electronics

Article

Planning Collision-Free Robot Motions in a Human–Robot Shared Workspace via Mixed Reality and Sensor-Fusion Skeleton Tracking

Saverio Farsoni *, Jacopo Rizzi, Giulia Nenna Ufondu and Marcello Bonfè

Department of Engineering, University of Ferrara, 44122 Ferrara, Italy; jacopo.rizzi@unife.it (J.R.); giulianenna.ufondu@edu.unife.it (G.N.U.); marcello.bonfe@unife.it (M.B.)
* Correspondence: saverio.farsoni@unife.it

Abstract: The paper describes a method for planning collision-free motions of an industrial manipulator that shares the workspace with human operators during a human–robot collaborative application with strict safety requirements. The proposed workflow exploits the advantages of mixed reality to insert real entities into a virtual scene, wherein the robot control command is computed and validated by simulating robot motions without risks for the human. The proposed motion planner relies on a sensor-fusion algorithm that improves the 3D perception of the humans inside the robot workspace. Such an algorithm merges the estimations of the pose of the human bones reconstructed by means of a pointcloud-based skeleton tracking algorithm with the orientation data acquired from wearable inertial measurement units (IMUs) supposed to be fixed to the human bones. The algorithm provides a final reconstruction of the position and of the orientation of the human bones that can be used to include the human in the virtual simulation of the robotic workcell. A dynamic motion-planning algorithm can be processed within such a mixed-reality environment, allowing the computation of a collision-free joint velocity command for the real robot.

Keywords: mixed reality; collaborative robotics; skeleton tracking; sensor-fusion; 3D perception

check for updates

Citation: Farsoni, S.; Rizzi, J.; Ufondu, G.N.; Bonfè, M. Planning Collision-Free Robot Motions in a Human–Robot Shared Workspace via Mixed Reality and Sensor-Fusion Skeleton Tracking. *Electronics* **2022**, *11*, 2407. https://doi.org/10.3390/electronics11152407

Academic Editor: Cecilio Angulo

Received: 29 June 2022
Accepted: 30 July 2022
Published: 1 August 2022

1. Introduction

The advent of collaborative robotics enhances the safety requirement for human operators that have to cooperate with robots sharing the same workspace [1]. Indeed, prolonged endeavors are made to develop robots that intrinsically provide safety for humans. Such a paradigm involves research activities focused on collision detection, how to distinguish them from intentional contacts, and how to react to unwanted ones [2]. However, because the intrinsic safety of the robots cannot be completely reached, particularly when dealing with robots moving heavy payloads, it can be convenient to generate alternative trajectories to avoid collisions with humans at all [3]. In this context, recent advances in mixed, augmented, and virtual reality enable challenging scenarios for improving human–robot interactions from human-centered perspectives. A new way to implement human–robot communication can be enabled by exploiting the advantages of mixed reality for controlling, configuring, and planning the motion of the robot [4]. In [5] the authors conducted a deep literature search, selecting studies from the last two decades to show the main applications and the critical factors involved in the usage of mixed reality in collaborative robotics.

This paper proposes the usage of mixed reality, understood as the projection of contents from reality towards a virtual environment [6], to implement a dynamic motion planner, processed into virtuality, but whose output can be used to command the real robot to avoid dangerous collisions with humans. As a matter of fact, an effective and safe human–robot collaboration requires that robots generate human-aware behaviors during planning [7]. Therefore, having a fast and accurate 3D perception of the humans that act inside the robot workspace is of fundamental importance to avoid dangerous collisions between the robot and the operators. A suitable approach to cope with this issue is the usage of exteroceptive sensors, such as laser scanners, light projectors [8] or, more recently, depth cameras [9], that can enable advanced computer-vision (CV) algorithms to provide the real-time detection and tracking of the different human bones in a 3D environment [10]. Such a problem is referred to as skeleton tracking. Nuitrack, among others, is a 3D tracking software developed by 3DiVi Inc. (https://www.3divi.com) whose SDK provides a set of APIs for skeleton tracking, gesture recognition and communication with different commercial depth cameras. Recent studies [11] have demonstrated that the accuracy of measurements retrieved from low-cost depth cameras such as Realsense D415/435 can be not appropriate to enable robust skeleton tracking.

Different approaches to the problem of skeleton tracking involve the usage of wearable sensors, in particular inertial measurement units (IMUs) fixed to human bones [12]. Such sensors allow us to obtain an accurate estimation of the orientation of the rigid body to which they are connected, by processing the measurements of the inner components (i.e., the accelerometer, the gyroscope and, in most cases, the magnetometer) by means of a sensor-fusion algorithm. The most common ways to extract the orientation of a rigid body from IMUs involves the usage of Kalman filters or non-linear complementary filters [13]. It is worth observing that IMUs cannot provide a complete pose reconstruction, so that a further integration with different kinds of sensors is required to include the position in the final pose estimation.

Therefore, a further contribution of the proposed work is the design of a sensor-fusion algorithm to merge the pose reconstruction provided by the CV software Nuitrack with the partial, but accurate, estimation of bone pose provided by custom-made wearable IMU boards. For this purpose, a bank of multirate Kalman filters is proposed to fuse Nuitrack data with the orientation estimations acquired from the IMU-boards fixed to the considered human bones.

It is worth observing that the proposed mixed-reality workflow could be further exploited to program and validate robot motions of complex robotics workcells for mass production. Indeed, several research efforts are being made to improve the productivity of robotized workcells [14,15]. Mixed reality can be a valuable tool by which to analyze deadlock conditions caused by unpredictable obstructions (e.g., humans that move inside the robot workspace), so as to allow the design of optimized robot motion aiming at avoiding both collisions and deadlock conditions.

The remainder of the paper is organized as follows: Section 3.1 describes the design of the wearable devices, Section 2 summarizes the problem statement and the contribution of the paper. Section 3.2 describes the proposed sensor-fusion algorithm that merges the depth camera and IMU data to improve the accuracy of the estimation of the human bone poses. Section 3.3 describes the integration of the algorithm in a mixed-reality application with the aim of enabling a dynamic collision-avoidance motion planner. Section 4 depicts the experimental setup and shows the obtained results. The proposed methods have been compared with alternative strategies in Section 5. Finally Section 6 makes the final remarks.

