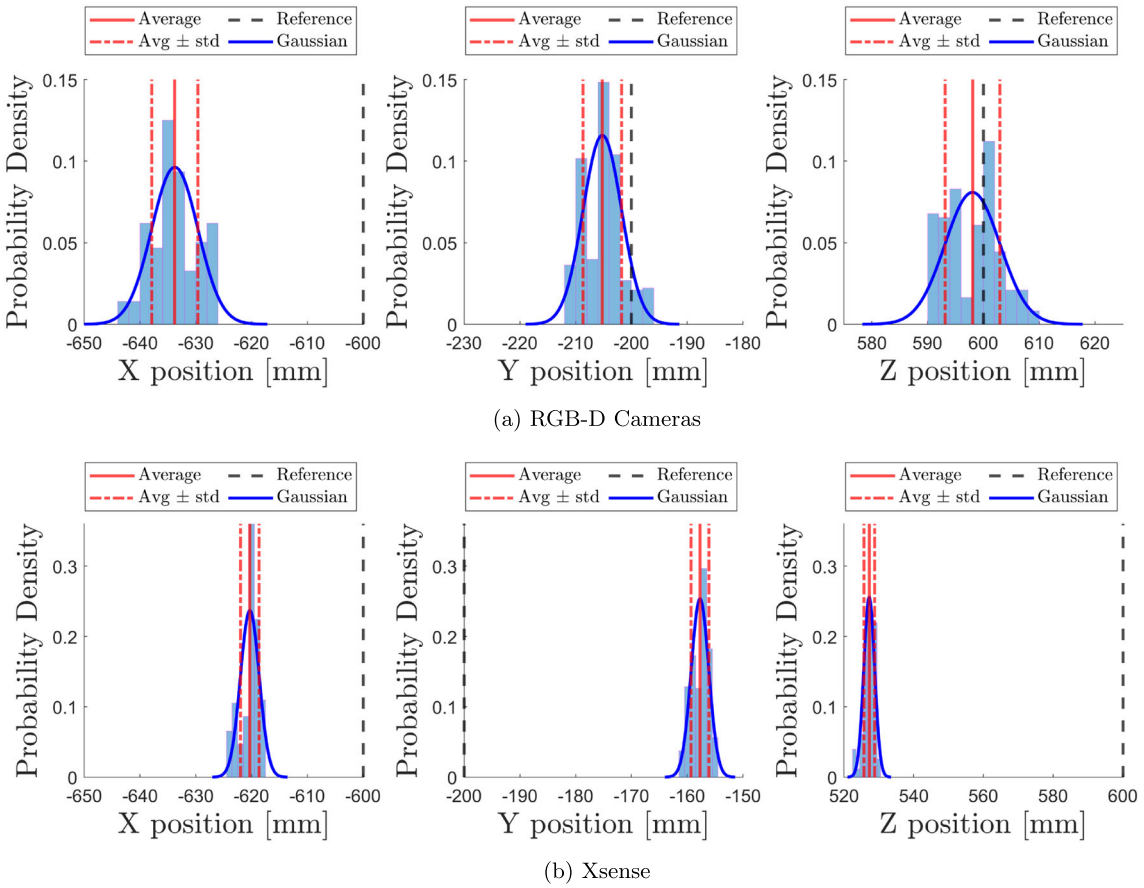

Fig. 9 Probability density of position measurements in the stationary test for **a** RGB-D cameras and **b** Xsens

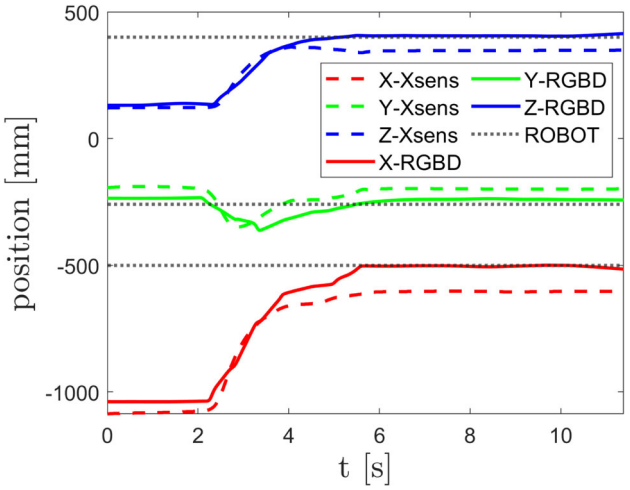

Fig. 10 Data of the skeleton detection systems comparison - dynamic test

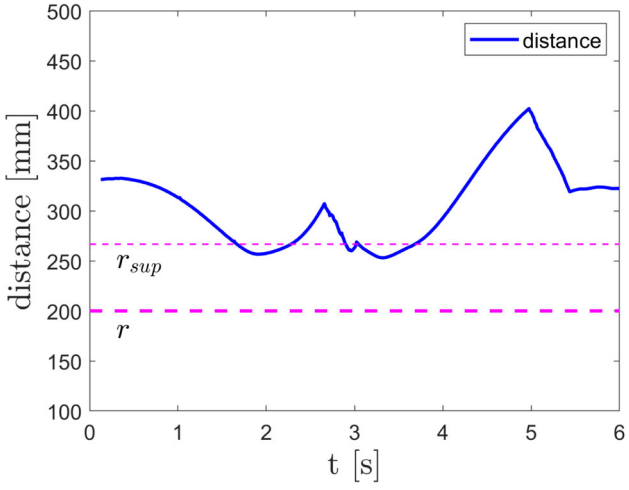

Fig. 11 Test Case 1 - minimum human-robot distance - Mode I

software of the robot. As an alternative strategy, hard planes or walls or other fixed obstacles can be simulated as a fix set of points considered as obstacles, thus normally processed by the collision avoidance algorithm. For the obstacle avoidance algorithm, basic parameters were set at values reported in Table 1. The radius of the safety volume around the links was set to $r = 200$ mm by trial and error related to the time of arrest of the robot and the speed of the operator. The total duration of the test is $T = 6$ s. The maximum joint speed assignable to the robot is represented by $\dot{\theta}_{max}$, whereas k_{ep} and k_{er} are gain values related to translation and orientation error which compose the diagonal gain matrix $\mathbf{K}$.

