sensors

MDPI

Article

Human Arm Motion Prediction for Collision Avoidance in a Shared Workspace

Pu Zheng [1,*], Pierre-Brice Wieber [2], Junaid Baber [1,3] and Olivier Aycard [1]

[1] The Laboratoire d'Informatique de Grenoble, University of Grenoble Alpes, 38000 Grenoble, France; junaid.baber@univ-grenoble-alpes.fr (J.B.); olivier.aycard@univ-grenoble-alpes.fr (O.A.)
[2] Inria Centre at the University Grenoble Alpes, 38000 Grenoble, France; pierre-brice.wieber@inria.fr
[3] Department of Computer Science and Information Technology, University of Balochistan, Quetta 87300, Pakistan
* Correspondence: pu.zheng@univ-grenoble-alpes.fr

Abstract: Industry 4.0 transforms classical industrial systems into more human-centric and digitized systems. Close human–robot collaboration is becoming more frequent, which means security and efficiency issues need to be carefully considered. In this paper, we propose to equip robots with exteroceptive sensors and online motion generation so that the robot is able to perceive and predict human trajectories and react to the motion of the human in order to reduce the occurrence of the collisions. The dataset for training is generated in a real environment in which a human and a robot are sharing their workspace. An Encoder–Decoder based network is proposed to predict the human hand trajectories. A Model Predictive Control (MPC) framework is also proposed, which is able to plan a collision-free trajectory in the shared workspace based on this human motion prediction. The proposed framework is validated in a real environment that ensures collision free collaboration between humans and robots in a shared workspace.

Keywords: human robot collaboration; human motion prediction; collision avoidance

Citation: Zheng, P.; Wieber, P.-B.; Baber, J.; Aycard, O. Human Arm Motion Prediction for Collision Avoidance in a Shared Workspace. *Sensors* **2022**, *1*, 0. https://doi.org/

Academic Editors: Alwin Poulose and Antonio Fernández-Caballero

Received: 7 July 2022
Accepted: 7 August 2022
Published:

1. Introduction

The third industrial revolution brought rapid progress to industrial automation and provided the solid foundation of modern manufacturing. During this era, many enterprise companies expanded and created a number of opportunities and businesses around the world. However, the main focus of this revolution was on automation of repetitive tasks in manufacturing industry and assembly lines [1]—for example, a robot doing a fixed task with hard-coded trajectories. The fourth industrial revolution is changing the life of every individual by altering the way of living. In this revolution, human and robots are working smartly together in shared environments and the the trajectories for robots are not hard-coded—these trajectories are dynamic or predicted using machine learning.

An important component of human capacity to interact with the world resides in the ability to predict its evolution over time. Handing an object to another person, playing sports, or simply walking in a crowded street would be extremely challenging without our understanding of how people move, and our ability to predict what they are likely to do in the following instants. Similarly, machines that are able to perceive and interact with moving people, either in physical or virtual environments, must have a notion of how people move. Since human motion is the result of both physical limitations (e.g., torque exerted by muscles, gravity, moment preservation) and the intentions of subjects (how to perform an intentional motion), motion modeling is a complex task that should be ideally learned from observations.

Collaborative robots, a.k.a cobots, offer a solution for small and medium-sized companies that require a flexible, fast, and precise operational solution in a shared workspace [2]. These cobots have proven to be intrinsically safe due to their ability to detect collisions and

react accordingly [3]. However, it is always preferable to avoid collisions [4], and if a cobot is able to re-plan its trajectory to avoid collisions with a human in a shared workspace, there is an increase in productivity with respect to task completeness, effectiveness, safety, and throughput of human and cobot working time [5].

In our previous work, we proposed a model for predictive control scheme to generate trajectories for cobots that ensures a collision free environment in a shared workspace with humans by separating the work plane for the cobot [6]. The cobot automatically changes its trajectory to avoid the collision if a human hand enters the cobot plane. In this case, the collision can not be avoided, and the cobot ensures being at rest at the time of collision.