2. Problem Statement and Contributions

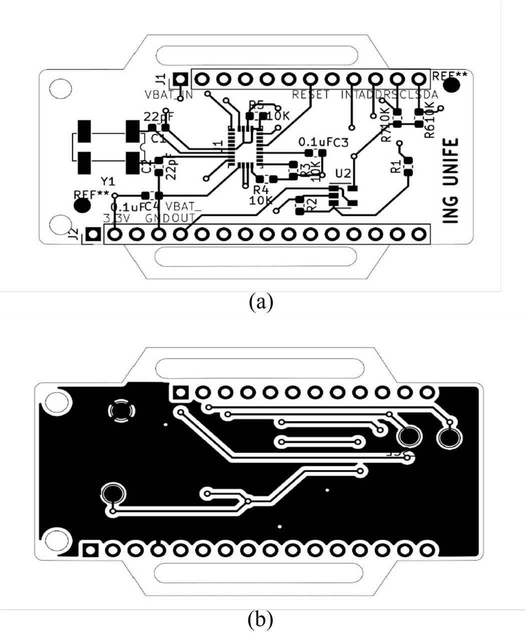
The paper addresses the challenge of planning the motions of an industrial manipulator inserted in a human–robot collaborative application where humans could share the workspace with the robot. The main contribution of the paper is the design of a workflow that exploits mixed reality as a tool to safely program, test and validate the adopted motion planning algorithm. The proposed method requires an accurate projection of the real content in the virtual scene. In particular, the pose of the human bones inside the robot workspace has to be fast and accurately tracked to enable the generation of a safe robot reaction. The 3D perception of humans can be achieved by means of pointcloud-based skeleton trackers. However, the usage of low-cost depth cameras could not result in an adequate tracking. Therefore, as a further contribution of the work, a custom-made wearable device has been used to acquire and transmit bone orientation data and a sensor-fusion algorithm has been designed to improve the 3D perception of the human.

3. Materials and Methods

3.1. Custom-Made Wearable Devices

The wearable device used in the experiments consists in a custom-made board that includes the IMU and a WiFi module. In more detail, the adopted IMU is the smart sensor Bosch BNO055 [16], a system in package (SiP) solution that integrates a three-axis accelerometer, a three-axis gyroscope and a three-axis geomagnetic sensor. The chip also includes a 32-bit microcontroller that executes a proprietary algorithm providing the absolute orientation of the board in form of unit quaternions via I2C. The printed circuit board is depicted in Figure 1, whose components are listed in Table 1. The pinout of the board has been designed to allow the direct connection with the Adafruit Feather Huzzah ESP8286 development board, which features built-in USB and battery charging and includes an 80 MHz ESP8266 WiFi microcontroller, very popular for Internet of Things (IoT) applications [17].

The firmware running on the ESP8266 microcontroller acquires the orientation data from the BNO055 IMU via I2C and then transmits such information to the PC where the overall application is running. The wireless communication between the PC and the wearable IMU boards is achieved via UDP. The PC acts as the server and is in charge of sending a multicast synchronization message to all the connected wearable devices, acting as clients. Afterward, all the connected wearable devices start transmitting orientation data with timestamp to the server, with a loop rate of 100 Hz. Moreover, the server can command a single wearable board to perform the calibration procedure for the BNO055, resulting in saving the sensor offsets in the microcontroller EEPROM. The scheme of Figure 2 shows the architecture and the interface between the PC and the wearable devices.



(a)

(b)

Figure 1. The custom-made board including the BNO055 IMU. (**a**) top layer, (**b**) bottom layer.

Table 1. List of components included in the board, with reference to Figure 1.

Symbol	Qty	Manufacturer	Features
Y1	1	IQD frequency products	Crystal SMD, 32.768 KHz, 12.5 pF
C1, C2	2	Generic	22.0 pF, 50 V
C3, C4	2	Generic	100.0 nF, 50 V
R3, R4, R5, R6, R7	5	Generic	10 kOhm, 50 V
U1	1	Bosch Sensortec	BNO055, IMU Accel/Gyro/Mag I2C
U2	1	Microchip	General purpose amplifier

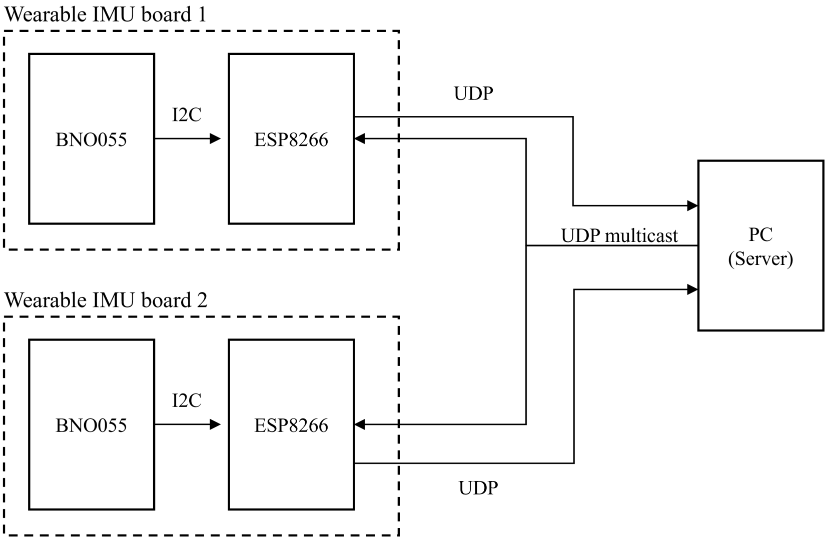


Figure 2. The interface between the PC and the wearable IMU boards.

3.2. Sensor-Fusion Algorithm

The multirate Kalman filter is the suitable framework with which to merge measurements coming from different kinds of sensors, running at different rates [18]. It consists in a loop of two main steps, namely the prediction and the correction. The former is used to predict the value of a set of variables (i.e., the state of the filter) at the current time instant on the basis of the analytical model describing their evolution in time, whereas the latter adjusts the prediction on the basis of the available measurements by means of the so-called Kalman filter gain, which computes management of the stochastic characteristics of the process and of the measurements. The multirate variant of the Kalman filter allows us to correct the predicted values of the state variables by processing the measurement of different sensors as soon as they become available. In particular, the filter parameters have to be tuned, taking into account that the depth camera provides through Nuitrack the measurements of the position of the detected human joints and the orientation of the bones with a high level of noise, whereas the IMUs provide just the orientation of the bones, but with a low level of noise. The multirate Kalman filter model previously developed by the authors [19] can be modified to adapt to the different application. Two Kalman filters per bone are exploited here to estimate, on the one hand, the linear kinematics variables (i.e., the position, the linear velocity and the linear acceleration) and, on the other hand, the angular kinematics variables (i.e., the orientation, the angular velocity and the angular acceleration).

In more detail, the linear filter uses the following model to predict the evolution in time of the linear kinematics variables:

$$\hat{\mathbf{x}}_{l,k+1}^{-} = \mathbf{A}_l \hat{\mathbf{x}}_{l,k}, \tag{1}$$

where k is the current time step, and the symbols $\hat{\cdot}$ and $\cdot^-$ mean the estimated and the predicted value, respectively. $\mathbf{x}_l = [\mathbf{p}, \mathbf{v}, \mathbf{a}]^T$ is the linear filter state and the dynamics matrix $\mathbf{A}_l$ can be written as

$$\mathbf{A}_l = \begin{bmatrix} \mathbf{I}_{3\times3} & \Delta t\mathbf{I}_{3\times3} & \frac{1}{2}\Delta t^2\mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \Delta t\mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix}, \tag{2}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{0}$ the matrix of zeros, and Δt is time elapsed since the last filter estimation.