To validate the skeleton detection algorithm discussed in Sect. 2, a comparison was made with the joint positions provided by another human pose detection system based on operator-worn inertial sensors, i.e. the Xsens MVN suite motion capture system. The Xsens system provides the position of the human joints relative to the right heel coordinate frame with the positive direction of the X-axis pointing in front of the operator, the Z-axis pointing up, and the Y-axis pointing left. A common coordinate reference system can be obtained by a calibration procedure: the operator is asked to rest the right heel on a stand (whose position and orientation with respect to the robot is known) during the calibration of the IMU sensors; thus, the position of the right heel in space with respect to the robot frame is known, and then, by applying a transformation matrix, the position measurements provided by the IMUs with respect to the robot frame are obtained.

Table 3 Concordance correlation coefficient (CCC) analysis between RGB-D cameras and Xsens systems in the dynamic test

	X	Y	Z
CCC	0.940	0.476	0.936

The comparison between RGB-D and IMU-based systems was done by two experimental tests. In the first test (stationary test), the operator is stationary in front of the robot with right wrist under the robot's end-effector (Fig. 7). With this test, the stability of the values provided by the two systems and their accuracy are evaluated, since the actual position of the wrist relative to the robot's base is provided by the robot's kinematic chain.

The plot of Fig. 8 shows the X, Y, Z coordinates measured by the RGB-D cameras and by the Xsens sensors, whereas dotted curves refer to values obtained from the kinematic chain of the robot. Some measurement error is expected because the wrist is modelled as a point in both skeletal detection systems, thus simplifying its true 3D shape.

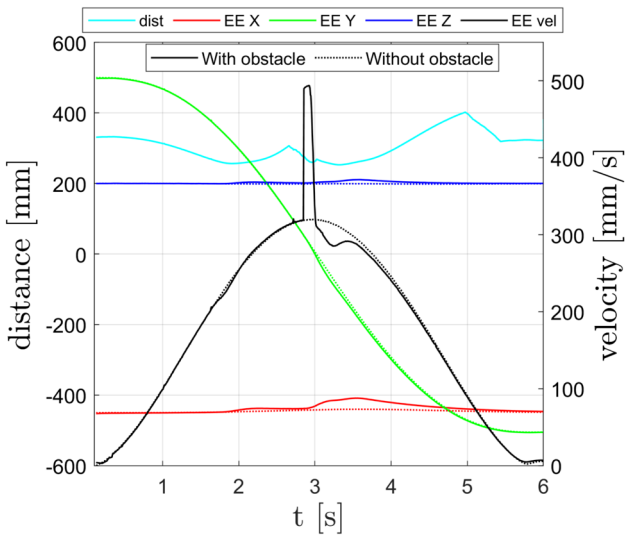

Fig. 12 Test Case 1 - end-effector position (X, Y, Z) and velocity in relation with the minimum human-robot distance - Mode I

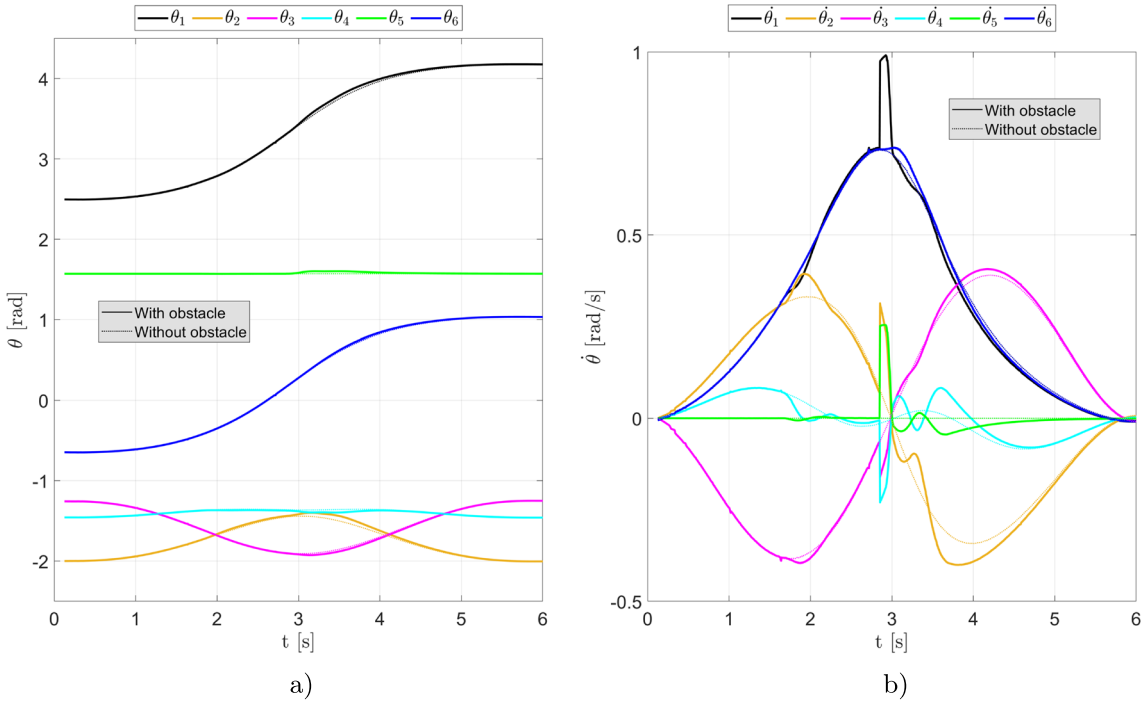

Fig. 13 Test Case 1 - joints angular position (**a**) and velocity (**b**) - Mode I

In Table 2, the main statistical parameters of the measurements of the right wrist position in the stationary test, obtained using the RGB-D and the Xsens systems, respectively, are reported. Root mean square error (RMSE) and normalized root mean square error (NMRSE), as defined in Eqs. 9 and 10, are calculated considering the robot end-effector pose read from the controller as ground truth.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - y_i)^2} \tag{9}$$

$$\text{NRMSE} = \frac{\text{RMSE}}{\hat{y}} \tag{10}$$

where N is the number of samples and $\hat{y}$ is the mean value.