In this paper, we extend our previous work by predicting trajectories of human hands with our online trajectory generation framework [6] to ensure a safe and efficient human–robot collaboration in a shared workspace. The trajectories generated for cobots are fused with the predicted trajectories of a human hand to ensure a collision free environment. The perception module is composed of two parts: (1) detection and localization of the human hand in shared workspace, and (2) prediction of the human hand trajectory. Object detection from an RGB image is extensively studied since the development of deep learning libraries such as Openpose [7] and Mediapipe [8], which can achieve real-time skeleton detection of a human in a 2D images. However, the Cartesian 3D position of the human is necessary for the generation of the cobot trajectory. By using the RGB-D camera, we can re-project the 2D coordinates of the RGB image into Cartesian space.

This paper is organized as follows: Section 2 gives brief discussion on related work, Section 3 gives a formulation of our collision-free motion generator with a definition of collision avoidance constraints through separating planes, and a formulation through Quadratic Programs (QP). Section 4 explains the detection of the human pose from a single monocular 3D camera and the projection into Cartesian space. The prediction model is explained in Section 5. Experiments and results are discussed in Section 6; finally, conclusions and future work are discussed in Section 7.

2. Related Work

Our work mainly contributes to human robot collaboration (HRC) using human motion prediction, particularly the human hand, in a shared workspace with a cobot. There are multiple ways and practices to detect the human for tracking and prediction in HRC systems [9–12]. Use of monocular camera, depth camera, 3D LiDAR, and inertial measurement unit (IMU) sensors for motion capture is a common practice. The IMU sensor based methods are not suitable for collaborative tasks, and the 3D LiDARs are relatively expensive in terms of cost and computation. However, using the vision sensors such as RGB cameras is a widely used approach for human motion prediction [9]. The main limitation of RGB cameras includes a range of camera sensors and enabling 3D human position w.r.t the robot. Using an RGB-D camera, which is a combination of a single view from monocular and depth sensors, provides a balance between simplicity and performance [13].

The depth camera based on time-of-flight (TOF) and structured light technologies can provide the distance information from a single depth image; thus, it creates the possibilities to deal directly with 3D data [14]. Moreover, human pose estimation has some unique characteristics and challenges such as flexible body configuration indicating complex interdependent joints, diverse body appearance, and complex environment may cause occlusion. The existing approaches can be divided into three categories: template-based method, feature-based method, and learning-based method [15]. The template-based method compares the similarity between the detected object and the constructed template to identify the motion category [16,17]. The template-based methods need to establish a template library of parameterised template to compare with the human body, which is time-consuming, and the accuracy of template-based methods is very limited due to the diversity of the different human pose in space. Feature-based methods use geodesic distance information [18], geometric features such as silhouette [19], to estimate the human joints. The feature-based template has some disadvantages; for instance, it requires prior

knowledge to combine with extracted global or local features to obtain the 3D pose, and it is not suitable for changing poses.The learning-based method use the network structure to automatically learn the required features from input data. The learned features can be further used to extract the human poses [20].

Recently, deep learning based models are getting popular [11,21–24]. Deep learning methods can take unprocessed data such as point clouds and produce human poses with high accuracy [24]. However, deep learning models are data greedy models and require heavy data for training which are not easily available for human hand prediction, and datasets contain point clouds that are not suitable for real-time applications. Therefore, as a more accessible approach, estimating human poses from RGB images captured by regular cameras and then mapping the 2D information into 3D space is efficient and widely practiced in industry and academia [6,9].

The trivial way to estimate 3D human poses is to design an end-to-end network to predict the 3D key-point locations directly from 2D images [25,26]. However, recovering a 3D human pose from a single image is still challenging and leads to many false positives. Using depth information from RGB-D cameras can effectively transform 2D pose location into 3D [6].