The angular filter is an extended Kalman filter (EKF) because it exploits a non-linear model to describe the evolution of its state $\mathbf{x}_a$:

$$\hat{\mathbf{x}}^-_{a,k+1} = \begin{bmatrix} \hat{\mathbf{q}} \\ \hat{\boldsymbol{\omega}} \\ \hat{\dot{\boldsymbol{\omega}}} \end{bmatrix}_{k+1} = \mathbf{f}(\hat{\mathbf{x}}_k) = \begin{bmatrix} \hat{\mathbf{q}}_k \otimes e^{\frac{1}{2}(\hat{\boldsymbol{\omega}}_k\Delta t + \frac{1}{2}\hat{\dot{\boldsymbol{\omega}}}_k\Delta t^2)} \\ \hat{\boldsymbol{\omega}}_k + \hat{\dot{\boldsymbol{\omega}}}_k\Delta t \\ \hat{\dot{\boldsymbol{\omega}}}_k \end{bmatrix}, \tag{3}$$

where $\mathbf{q}$ is the bone quaternion, $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ are the angular velocity and acceleration and the symbol $\otimes$ indicates the quaternion product. It is worth noting that the exponential formula for describing the evolution of the quaternion is used to preserve the unitary norm of the quaternion. More details can be found in [19].

The correction of the predicted state takes into account the acquired measurements $\mathbf{y}_l$, $\mathbf{y}_a$ and the Kalman gain matrices $\mathbf{K}_l$, $\mathbf{K}_a$ computed as follows for the linear variables:

$$\begin{aligned} \mathbf{P}^-_{l,k+1} &= \mathbf{A}_l\mathbf{P}_{l,k}\mathbf{A}_l^T + \mathbf{Q}_l \\ \mathbf{K}_{l,k} &= \mathbf{P}^-_{l,k+1}\mathbf{C}_l^T(\mathbf{C}_l\mathbf{P}^-_{l,k+1}\mathbf{C}_l^T + \mathbf{R}_l)^{-1} \\ \hat{\mathbf{x}}_{l,k+1} &= \hat{\mathbf{x}}^-_{l,k+1} + \mathbf{K}_{l,k}(\mathbf{y}_{l,k+1} - \mathbf{C}_l\hat{\mathbf{x}}^-_{l,k+1})' \\ \mathbf{P}_{l,k+1} &= (\mathbf{I} - \mathbf{K}_{l,k}\mathbf{C}_l)\mathbf{P}^-_{l,k+1} \end{aligned} \tag{4}$$

where $\mathbf{P}_l$ is the error covariance matrix, $\mathbf{Q}_l$ is the process noise covariance matrix, and $\mathbf{R}_l$ is the measurement noise covariance matrix. $\mathbf{y}_l$ contains the measured position $\mathbf{p}_{\text{meas}}$, that consists in a selection of the state variables achieved by means of the matrix $\mathbf{C}_l = [\mathbf{I}_{3\times3}\mathbf{0}_{3\times6}]$, so that, in absence of noise and model error $\mathbf{y}_l = \mathbf{C}_l\mathbf{x}_l$. It is worth noting that $\mathbf{p}_{\text{meas}}$ is the 3D position of the bone elaborated by Nuitrack on the basis of the pointcloud acquired from the depth camera.

In order to obtain the estimation of the angular variables it should be noticed that Equation (3) is expressed through a nonlinear function. Therefore, the dynamics matrix $\mathbf{A}_a$ has to be computed by linearizing $\mathbf{f}$ as follows:

$$\mathbf{A}_a = \left.\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}. \tag{5}$$

Then the estimation of the angular variables produces the next state estimation similarly to the linear filter correction of Equation (4), taking care to replace $\mathbf{A}_l$,$\mathbf{K}_l$,$\mathbf{P}_l$,$\mathbf{R}_l$,$\mathbf{Q}_l$,$\mathbf{x}_l$,$\mathbf{y}_l$,$\mathbf{C}_l$ with $\mathbf{A}_a$,$\mathbf{K}_a$,$\mathbf{P}_a$,$\mathbf{R}_a$,$\mathbf{Q}_a$,$\mathbf{x}_a$,$\mathbf{y}_a$,$\mathbf{C}_a$, where $\mathbf{y}_a$ contains the measured quaternion $\mathbf{q}_{\text{meas}}$ and $\mathbf{C}_a = [\mathbf{I}_{4\times4}, \mathbf{0}_{4\times6}]$.

When $\mathbf{q}_{\text{meas}}$ is acquired by means of the IMU fixed to the bone, the matrix $\mathbf{R}_a$ takes the value of the covariance matrix of the IMU measurement noise. Similarly, when $\mathbf{q}_{\text{meas}}$ is elaborated by Nuitrack, $\mathbf{R}_a$ takes into account the measurement noise associated to the depth camera. Therefore, the final estimation of the orientation is actually generated by means of a sensor-fusion. The block diagram of Figure 3 summarizes the sensor-fusion algorithm to track the pose of a single human bone. The same architecture has to repeated for each of the monitored bones (e.g., left/right arm/forearm/leg, thorax, head . . .).

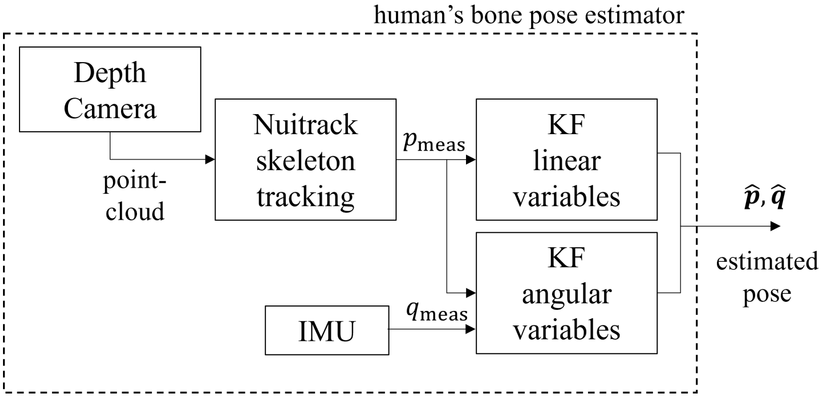


Figure 3. The block diagram of the sensor-fusion algorithm to estimate the pose of a human bone.