From Table 2 and Fig. 9, it can be seen that the skeleton tracking system based on RGB-D cameras has a lower RMSE and NRMSE than Xsens: the average values are closer to the ground truth provided by the robot's position. However, the Xsens system is characterized by a lower dispersion of measurements, as indicated by its lower standard deviation values: this means that even though this system can be considered more precise, data drift and the difficulty in referring measurements to a localized reference system with respect to the robot compromise the effective accuracy of the system.

In the second test (dynamic test), the operator moves the right wrist from the workbench to the robot's end-effector; this assesses the response of the two systems to a human body motion. Results are shown in Fig. 10. RGB-D and Xsens systems show a similar dynamic behaviour with comparable

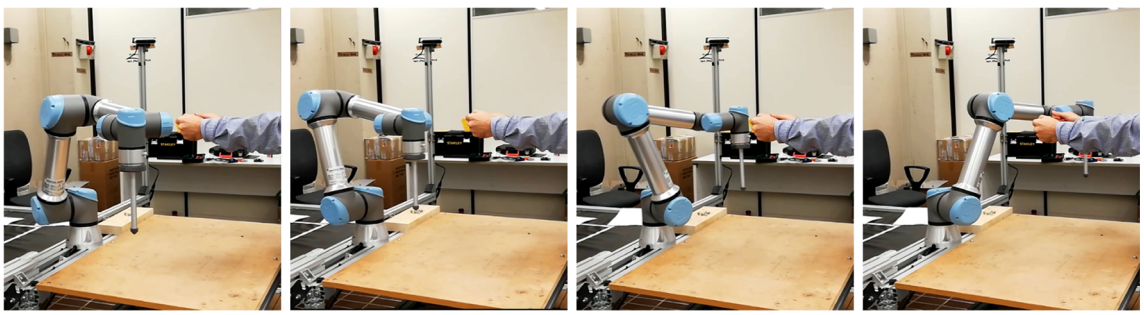

Fig. 14 Test Case 1 - Mode I. Video frames at $t = 0$ s (**a**), $t = 2$ s (**b**), $t = 4$ s (**c**), $t = 6$ s (**d**)

response times, whereas as in the stationary test, the RGB-D system gives a better accuracy. The results of the concordance correlation coefficient (CCC) analysis between the two measurement systems is given in Table 3. The analysis takes into account data related to the motion phase of the test, specifically from $t = 2$ s to $t = 6$ s.

The CCC values for X and Z measurements, which are higher than 0.90, indicate a strong positive correlation and agreement between the values recorded by the RGB-D cameras system and those recorded by Xsens. On the other hand, a CCC of 0.476 is found for the Y measurements, indicating a positive correlation, but a lower agreement between data. This behaviour can be attributed to the substantial alignment of the Y-axis of the robot's reference system with the optical axes of the cameras; as a result, depth estimation via the depth sensor is subject to greater error with respect to in-plane measurements (X and Z directions).

Thus, it is possible to conclude that the detection system based on RGB-D camers is generally more accurate than the Xsens system, which suffers from the main problem of drifting, as is typical for IMU sensors. An error in the order of 30 mm is expected in any case, which have to be compensated with more conservative safety parameters, as for example, a larger radius r of the safety volumes encapsulating the robot.

5 Results

Four different experimental tests were carried out, which can be divided into two main cases: in the first case (Sect. 5.1), the robot moves along a straight trajectory while the operator places his hands along the trajectory interfering with the end-effector; in the second case (Sect. 5.2), the robot is stationary and the operator tries to impact the tool and the robot arm with his hands. For both cases, tests were performed using the first and third obstacle avoidance mode described in Sect. 3. In the remaining part of this section, plots showing the angular position and velocity of each joint for a representative case of each type of tests are presented under two different conditions: without obstacles, allowing the robot to follow the originally planned trajectory, and with obstacles, where the avoidance algorithm is activated. Other graphs are also plotted to show the Cartesian displacement and velocity of the end-effector in relation to the minimum distance between the operator and the robot. For each type of tests, eleven repetitions are carried out in order to obtain a statistical information of the functionality of the entire system.

5.1 Test Case 1

This scenario aims to simulate a possible collision during an assembly task: the human and the robot are working in the same workspace; the robot is moving some parts, and the human accidentally approaches the robot. To avoid the collision, the avoidance algorithm is triggered when the distance between the human and the robot is less than a predefined threshold r_{sup}.

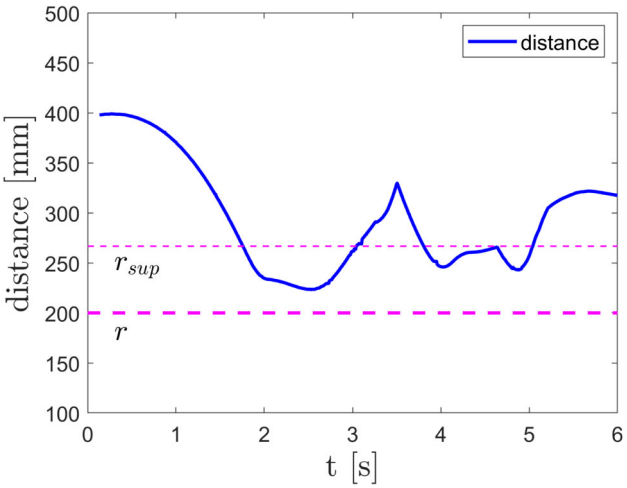

Fig. 15 Test Case 1 - minimum human-robot distance - Mode III

The minimum human-robot distance, plotted in Fig. 11, decreases during the test and goes below r_{sup} (fine dashed purple line) in two phases, namely when the robot first encounters the left wrist and then the right wrist. As a consequence, the collision avoidance algorithm is activated, and a repulsive velocity is assigned to the robot to avoid the collision. In this way, the distance is kept greater than the radius r (thick dashed purple line) of the safety region around the links of the robot. When the risk of collision is overcome, that is, when the human-robot distance is again greater than r_{sup},