Existing human motion prediction methods can generally be divided into model-based and learning-based approaches. Model-based approaches attempt to directly model the kinematics or dynamics of the human and thus find the corresponding arm motor control [27], and human movements follow an optimal feedback control strategy that connects together motor behaviour, limb mechanics, and neural control. The detailed description on human arm motor control can be found in [27]. However, the choice of optimal trajectory cost is not trivial because the human musculoskeletal system presents more degrees of freedom (DoF). This kinematic, dynamic, and actuation redundancy issue is not straightforward in terms of motion equations. Numerous cost functions have been identified in literature [28–31]. In [28], the authors model the hand's point-to-point kinematic motions with minimum Cartesian jerks (third derivative of Cartesian coordinates) for an arm movement in the horizontal plane. Authors in [29] incorporate dynamics with a minimum torque change model in the horizontal plane; however, the results are not validated for 3D movements. While defining these motor control criteria manually is difficult, the author in [30] defines a combination of seven different criteria (such as Cartesian jerk, angle jerk, angle acceleration, torque change, torque, geodesic and energy) and an inverse optimisation method has been used to find the weight associated with each criteria. In [31], instead of finding arm motor control artificially, the authors over-approximate the occupancy of the arm with a maximum velocity model, but this can be too restrictive if the prediction horizon is long. These approaches have several limitations such as the dynamics of the human being highly nonlinear and non-deterministic; it can vary according to emotions and physical condition, so direct modelling can be quite inaccurate in different situations. Moreover, the hypothesis about the human's rationality is often invalid; hence, constructing an optimisation criteria based on this hypothesis can be very ambiguous, and the combination of the different criteria is chosen manually.

Human motion is the result of complex bio-mechanical processes that are challenging to model. As a consequence, state-of-the-art work on motion prediction focuses on data-driven models [9,32–37] such as probabilistic models [35,36] and deep learning models [9,37]. Recent work on short-term human motion prediction has centered on a Recurrent Neural Network (RNN) due to their capacities to handle sequential data. The RNN can remember important things about the input received that allow them to be very accurate in predicting the output. In the proposed framework, an RNN based approach is used to predict the human hand motion which is then passed to our motion planning package for collision avoidance with the cobot.

3. General Trajectory Planning Scheme

The goal of our scheme is to generate a collision-free trajectory for a cobot (e.g., a 7-DoF manipulator cobot) that has to perform a task in a workspace shared with a human worker. This scheme is composed of three parts, as shown in Figure 1: (i) a human motion prediction module, (ii) a collision-free trajectory generation module, and (iii) a low-level robot motion control module.