Alternative Formulations

Known issues of the EKF formulation are related to the linearized approximation based on the Jacobian matrix, whose calculation may require a strong computational effort and whose use in the estimate and covariance propagation may lead to sub-optimal performance and sometimes divergence of the filter. Such issues are addressed, as described in [20], by the so-called unscented Kalman filter (UKF), based on random variables propagation through the actual nonlinear model around carefully chosen sigma points.

Another possible alternative formulation of the filter could use the quaternion integration approach proposed in [21], in which the authors describe the evolution of the quaternion between two consecutive time steps by means of two intermediate approximate interpolations.

In more detail, the next quaternion $\mathbf{q}_{k+1}$ depends on the angular rates at a quarter ($\boldsymbol{\omega}_{k+\frac{1}{4}}$) and at the mid-point ($\boldsymbol{\omega}_{k+\frac{1}{2}}$) of the next time step, which are computed based on the assumption of constant angular acceleration $\dot{\boldsymbol{\omega}}_k$:

$$\begin{aligned} \boldsymbol{\omega}_{k+\frac{1}{4}} &= \boldsymbol{\omega}_k + \tfrac{1}{4}\dot{\boldsymbol{\omega}}_k \Delta t \\ \boldsymbol{\omega}_{k+\frac{1}{2}} &= \boldsymbol{\omega}_k + \tfrac{1}{2}\dot{\boldsymbol{\omega}}_k \Delta t \end{aligned}. \tag{6}$$

Then, the quaternion at the mid-point is calculated by exploiting the exponential formula and neglecting the second-order terms:

$$\mathbf{q}_{k+\frac{1}{2}} = \mathbf{q}_k \otimes e^{\frac{1}{4}\boldsymbol{\omega}_{k+\frac{1}{4}}\Delta t}. \tag{7}$$

Afterward, the mid-point angular rate is transformed into the world coordinate system by using the mid-point quaternion rotation:

$$\boldsymbol{\omega}^{w}_{k+\frac{1}{2}} = \mathbf{q}_{k+\frac{1}{2}} \otimes \boldsymbol{\omega}_{k+\frac{1}{2}} \otimes \mathbf{q}^{-1}_{k+\frac{1}{2}}, \tag{8}$$

where the superscript w represents the world frame system.

Finally, the next quaternion is given by

$$\mathbf{q}_{k+1} = e^{\frac{1}{2}\boldsymbol{\omega}^{w}_{k+\frac{1}{2}}\Delta t} \otimes \mathbf{q}_k. \tag{9}$$

This approach can be implemented into the Kalman filter model by replacing the quaternion dynamics of Equation (3) with the expression of Equation (9).

3.3. Mixed-Reality Motion Planner

In collaborative robotics, the robot could share the workspace with humans whose movements could be often unpredictable. Therefore it is of fundamental importance to implement the robot motion planner so that it can promptly react to approaching humans to avoid dangerous collisions. The proposed dynamic motion planner involves the generation of repulsive artificial potential fields (APF) [22], that are computed by exploiting the advantages of a mixed-reality environment, in which the real humans are included in a scene where the virtual robot can simulate movements without risks to the human safety, in order to facilitate the computation of a collision-free velocity command for the real robot. The robot simulator used to create the mixed-reality scene is CoppeliaSim [23] whose main features are:

- 3D and kinematics robot models, including a dedicated plugin for inverse kinematics calculations;
- scripting Lua and C++ APIs to create custom scripts for the scene objects; and
- remote interfacing, including a ROS plugin to interface the simulation with ROS publishers and subscribers

The main idea is to let the robot execute its original task when humans are further away than a certain distance threshold. Then, when humans are detected and their bones are tracked inside the robot workspace, the task is suspended and the robot starts to be driven by repulsive artificial fields that move it away from the closest bone. Obviously, due to the limited robot physical capabilities, not all the collisions can be avoided, so it is important to set a safety distance threshold ϵ that, when exceeded by a human bone, forces the robot to stop its movements immediately.

For this purpose, in the virtual scene, the robot links and the tracked bones are encapsulated by convex shapes, allowing fast computation of the distances among them. In more detail, capsules (i.e., a cylinder with half spheres on the bases) are suitable convex shapes to cover bones and links because distances between two capsules and their closest points can be computed by means of analytical expressions, as described in [24]. Once the current minimum distances between the capsules of robot links and human bones have been computed, the dynamic motion planner can generate the robot velocity command $\dot{\mathbf{q}}_{\text{ref}}$ to move away the robot from the current closest obstacle by means of the following formula:

$$\dot{\mathbf{q}}_{\text{ref}} = \frac{k_r}{\eta^2(\mathbf{q})}(\frac{1}{\eta(\mathbf{q})} - \frac{1}{\eta_0})^2 \nabla\eta(\mathbf{q}), \tag{10}$$

where $\mathbf{q}$ contains the joint positions, $\eta(\mathbf{q})$ is the current minimum distance between the human bone and the robot links. Such a minimum distance is supposed to be shorter than the threshold η_0; otherwise the robot could be programmed to perform the desired task, but higher than ϵ otherwise the robot has to be stopped. $\nabla\eta_i(\mathbf{q})$ is the gradient vector of the minimum distance. The parameter $k_r > 0$ is a positive scaling factor that determines how intense is the reaction to an approaching obstacle. It has to be tuned by taking into account the physical capabilities of the robot and the safety requirements of the application.

It is worth noting that $\nabla\eta_i(\mathbf{q})$ can be safely computed by exploiting the advantages of mixed reality. Indeed, the virtual robot can simulate small displacements of its joints without risk to the human, so to allow the calculation of the variations of the minimum distance in response to such displacements. Possible future developments of the work could involve the extension to the case study of multiple concave obstacles, addressing also the problem of avoiding local minima in the related APF planning algorithms, as described e.g., in [25].

Note finally that the proposed architecture has a generalized structure that can fit with different kinds of industrial manipulators, e.g., SCARA and Delta, possibly mounted on a mobile base.

The implementation of the overall architecture is depicted in the block diagram of Figure 4, where the green block on the left contains the reality elements, i.e., the human and the robot that share the workspace, whereas the yellow block on the right represents the virtual environment of CoppeliaSim. The bank of Kalman filters within the green block (i.e., four instances of the block diagram in Figure 3) processes the measurements of the wearable devices and of the depth camera to reconstruct the pose of each bone, whereas the robot configuration can be acquired from kinematics data. Once the real elements are inserted into the virtual scene, they are covered by convex capsules so that the minimum distance can be calculated. The motion planner lives in the virtual context, where its output can be safely computed by simulating small joint displacements, and the effects of the artificial potential field can be validated without the risk of collisions.