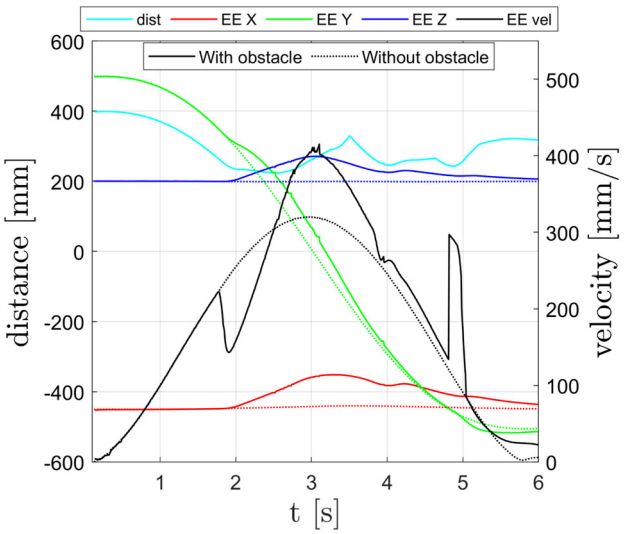

Fig. 16 Test Case 1 - end-effector position (X, Y, Z) and velocity in relation with the minimum human-robot distance - Mode III

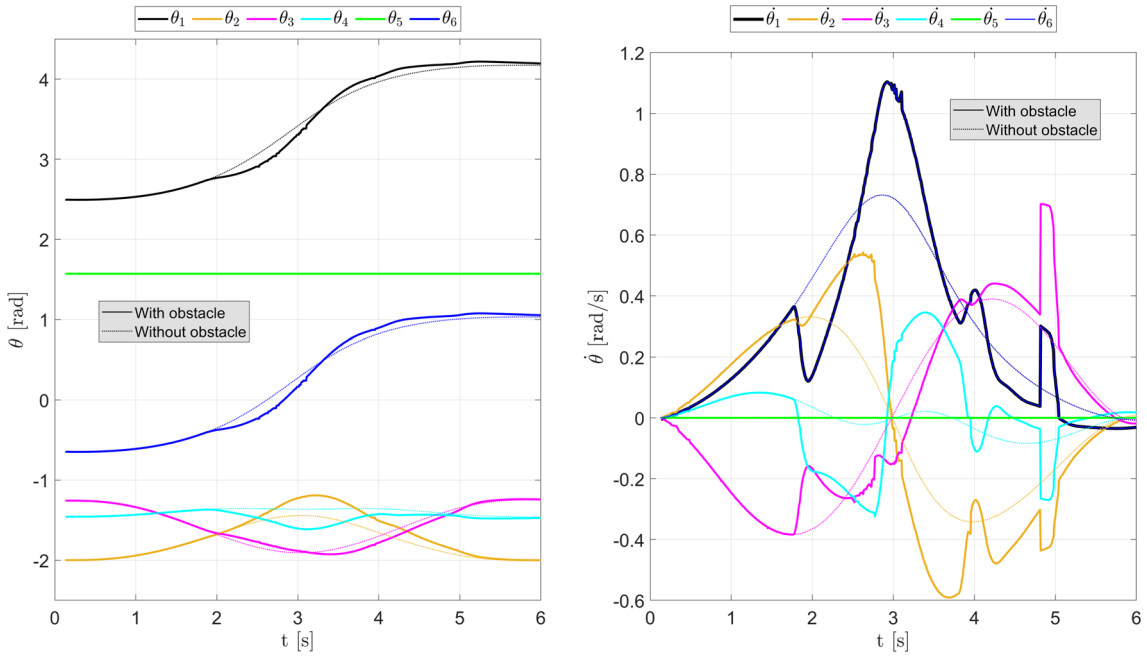

Fig. 17 Test Case 1 - joints angular position (**a**) and velocity (**b**) - Mode III

the resulting positioning error of the robot can be recovered by the proportional correction term **Ke** of the control law. Position and velocity of the end-effector in relation with the minimum human-robot distance for Test Case 1 - Mode I are plotted in Fig. 12, whereas joints angular positions and velocity are shown in Fig. 13. Most significant frames of the test are shown in Fig. 14.

The same test case, so far executed selecting mode I for obstacle avoidance, was executed switching to mode III, i.e. allowing a perturbation of the end-effector with a constant orientation of the tool. The plot of human-robot distance (Fig. 15) is similar to that of mode I, except for the lower values, which confirm that the reduced mobility for perturbation limits the ability to avoid obstacles. Analyzing the angular position and velocity curves (Fig. 17), it can be seen that the velocity peaks are higher than the corresponding peaks in mode I, as can be expected due to the reduced Cartesian mobility of the end-effector which requires higher joint speeds in order to avoid the obstacles. It is clearly visible