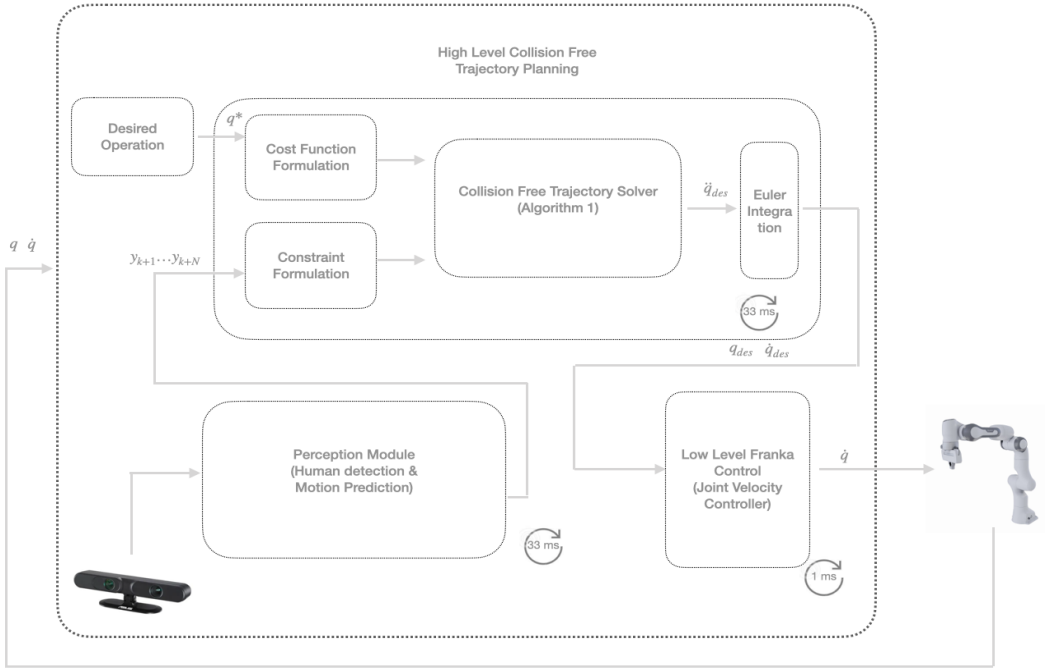


Figure 1. Abstract flow diagram of control architecture.

In this section, we summarise how to compute a collision-free trajectory based on the perception module and the cobot's dynamics. In addition, we emphasise the role of the terminal constraint to guarantee safety. This terminal constraint provides a *passive motion safety* guarantee [38], which means that, if a collision occurs, the cobot is at rest at the time of the collision so that it does not inject its own kinetic energy. In the following, we recall the main equations from the MPC approach developed in our previous work [6].

3.1. Separating Plane Optimisation

As illustrated in Figure 2, if there exists at the prediction time $k \in \mathbb{N}$ a plane defined by a normal vector $a_k \in \mathbb{R}^3$ and a scalar constant $b_k \in \mathbb{R}$ such that all vertices y^j related to the human stay on one side between instants k and $k+1$ while all vertices r^i related to the cobot stay on the other side, then we have evidence that they do not collide over this interval of time. Here, $j \in \{1, \dots N_p\}$ and $i \in \{1, \dots N_r\}$ where N_p and N_r are the number of vertices associated with the human and the cobot, as appears in the constraints, Equations (1b)–(1e), and the distance $d \in \mathbb{R}$ controls the position of the separating plane between the human and the cobot.

$$\min_{a_k, b_k, d} \; -d + \alpha d^2 + \beta \|a_k - a_k^p\|^2 + \beta ||b_k - b_k^p||^2 \tag{1a}$$

$$\text{s.t. } \forall j \in \{1, \dots N_p\},\ a_k^T y_k^j \leq b_k, \tag{1b}$$

$$\forall j \in \{1, \dots N_p\},\ a_k^T y_{k+1}^j \leq b_k, \tag{1c}$$

$$\forall i \in \{1, \dots N_r\},\ a_k^T r_k^i \geq b_k + d, \tag{1d}$$

$$\forall i \in \{1, \dots N_r\},\ a_k^T r_{k+1}^i \geq b_k + d, \tag{1e}$$

$$-1_3 \leq a_k \leq 1_3, \tag{1f}$$

$$1 - \varepsilon \leq a_k^T\, a_k^p \leq 1 \tag{1g}$$

We want to maximise the distance d between the separating plane and the cobot. Given the formulation as a minimisation problem, we include the term $-d$ in the cost function, Equation (1a). The following term in the cost function smooths the variations of separating planes, with a_k^p and b_k^p the separating plane parameters obtained at the previous sampling time, and some small weights α and β. Finally, we use constraints, Equations (1f)–(1g), to approximate a nonlinear constraint to bound the vector a_k to a unit norm, where $1_3 \in \mathbb{R}^3$ is a row vector of ones.