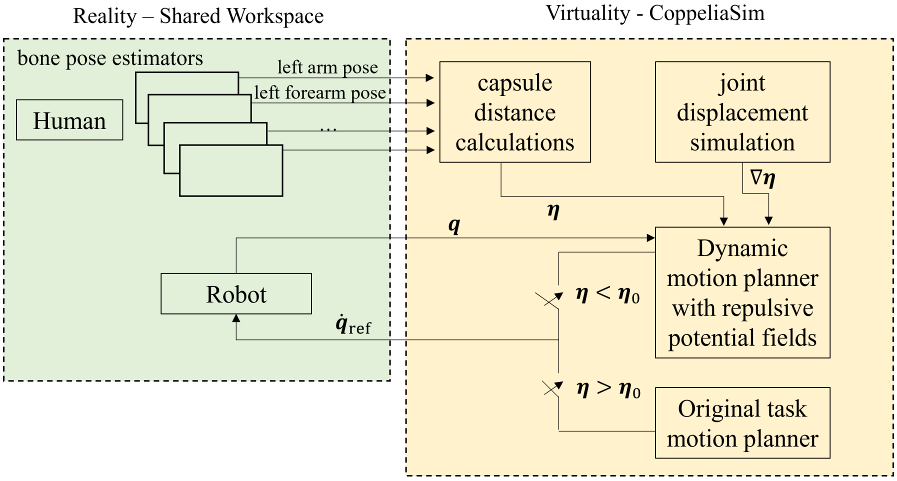

Reality – Shared Workspace
Virtuality - CoppeliaSim
bone pose estimators
Human
left arm pose
left forearm pose
...
Robot
capsule distance calculations
joint displacement simulation
$\nabla\eta$
η
q
Dynamic motion planner with repulsive potential fields
$\eta < \eta_0$
$\dot{q}_{\text{ref}}$
$\eta > \eta_0$
Original task motion planner

Figure 4. The architecture of the overall system. The estimated pose of the bones and the robot joint configuration are used to insert real elements into the virtual scene of CoppeliaSim so that the motion planner can compute the collision-free command.

4. Experiments

In order to assess the performance of the overall architecture methods, the output of the motion planner has been connected to a simulated Franka Emika Panda robot. However, because the communication between the different entities of the scene is implemented by means of ROS topics, the real robot can be straightforwardly integrated into the proposed architecture. The Franka Emika Panda is a seven degrees-of-freedom (DOF) robot with torque sensors at each joint allowing compliance and admittance control for collaborative applications. It can handle a maximum payload of 3 Kg with a reachability of 855 mm and a workspace coverage of 94.5%.

4.1. Experimental Setup

The depth camera Realsense D415 [11] has been used to acquire the pointcloud processed by Nuitrack. An ROS C++ application is in charge of executing the Nuitrack skeleton tracking algorithm. The wearable IMU boards have been designed by integrating the IMU Bosch BNO055 with the electronic board Adafruit Feather Huzzah, which includes the Wi-Fi module ESP8266. The BNO055 chip runs an integrated filter that directly provides orientation estimations by processing data from the internal inertial sensors. Such estimations are then considered in the proposed sensor-fusion algorithm as orientation measurements $\mathbf{q}_{\text{meas}}$. The IMU covariance matrix for the Kalman filter algorithm has been experimentally identified in static conditions. The wearable devices fixed to the human bones establish a UDP socket toward a server application with synchronized timestamps. The IMU board server is interfaced also with the C++ application executing Nuitrack algorithm by means of ROS topics, so that the sensor-fusion algorithm can be processed. In particular, the application runs on a PC with an Intel® Core$^{\text{TM}}$ i5-2300 CPU @2.80 GHz × 4, 8 Gb RAM and Ubuntu 18 OS and ROS Melodic.

Therefore, two wearable IMU boards have been fixed to the left arm and the left forearm of the user. When he/she enters the robot workspace, the sensor-fusion algorithm tracks the position and the orientation of the considered bones, allowing the insertion of the bone capsules into the virtual scene. In the experiment, the virtual robot has been programmed to accomplish a linear vertical movement of the end-effector, simulating the approaching of the gripper to the object to be grasped. In the meantime, the human moves his left forearm toward the simulated robot until the minimum distance to the robot links exceeds the safety threshold of 0.2 m. Then, the robot suspends its task and the motion planner computes the repulsive potential fields to guide the robot further away from the human. Afterward, the human retracts his arm so that the robot can continue the original task.

Figure 5 shows the setup of the experimental workcell, with the Franka Emika Panda robot ready to be controlled by means of the proposed architecture.

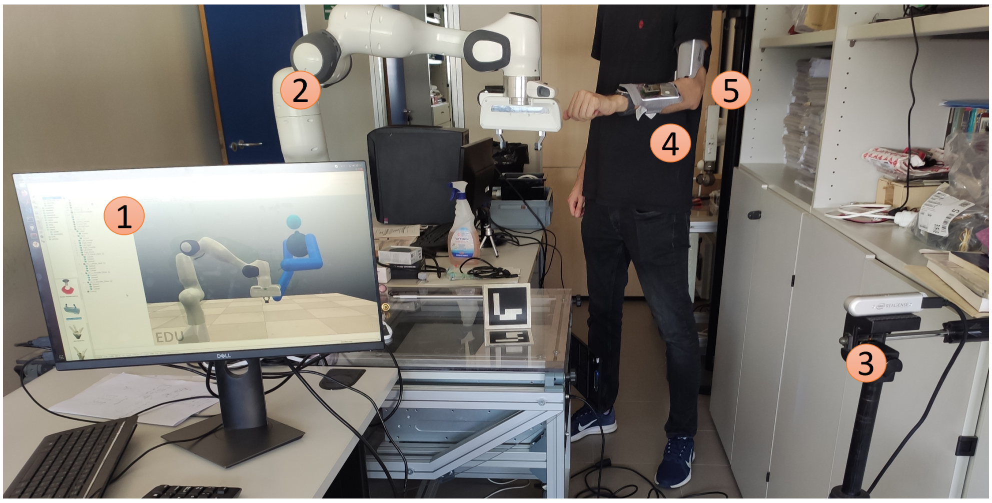


Figure 5. The experimental setup, consisting of: (1) the CoppeliaSim mixed reality scene, (2) the Panda robot, (3) the depth camera Realsense D415, (4) the left forearm wearable IMU board, and (5) the left arm wearable IMU board.

4.2. Workcell Registration

The reference systems of the robot (R), of the depth camera (C) and of the considered bones (B_1, B_2) have to be aligned to a common reference system (W) to allow the proper computation of the minimum distances in the virtual scene. Figure 6 shows the different reference systems of the devices involved in the experiments. In more detail, the common reference system W has been placed inside the robot workspace and oriented according to the absolute inertial orientation measured by the IMUs. Then, the following 4 × 4 homogeneous transformation matrices have to be computed to perform the overall registration:

- ${}^{W}\mathbf{T}_{C}$: the transformation matrix of C to W can be computed by means of the ArUco marker-based [26] procedure described in [27].
- ${}^{W}\mathbf{T}_{R}$: the transformation matrix of R to W can be found using the three-point calibration method, as described e.g., in [28].