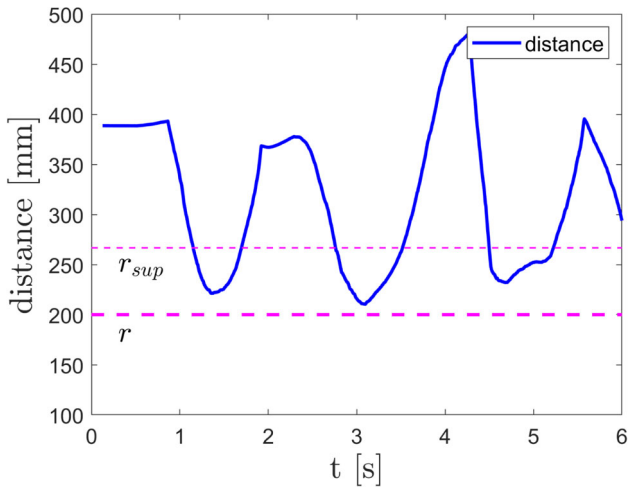

Fig. 18 Test Case 2 - minimum human-robot distance - Mode I

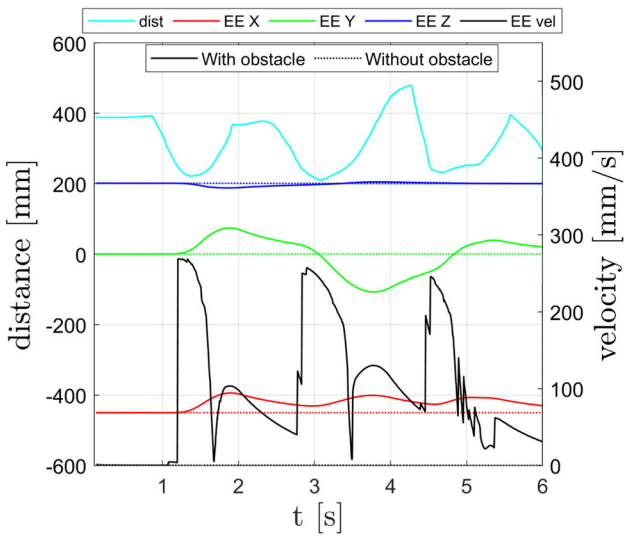

Fig. 19 Test Case 2 - end-effector position (X, Y, Z) and velocity in relation with the minimum human-robot distance - Mode I

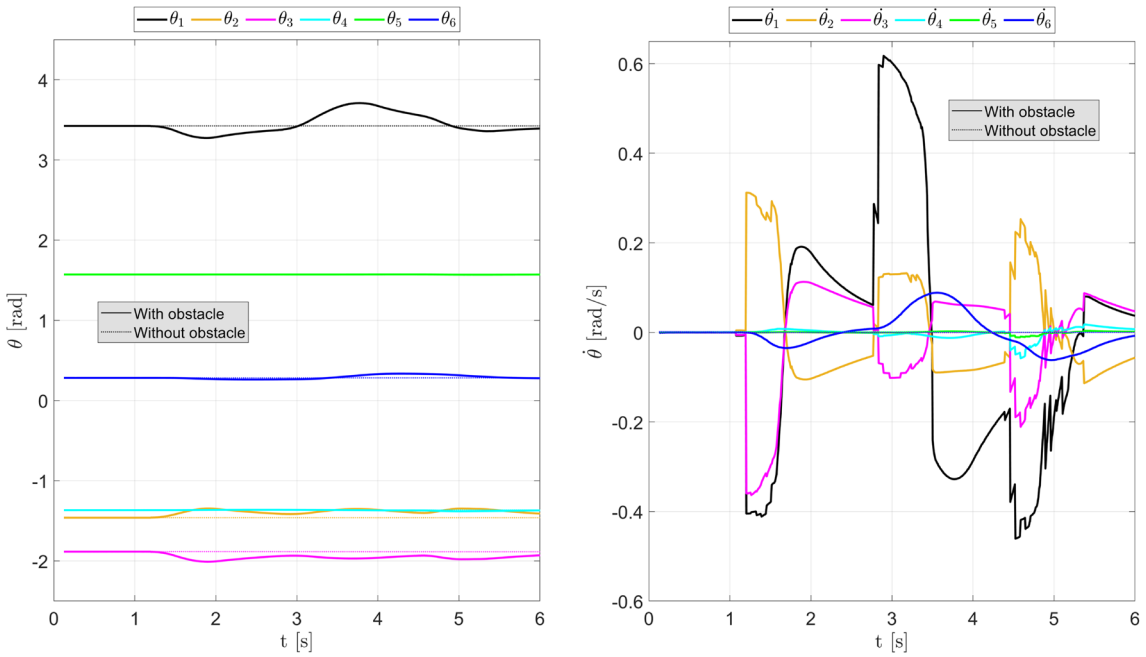

Fig. 20 Test Case 2 - joints angular position (**a**) and velocity (**b**) - Mode I

in Fig. 16 that when the distance falls below r_{sup}, the avoidance control is activated and the motion of the end-effector is perturbed in order to avoid the collision.

5.2 Test Case 2

In the second test case, the robot is stationary while the operator moves his hands around it trying to collide. This situation was chosen to evaluate the response of the skeleton detection and the collision avoidance algorithms to human body motion.

In mode I, the distance falls below r_{sup} three times (Fig. 18), to which three reactions of the motion controller correspond, as visible in Figs. 19 and 20: for example, three peaks of $\dot{\theta}_1$ and $\dot{\theta}_2$ can be noticed at $t \simeq 1$ s, $t \simeq 3$ s, and $t \simeq 4.5$ s, which are the instants corresponding to minimum distances. Frames of Test Case 2 - Mode I are shown in Fig. 21.

The human-robot distance relative to mode III, plotted in Fig. 22, has the same trend as mode I. However, it drops below r in some instants. This is due to the excessive speed of the wrist motion, so the obstacle avoidance algorithm is unable to respond quickly enough to the dynamics of the obstacle. Even though collision with the robot is still avoided (a minimum distance value of 130 mm is reached), the test suggests that higher values of the safety radius should be set for natural operator speed. The plot of end-effector position and velocity related to the human-robot distance is given in Fig. 23. For completeness, joint position and speed are plotted in Fig. 24.