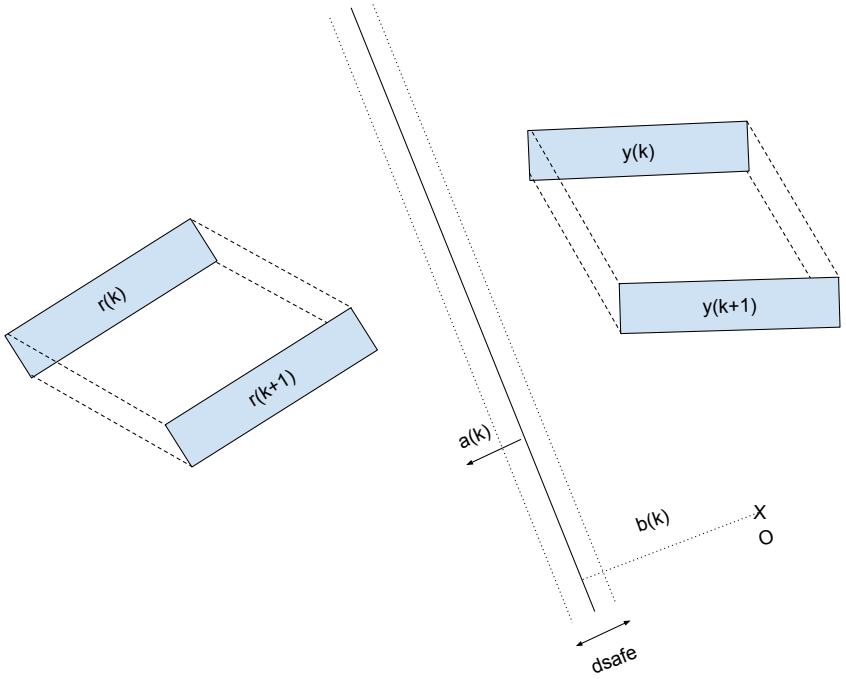


Figure 2. An illustration of separating plane between two objects.

3.2. Optimal Motion Generation

Once we have a sequence of separating planes parameters, we can include them in our MPC scheme to compute an optimal collision-free trajectory:

$$\min_{\mathbf{u}} \sum_{k=0}^{N-1} \|s_{k+1} - s_{k+1}^{des}\|_Q^2 + \|u_k - u_k^{des}\|_R^2 \tag{2a}$$

$$\text{s.t. } \forall k \in \{0, \dots N-1\},\ \underline{u} \leq u_k \leq \overline{u}, \tag{2b}$$

$$\forall k \in \{1, \dots N\},\ \underline{q} \leq q_k \leq \overline{q}, \tag{2c}$$

$$\forall k \in \{1, \dots N-1\},\ \underline{\dot{q}} \leq \dot{q}_k \leq \overline{\dot{q}}, \tag{2d}$$

$$\dot{q}_N = 0, \tag{2e}$$

$$\forall k \in \{0, \dots N-1\},\ \forall i,$$
$$a_k^T r_k^i(q_k^p) + a_k^T J(q_k^p)(q_k - q_k^p) \geq b_k + d_{safe}, \tag{2f}$$

$$\forall k \in \{0, \dots N-1\},\ \forall i,$$
$$a_k^T r_k^i(q_{k+1}^p) + a_k^T J(q_{k+1}^p)(q_{k+1} - q_{k+1}^p) \geq b_k + d_{safe} \tag{2g}$$

where $q_k \in \mathbb{R}^n$ and $\dot{q}_k \in \mathbb{R}^n$ are respectively the joint position and velocity, with n the number of degrees of freedom. The state $s_k \in \mathbb{R}^{2n}$ includes q_k and $\dot{q}_k$, and $u_k \in \mathbb{R}^n$ is the control input (acceleration) of the cobot.

Algorithm 1 Collision free trajectory computation

Input: U_k^p, S_k, a_k^p, b_k^p
Output: U_k

```
i = 0;
while (||U_k − U_k^p||^2 OR i ≤ k) do
    U_k^p = U_k;
    /* Updating Robot Parameters */
    {a,b} ⟵ Solve Equation (1) for k ∈ {0, ... N − 1} ;
    {U_k} ⟵ Solve Equation (2) ;
    i ++;
```