An additional calibration procedure is required to map the Nuitrack convention for bones orientation to the absolute inertial reference system of the IMUs. Finally, the pose of the considered bones, estimated in C by the proposed algorithm (represented through ${}^{C}\mathbf{T}_{B}$), can be mapped in W by means of the following equation:

$$ {}^{W}\mathbf{T}_{B_{1,2}} = {}^{W}\mathbf{T}_{C}\,{}^{C}\mathbf{T}_{B_{1,2}}. \tag{11} $$

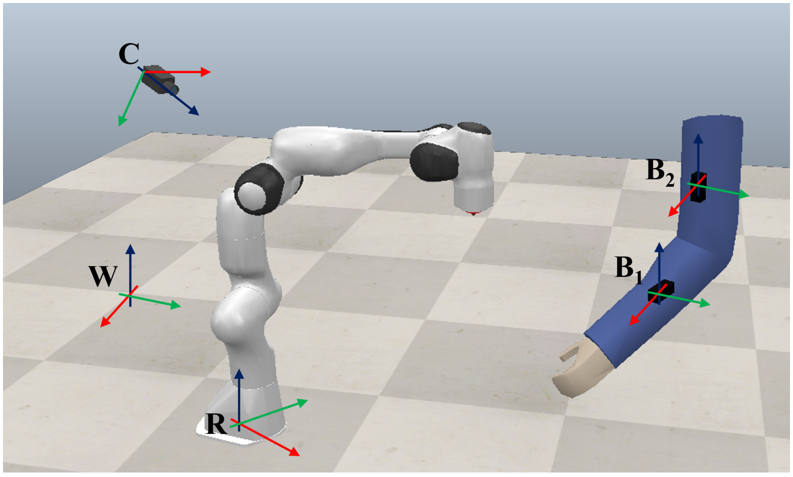


Figure 6. The reference systems of the robot (R), of the camera (C), of the bones ($B_{1,2}$) that have to be aligned to the common reference system (W).

4.3. Results

The graphs of Figure 7 show the trajectory of the robot end-effector (the first three subplots) and the evolution of the minimum distance between robot and human (the last subplot). It can be seen that when the minimum distance is shorter than the threshold $\eta_0 = 0.2$ m (highlighted by the dashed line) the end-effector stops its task (consisting of a linear movement along the z-axis) and starts the collision-avoidance motion. The effects of the repulsive fields are highlighted by the fact that even if the bone is still approaching the robot, the minimum distance is quite constant. When the human moves away, the robot resumes its original task and the end-effector motion becomes constant after a transient.

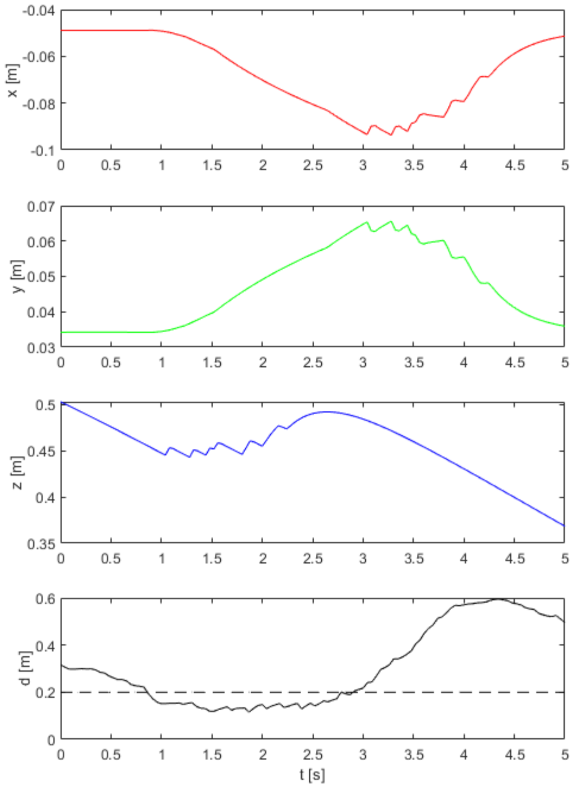
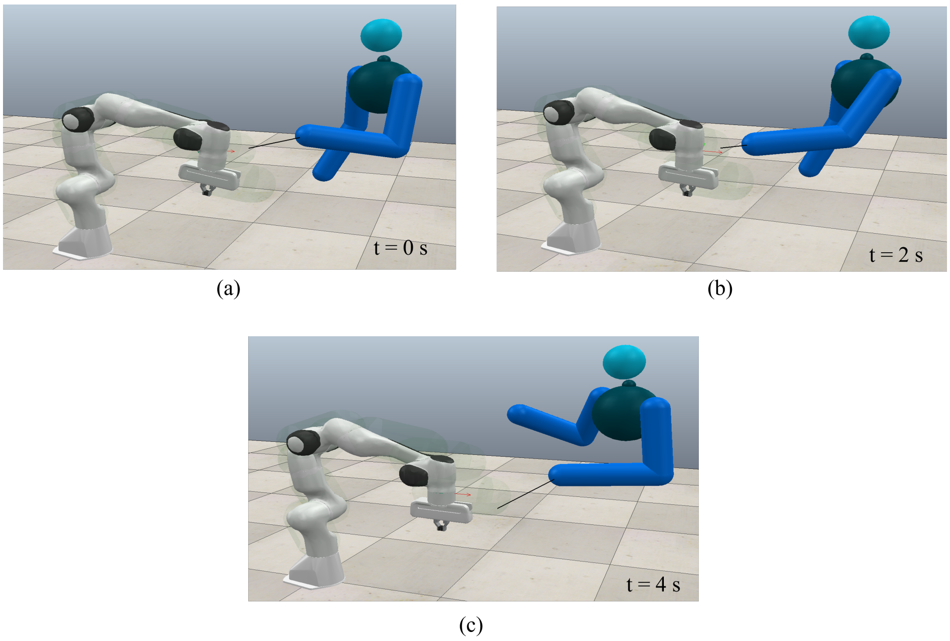


Figure 7. The end-effector position along the x, y, z axis of the workcell reference system and the minimum distance η between the robot and the human. The dashed line in the last subplot shows threshold η_0.

The screenshots depicted in Figure 8 show the human inserted into the mixed-reality scene in three phases of motion, i.e., in its starting position, when the minimum distance overcomes the threshold and finally when the human retract his forearm.



(a) (b)

(c)

Figure 8. Three screenshots taken from the mixed reality scene of CoppeliaSim. (**a**) Starting configuration. (**b**) The human forearm bone exceeds the minimum distance threshold so that the robot follows the repulsive potential fields to avoid collisions. (**c**) The human moves away and the robot resumes its original task.