For all conditions tested, the motion of the manipulator is characterized by a smooth behaviour; videos of the tests, available at https://github.com/matteoforlini/human_

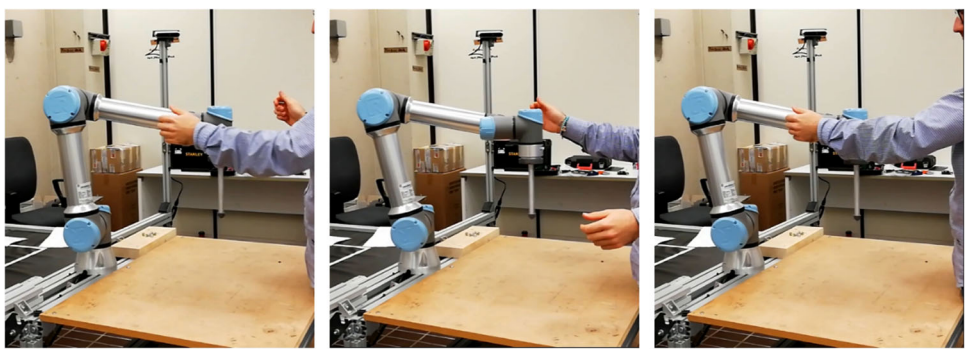

Fig. 21 Test Case 2 - Mode I. Video frames corresponding to instants of minimum human-robot distance: $t = 1$ s (**a**), $t = 3$ s (**b**), $t = 4.5$ s (**c**)

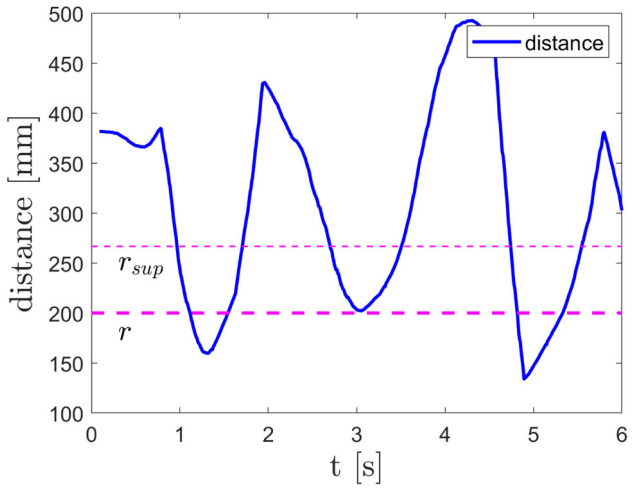

Fig. 22 Test Case 2 - minimum human-robot distance - Mode III

obstacle_avoidance.git, are useful to better understand the robot's reaction to the control.

5.3 Repeatability and reliability tests

To assess the system's reliability and repeatability, eleven executions for each test case were conducted using the same algorithm parameters as previously discussed. Five critical indicators were used to evaluate each test type: minimum distance, maximum joint speed, index of the joint with the highest speed, end-effector linear velocity, and end-effector angular velocity. For each indicator, both the average value across the eleven tests and the minimum/maximum values observed are reported.

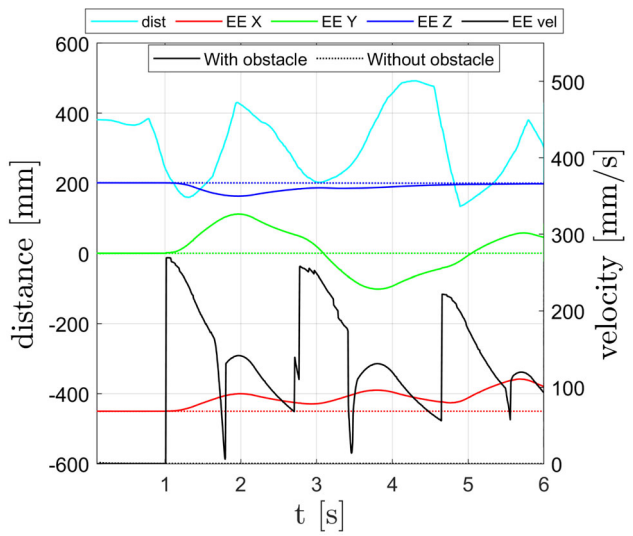

Fig. 23 Test Case 2 - end-effector position (X, Y, Z) and velocity in relation with the minimum human-robot distance - Mode III

As illustrated in Table 4, for Case 1 (in both modes I and III), both the average minimum distance and the absolute minimum distance (the smallest recorded in all tests) are higher than the safety threshold of 200 mm. In Case 2, for modes I and III, the average distance generally remains above the safety threshold, although it sporadically falls below in some tests. This issue, as detailed in Sect. 5.2, is attributed to the speed of human wrists that can be occasionally too fast. In some Case 1 tests, the joint speed reached the maximum allowable value of $\pi/2$ rad/s, but the average speed remained below this limit, indicating the system's capability to avoid obstacles generally under the speed saturation threshold. This ensures that the motion of the manipulator is smooth and natural, preventing any alarm or discomfort for the operator. The recorded values of the end-effector's linear and angular velocity further confirm this. Furthermore, in Case 2, for both modes I and III, the maximum speed was never exceeded, so it would be possible to adopt a higher gain for the anti-collision term of the control in order to ensure a more responsive response, while also increasing the minimum human-robot distance.

6 Conclusions

In this study, a collaborative work cell was accomplished to facilitate human-robot interaction with the main objective of ensuring operator safety and well-being by allowing the robot to avoid any physical contact with the human body.

To achieve this goal, a noncommercial skeleton detection algorithm, based on machine learning techniques, was developed. The algorithm was designed to accurately detect the human skeleton and its movements. To validate its accuracy, the developed system was compared with a commercial pose detection system based on IMU sensors. This analysis showed that the RGB-D system proves to be sufficiently accurate for the purpose of its application, guarantying a NRMSE of 3.23%.

The skeleton tracking algorithm was then integrated with a collision avoidance algorithm to enable real-time obstacle detection and avoidance, so that the robot responds to any operator potential collision changing its motion and avoiding the obstacles.

As a result, the robot is able to guarantee in average a safe minimum distance of 208 mm from the operator's limbs over 44 tests, being 200 mm the threshold limit set by algorithm.