Our prediction horizon has a length $N \in \mathbb{N}$. The cost function, Equation (2a), is designed to track a desired joint state trajectory q_k^{des} with acceleration u_k^{des}, while $\underline{q}$, $\overline{q}$, $\underline{\dot{q}}$, $\overline{\dot{q}}$, $\underline{u}$, $\overline{u}$ indicate minimum and maximum joint positions, speed, and acceleration (we assume that $\underline{\dot{q}} \leq 0 \leq \overline{\dot{q}}$ and $\underline{u} \leq 0 \leq \overline{u}$). The terminal constraint, Equation (2e), ensures that the cobot is at rest at the end of the prediction horizon in order to provide a passive motion safety guarantee, making sure that the cobot is able to stop and stay at rest before any collision happens in the future. Equations (2f)–(2g) introduce the collision avoidance constraint based on separating planes, which is computed by linearising the kinematics of the cobot around the previously computed trajectory:

$$r_k^i = r^i(q_k) \approx r^i(q_k^p) + J(q_k^p)(q_k - q_k^p) \tag{3}$$

4. Human Pose Detection

In the algorithm presented above, a sequence of future human positions has to be known. In this section, we explain how to detect a human's position and train the model for prediction.

The perception system used in this work is an ASUS Xtion Pro depth camera which provides cloud points, colour and depth images. We can then recover the (x,y,z) position in Cartesian space by combining color and depth image information. The advantage of working with 2D images is that we can directly use existing deep learning libraries such as OpenPose [39] or MediaPipe [8], which are fast and robust. Mediapipe is widely used for

several real-time applications such as tracking [40], and sign language understanding [41]. An example of human upper-body key points extraction is shown in Figure 3a. With the lightweight version of this deep learning model, the inference speed is performing at 0.25 s on a MacBook Pro (2017).

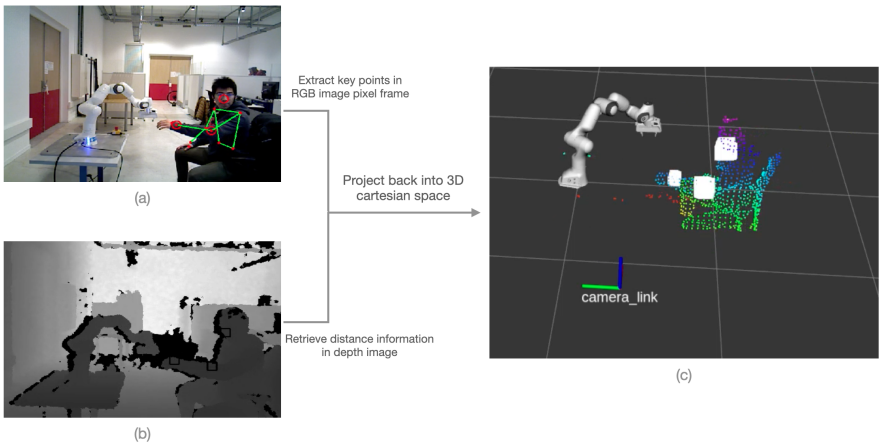


Figure 3. Example demonstration of RGB-D image for mapping human hand in 3D space, (**a**) shows RGB image on which hand joins are detected, (**b**) shows the bounding box around the points in the depth image, and (**c**) shows the mapping of points in 3D space.

After obtaining the coordinates of the joints in the colour image frame, we can map the corresponding coordinates in the depth image to find the distance between the camera and the pixel points. This mapping necessitates a proper calibration of the camera [42]. The distance information allows us to compute the key point's Cartesian location using the pinhole camera projection model, since we know the camera's intrinsic parameters:

$$X = (u - c_x)\frac{Z}{f_x} \tag{4}$$

$$Y = (v - c_y)\frac{Z}{f_y} \tag{5}$$

$$Z = Z \tag{6}$$

where u and v are key point locations in pixel coordinates, c_x and c_y are camera offset, f_x and f_y are camera focal parameters, and Z is the distance given by the depth image.

The information contained in the depth image is sensitive to disturbances and background elements. To make the result more robust and accurate, we define a bounding box around the key point to eliminate outliers and average the distances, as shown in Figure 3b. Finally, we successfully map the key point's from RGB image to 3D location, as shown in Figure 3c.