5. Comparative Analyses

The following experiments have been carried out to compare the performance of the proposed methods with possible alternative approaches. First the proposed multirate EKF for the estimation of the bone orientation has been compared with two alternative formulations based on the UKF paradigm. Then, the second experiment aims at showing the advantage of using the proposed motion planning algorithm with respect to alternative planners.

5.1. Comparison among Alternative Kalman Filter Formulations

Three different versions of the multirate Kalman filter have been implemented in C++ by using the Eigen library: as an EKF by using the quaternion update formula of Equation (3); as a UKF using the same formula; and as a UKF by using the quaternion update formula of Equation (9). Tests performed on a PC with an Intel® Core™ i5-2300 CPU @2.80 GHz × 4, 8 Gb RAM and Ubuntu 18 OS, revealed that the mean execution time of the basic EKF is 149 µs, whereas the UKF using the same quaternion update formula has a mean execution time of 904 µs and the UKF using the alternative quaternion update formula has a mean execution time of 1543 µs. Moreover, the estimation accuracy of the three filters has been compared, setting the same values for $\mathbf{Q}_a$ and $\mathbf{R}_a$ matrices and the sampling time at 10 ms (smaller sampling times revealed even smaller differences). With these settings, the root-mean-square (RMS) difference between the estimated angular acceleration obtained with the basic EKF and the other filters is 1.8×10^{-4} rad/s^2 (peak difference 5.0×10^{-3} rad/s^2), for the UKF by using Equation (3), and 7.7×10^{-4} rad/s^2 (peak difference $1.5{\times}10^{-2}$ rad/s^2), for the UKF by using Equation (9). Negligible RMS differences are obtained for angular position and velocity. For these reasons, especially considering the computational effort, the UKF versions of the quaternion-based estimator will not be considered further as more efficient alternatives to the one proposed in [19].

5.2. Comparison between Different Planning Approaches

Most common approaches to collision-avoidance planning make use of probabilistic road map (PRM) and rapidly exploring random tree (RRT). Such strategies provide a collision-free motion from a starting to a goal robot configuration but they are not able to provide a prompt reaction to moving obstacles, so that the robot has to be stopped when the monitored minimum human–robot distance exceeds a certain threshold and the re-planning of the motion has to be triggered. In this work, mixed reality has been further used to better highlight the advantages of using the proposed dynamic APF-based motion planner with respect to re-planning strategies. Indeed, the monitored pose of the human bones has been inserted into a virtual scene where two overlapped virtual robots with the same original task have been driven by the proposed APF planner and by a replanning RRT-based algorithm. In particular, the replanning strategy is summarized as follows:

1. Find a collision-free robot configuration for the target end-effector pose;
2. Plan collision-free robot motions from the current configuration to the goal configuration exploiting the RRT-Connect algorithm provided by the Open Motion Planning Library (OMPL) [29] wrapper of CoppeliaSim; and
3. Execute the motion. If the robot has been moving for at least a minimum time and the minimum distance between the human and the robot exceeds the threshold:

   a Stop the robot;

   b Restart the procedure from point 1.

The minimum distance threshold has been set to 0.2 for both planners. The results depicted in Figure 9 show that the proposed planner reacts more promptly when the obstacle is approaching (i.e., from 5 s to 10 s and from 20 s to 25 s of simulation) whereas the RRT-based planner generates motions that result in strongly exceeding the threshold. This drawback is mainly due to the high computation time required to replan the motion (about 0.4 s on a PC with an Intel® Core^{TM} i5-2300 CPU @2.80 GHz × 4, 8 Gb RAM and Ubuntu 18 OS), during which the robot is stopped.

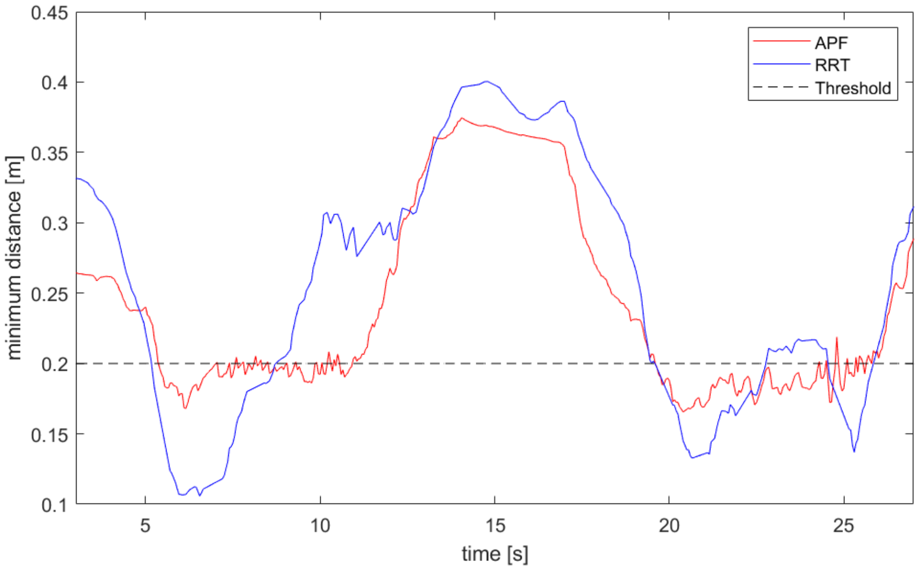


Figure 9. The minimum distance between human and robot provided by APF planner (in red) and by RRT planner (in blue).

6. Conclusions

The paper proposed the usage of mixed reality to enable a safe and effective motion planning for avoiding collisions between robots and humans that share the same workspace in a collaborative application. With the aim of improving the 3D perception of humans inside the workspace provided by pointcloud-based skeleton tracking software, a bank of

Kalman filters is used to perform a sensor-fusion algorithm to merge bone-pose estimations with data coming from a set of wearable electronics boards containing IMU sensors. The final pose estimations are exploited to insert real entities into a virtual scene of the simulator CoppeliaSim. Such a framework allows one to safely test and validate robot motions and enables the processing of a collision-avoidance motion planner whose output command can be used to control the real robot. The experiments demonstrated the feasibility of the proposed methods. Further improvements could involve extending the motion planner algorithm so as to take into account the mechanical constraints due to the physical capabilities of the robot. In the case study of seven degrees-of-freedom manipulators, the redundancy of the robot can be exploited to implement a planner, allowing the execution of multiple tasks with priority, to handle without abrupt switching the original robot task, the collision-avoidance requirements, and the fulfillment of the defined constraints on joint velocity and acceleration.