In summary, the tests presented in this paper confirm that the integration and implementation of external skeleton tracking and anti-collision systems and algorithms can be effectively leveraged to improve the control of commercial cobots. The implemented communication protocols ensure high data throughput, which results in responsive behaviour

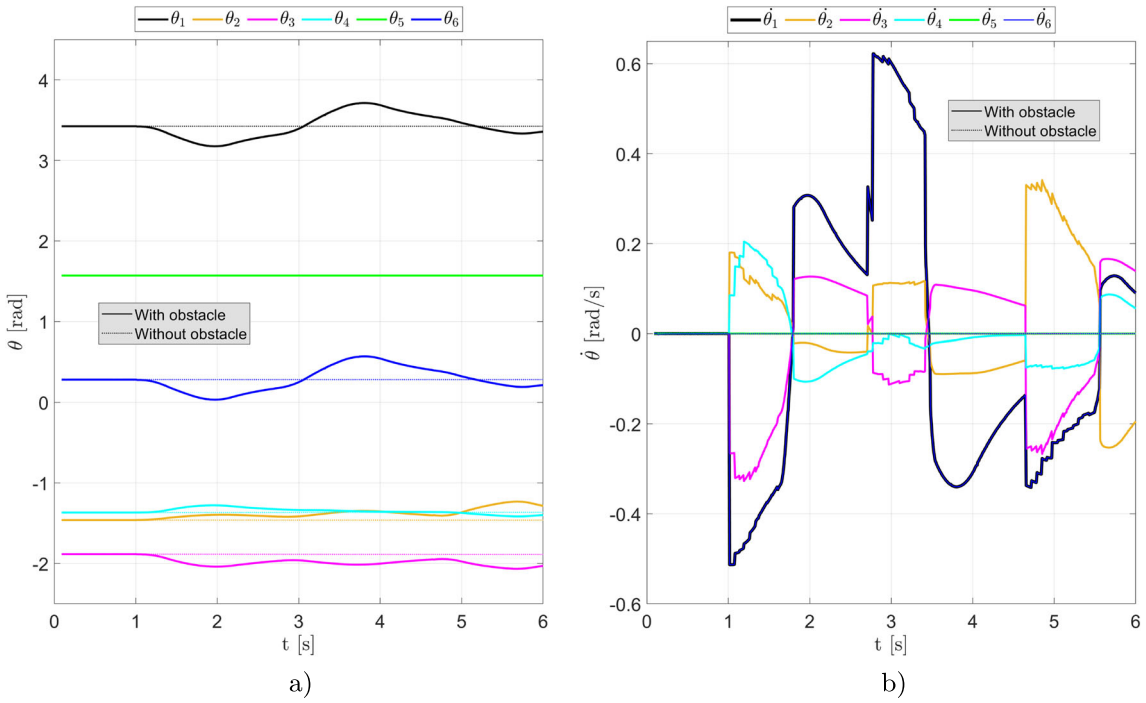

Fig. 24 Test Case 2 - joints angular position (**a**) and velocity (**b**) - Mode III

of the robot to human movement in case of collision risk. In general, the proposed system integrates state-of-the-art solutions for human motion tracking and robot anti-collision control while attempting to solve integration problems and optimize overall system performance. Strengths include the use of a markerless system for motion capture that manage the occlusion problem, based on customizable opensource software tools, the implementation of a control algorithm that considers the entire kinematic chain of the robot

For future developments, various aspects could be considered for enhancement. The update rate of skeleton detection system could be improved (depending also on the PC computing power); for example, in [15], a rate of 25 Hz was obtained using a commercial Kinect software. Moreover, it would be interesting to also consider the relative speed between the robot and the human, as well as the relative position, to improve real-time replanning of the robot's trajectory and ensure greater safety. Furthermore, predicting human motion and implementing a risk factor based on this prediction would be useful for determining whether the avoidance algorithm should be activated. This approach would provide the system with more time to react to potential collisions.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by Matteo Forlini, Federico Neri, and Marianna Ciccarelli. Project organization and supervision were performed by Giacomo Palmieri and Massimo Callegari. The first draft of the manuscript was written by Matteo Forlini and Giacomo Palmieri, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Table 4 Summary of tests results

		Case 1 - Mode I	Case 1 - Mode III	Case 2 - Mode I	Case 2 - Mode III
Mean Min Dis	[mm]	234	226	180	192
Absolute Min Dis	[mm]	225	223	150	135
Mean Max J Speed	[rad/s]	1.20	1.49	0.61	0.52
Absolute Max J Speed	[rad/s]	1.57 (joint 1)	1.57 (joints 1, 6)	0.76 (joint 1)	0.78 (joints 1, 6)
Mean Max EE Vel	[mm/s]	396	435	279	235
Absolute Max EE Vel	[mm/s]	494	447	321	327
Mean Max EE Ang. Vel	[rad/s]	0.32	0	0.61	0
Absolute Max EE Ang. Vel	[rad/s]	0.48	0	0.77	0

Funding Open access funding provided by Università Politecnica delle Marche within the CRUI-CARE Agreement. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data Availability Data for this work will be provided upon request. Materials are uploaded to the following repository link: https://github.com/matteoforlini/human_obstacle_avoidance Additional materials will be provided upon request.

Code Availability Code is uploaded to the following repository link: https://github.com/matteoforlini/human_obstacle_avoidance.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Consent for publication The authors consent to the publication.