Author Contributions: Conceptualization, S.F. and M.B.; methodology, S.F. and M.B.; software, S.F., J.R., and G.N.U.; validation, S.F., J.R., and G.N.U.; formal analysis, S.F. and J.R.; investigation, S.F.; resources, M.B.; data curation, J.R. and G.N.U.; writing—original draft preparation, S.F.; writing—review and editing, S.F. and M.B; supervision, M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Villani, V.; Pini, F.; Leali, F.; Secchi, C. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* **2018**, *55*, 248–266. [CrossRef]
2. Aivaliotis, P.; Aivaliotis, S.; Gkournelos, C.; Kokkalis, K.; Michalos, G.; Makris, S. Power and force limiting on industrial robots for human–robot collaboration. *Robot. Comput.-Integr. Manuf.* **2019**, *59*, 346–360. [CrossRef]
3. Farsoni, S.; Ferraguti, F.; Bonfè, M. Safety-oriented robot payload identification using collision-free path planning and decoupling motions. *Robot. Comput.-Integr. Manuf.* **2019**, *59*, 189–200. [CrossRef]
4. Szafir, D. Mediating human–robot interactions with virtual, augmented, and mixed reality. In *Proceedings of the International Conference on Human-Computer Interaction*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 124–149.
5. Badia, S.B.i.; Silva, P.A.; Branco, D.; Pinto, A.; Carvalho, C.; Menezes, P.; Almeida, J.; Pilacinski, A. Virtual Reality for Safe Testing and Development in Collaborative Robotics: Challenges and Perspectives. *Electronics* **2022**, *11*, 1726. [CrossRef]
6. Hönig, W.; Milanes, C.; Scaria, L.; Phan, T.; Bolas, M.; Ayanian, N. Mixed reality for robotics. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 5382–5387. [CrossRef]
7. Andersen, R.S.; Madsen, O.; Moeslund, T.B.; Amor, H.B. Projecting robot intentions into human environments. In Proceedings of the 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), New York, NY, USA, 26–31 August 2016; pp. 294–301.
8. Vogel, C.; Walter, C.; Elkmann, N. A projection-based sensor system for safe physical human–robot collaboration. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 5359–5364.
9. Casalino, A.; Guzman, S.; Zanchettin, A.M.; Rocco, P. Human pose estimation in presence of occlusion using depth camera sensors, in human–robot coexistence scenarios. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–7.
10. Nguyen, M.H.; Hsiao, C.C.; Cheng, W.H.; Huang, C.C. Practical 3D human skeleton tracking based on multi-view and multi-Kinect fusion. *Multimed. Syst.* **2022**, *28*, 529–552. [CrossRef]
11. Servi, M.; Mussi, E.; Profili, A.; Furferi, R.; Volpe, Y.; Governi, L.; Buonamici, F. Metrological Characterization and Comparison of D415, D455, L515 RealSense Devices in the Close Range. *Sensors* **2021**, *21*, 7770. [CrossRef] [PubMed]
12. Longo, U.G.; De Salvatore, S.; Sassi, M.; Carnevale, A.; De Luca, G.; Denaro, V. Motion Tracking Algorithms Based on Wearable Inertial Sensor: A Focus on Shoulder. *Electronics* **2022**, *11*, 1741. [CrossRef]
13. Farsoni, S.; Bonfè, M.; Astolfi, L. A low-cost high-fidelity ultrasound simulator with the inertial tracking of the probe pose. *Control Eng. Pract.* **2017**, *59*, 183–193. [CrossRef]

14. Gultekin, H.; Dalgıç, Ö.O.; Akturk, M.S. Pure cycles in two-machine dual-gripper robotic cells. *Robot. Comput.-Integr. Manuf.* **2017**, *48*, 121–131. [CrossRef]
15. Foumani, M.; Gunawan, I.; Smith-Miles, K.; Ibrahim, M.Y. Notes on feasibility and optimality conditions of small-scale multifunction robotic cell scheduling problems with pickup restrictions. *IEEE Trans. Ind. Inform.* **2014**, *11*, 821–829. [CrossRef]
16. Sensortec, B. Intelligent 9-Axis Absolute Orientation Sensor. BNO055 Datasheet. Available online: https://www.bosch-sensortec.com (accessed on 25 June 2022).
17. Mesquita, J.; Guimarães, D.; Pereira, C.; Santos, F.; Almeida, L. Assessing the ESP8266 WiFi module for the Internet of Things. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Torino, Italy, 4–7 September 2018; Volume 1, pp. 784–791.
18. Assa, A.; Janabi-Sharifi, F. A Kalman Filter-Based Framework for Enhanced Sensor Fusion. *IEEE Sens. J.* **2015**, *15*, 3281–3292. [CrossRef]
19. Farsoni, S.; Landi, C.T.; Ferraguti, F.; Secchi, C.; Bonfè, M. Real-time identification of robot payload using a multirate quaternion-based kalman filter and recursive total least-squares. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2103–2109.
20. Wan, E.; Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium, Lake Louise, AB, Canada, 1–4 October 2000; pp. 153–158.
21. Zhao, F.; van Wachem, B. A novel Quaternion integration approach for describing the behaviour of non-spherical particles. *Acta Mech.* **2013**, *224*, 3091–3109. [CrossRef]
22. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics: Modelling, Planning and Control*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2010.
23. Rohmer, E.; Singh, S.P.N.; Freese, M. V-REP: A versatile and scalable robot simulation framework. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 1321–1326. [CrossRef]
24. Sozzi, A.; Bonfè, M.; Farsoni, S.; De Rossi, G.; Muradore, R. Dynamic motion planning for autonomous assistive surgical robots. *Electronics* **2019**, *8*, 957. [CrossRef]
25. Abdi, M.I.I.; Khan, M.U.; Güneş, A.; Mishra, D. Escaping Local Minima in Path Planning Using a Robust Bacterial Foraging Algorithm. *Appl. Sci.* **2020**, *10*, 7905. [CrossRef]
26. Garrido-Jurado, S.; Muñoz-Salinas, R.; Madrid-Cuevas, F.J.; Marín-Jiménez, M.J. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognit.* **2014**, *47*, 2280–2292. [CrossRef]
27. Ferraguti, F.; Minelli, M.; Farsoni, S.; Bazzani, S.; Bonfè, M.; Vandanjon, A.; Puliatti, S.; Bianchi, G.; Secchi, C. Augmented reality and robotic-assistance for percutaneous nephrolithotomy. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4556–4563. [CrossRef]
28. Zhang, W.; Ma, X.; Cui, L.; Chen, Q. 3 points calibration method of part coordinates for arc welding robot. In Proceedings of the International Conference on Intelligent Robotics and Applications, Wuhan, China, 15–17 October 2018; Springer: Berlin/Heidelberg, Germany, 2008; pp. 216–224.
29. Sucan, I.A.; Moll, M.; Kavraki, L.E. The open motion planning library. *IEEE Robot. Autom. Mag.* **2012**, *19*, 72–82. [CrossRef]