References

1. (2011) ISO 10218-1:2011 Robots and robotic devices — safety requirements for industrial robots — Part 1: Robots. ISO organization
2. (2016) ISO/TS 15066:2016 Robots and robotic devices — collaborative robots. ISO organization
3. (2019) Collaborative industrial robot definition and estimates supply. International Federation of Robotics Secretariat Blog
4. (2023) Mediapipe, pose detection. https://google.github.io/mediapipe/solutions/pose.html
5. (2023) Xsens MVN motion capture. https://www.movella.com/products/motion-capture
6. Aaltonen I, Salmi T, Marstio I (2018) Refining levels of collaboration to support the design and evaluation of human-robot interaction in the manufacturing industry. Procedia CIRP 72:93–98
7. Antonsson EK, Mann RW (1985) The frequency content of gait. J Biomech 18(1):39–47
8. Bazarevsky V, Grishchenko I, Raveendran K, et al (2020) Blazepose: on-device real-time body pose tracking. arXiv:2006.10204
9. Bugarin CAQ, Lopez JMM, Pineda SGM, et al (2022) Machine vision-based fall detection system using Mediapipe pose with IoT monitoring and alarm. In: 2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC), IEEE, pp 269–274
10. Cherubini A, Navarro-Alarcon D (2021) Sensor-based control for collaborative robots: fundamentals, challenges, and opportunities. Frontiers in Neurorobotics p 113
11. Chiriatti G, Palmieri G, Scoccia C et al (2021) Adaptive obstacle avoidance for a class of collaborative robots. Machines 9(6):113
12. Cifuentes CA, Frizera A, Carelli R et al (2014) Human-robot interaction based on wearable IMU sensor and laser range finder. Robot Auton Syst 62(10):1425–1439
13. Deo AS, Walker ID (1995) Overview of damped least-squares methods for inverse kinematics of robot manipulators. J Intell Rob Syst 14(1):43–68
14. Docekal J, Rozlivek J, Matas J, et al (2022) Human keypoint detection for close proximity human-robot interaction. In: 2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids), IEEE, pp 450–457
15. Ferraguti F, Landi CT, Costi S et al (2020) Safety barrier functions and multi-camera tracking for human-robot shared environment. Robot Auton Syst 124:103388
16. Filippeschi A, Schmitz N, Miezal M et al (2017) Survey of motion tracking methods based on inertial sensors: a focus on upper limb human motion. Sensors 17(6):1257
17. Flacco F, De Luca A (2016) Real-time computation of distance to dynamic obstacles with multiple depth sensors. IEEE Robotics Automation Letters 2(1):56–63
18. Flacco F, Kroeger T, De Luca A et al (2015) A depth space approach for evaluating distance to objects. J Intell Robotic Syst 80(1):7–22
19. Gallagher A, Matsuoka Y, Ang WT (2004) An efficient real-time human posture tracking algorithm using low-cost inertial and magnetic sensors. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), IEEE, pp 2967–2972
20. Gasparetto A, Boscariol P, Lanzutti A, et al (2015) Path planning and trajectory planning algorithms: a general overview. Motion Oper Plan Robot Syst:3–27
21. Gualtieri L, Palomba I, Merati FA, et al (2020) Design of human-centered collaborative assembly workstations for the improvement of operators' physical ergonomics and production efficiency: a case study. Sustainability (Switzerland) 12(9). https://doi.org/10.3390/su12093606
22. Kagermann H, Wahlster W, Helbig J, et al (2013) Recommendations for implementing the strategic initiative Industrie 4.0. Final report of the Industrie 4(0):82
23. Khatib M, Al Khudir K, De Luca A (2021) Human-robot contactless collaboration with mixed reality interface. Robot Comput-Integrated Manufact 67:102030
24. Lasota PA, Fong T, Shah JA, et al (2017) A survey of methods for safe human-robot interaction. Foundations Trends® in Robotics 5(4):261–349
25. Lenz C, Grimm M, Röder T, et al (2012) Fusing multiple kinects to survey shared human-robot-workspaces. Computer Science, Engineering
26. Liu H, Fang T, Zhou T et al (2018) Towards robust human-robot collaborative manufacturing: multimodal fusion. IEEE Access 6:74762–74771
27. Morato C, Kaipa KN, Zhao B, et al (2014) Toward safe human robot collaboration by using multiple kinects based real-time human tracking. Journal of Computing and Information Science in Engineering 14(1)
28. Mukherjee D, Gupta K, Chang LH et al (2022) A survey of robot learning strategies for human-robot collaboration in industrial settings. Robotics Computer-Integrated Manufacturing 73:102231
29. Nascimento H, Mujica M, Benoussaad M (2020) Collision avoidance interaction between human and a hidden robot based on kinect and robot data fusion. IEEE Robotics Automation Letters 6(1):88–94
30. Polverini MP, Zanchettin AM, Rocco P (2014) Real-time collision avoidance in human-robot interaction based on kinetostatic safety field. In: 2014 IEEE/RSJ International conference on intelligent robots and systems, IEEE, pp 4136–4141

31. Polverini MP, Zanchettin AM, Rocco P (2017) A computationally efficient safety assessment for collaborative robotics applications. Robotics Computer-Integrated Manufacturing 46:25–37
32. Qiao D, Pang GK, Kit MM, et al (2008) A new PCB-based low-cost accelerometer for human motion sensing. In: 2008 IEEE International conference on automation and logistics, IEEE, pp 56–60
33. Rybski P, Anderson-Sprecher P, Huber D, et al (2012) Sensor fusion for human safety in industrial workcells. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp 3612–3619
34. Safeea M, Neto P (2019) Minimum distance calculation using laser scanner and IMUs for safe human-robot interaction. Robotics Computer-Integrated Manufacturing 58:33–42
35. Scimmi LS, Melchiorre M, Mauro S, et al (2018) Multiple collision avoidance between human limbs and robot links algorithm in collaborative tasks. In: ICINCO (2), pp 301–308
36. Scoccia C, Palmieri G, Palpacelli MC et al (2021) A collision avoidance strategy for redundant manipulators in dynamically variable environments: on-line perturbations of off-line generated trajectories. Machines 9(2):30
37. Singh AK, Kumbhare VA, Arthi K (2021) Real-time human pose detection and recognition using Mediapipe. In: International conference on soft computing and signal processing, Springer, pp 145–154
38. Wang W, Zhu M, Wang X et al (2018) An improved artificial potential field method of trajectory planning and obstacle avoidance for redundant manipulators. Int J Adv Rob Syst 15(5):1729881418799562
39. Zamora M, Caldwell E, Garcia-Rodriguez J, et al (2017) Machine learning improves human-robot interaction in productive environments: a review. In: International work-conference on artificial neural networks, Springer, pp 283–293
40. Zhang JT, Novak AC, Brouwer B et al (2013) Concurrent validation of Xsens MVN measurement of lower limb joint angular kinematics. Physiol Meas 34(8):N63
41. Zheng P, Wieber PB, Baber J et al (2022) Human arm motion prediction for collision avoidance in a shared workspace. Sensors 22(18):6951