5. Human Hand Motion Prediction

Figure 4 shows the overall architecture of human hand prediction, robot controlling, and collision avoidance. It has basically three modules; in the first module, it extracts the human hand trajectory from the RGB-D camera and passes to our trained prediction model: in the second module, the predicted trajectories are handled by a motion planning package to detect and avoid collision: finally, the last module controls the motion of the robot based on the feedback by a motion planning module. The configuration for PC used is also shown in Figure 4; the average time to extract hand trajectory and prediction is 0.04 seconds, whereas the frame per seconds (fps) from RGB-D camera is 33, which makes our prediction real-time on commodity hardware. The 0.04 seconds comprise time to extract CNN based keypoints for hand motion trajectory generation and LSTM based prediction. The motion planning package takes 0.01 seconds/frame to ensure collision avoidance.

The simulation demo of human and cobot collaboration on shared workspace can be seen online (https://www.youtube.com/watch?v=PAZZRtS7Qc4, accessed on 17-08-2022).

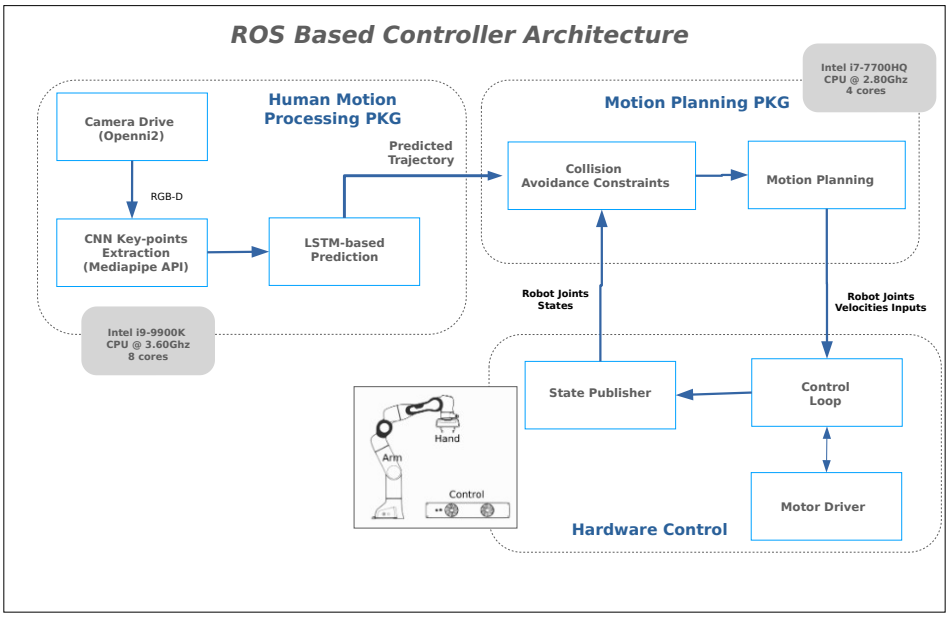


Figure 4. ROS based controller architecture to enable collaboration between humans and robots in shared environments.

The dynamics of the human can be described in state-space from Equations (7a) and (7b). Without loss of generality, we consider only the dynamics of a human's hand position to simplify notations:

$$x_{t+1} = g(x_t, w_t) \tag{7a}$$
$$y_t = h(x_t) \tag{7b}$$

where $x_t \in \mathbb{R}^3$ is the discrete time variable describing the human's hand position, $w_t \in \mathbb{R}^3$ is the muscular force or external effect that causes the human's movement, which is not known; the function g represents the human's hand dynamics, and $h_t \in \mathbb{R}^3$ is the measurable position given a state x_t. We assume that the movement of the human is not completely random and follows patterns as shown in Figure 5b, where dotted red lines denote representative motions to different goals.

In order to anticipate the human's future motion, it's not sufficient to predict only one-step ahead as shown in Equation (7a). In a more general scenario, we want to predict T steps ahead given a current state x_t and w_t and consider L-order Markov assumptions. Therefore, we can formulate this problem as: given a time-series input $\mathbf{x} = \{x_t, x_{t-1}, \ldots, x_{t-L}\}$, we want to find a function ϕ such that: $\phi : \mathbf{x} \rightarrow \mathbf{y}$, where L is the number of past observations and $\mathbf{y} = \{x_{t+1}, x_{t+2}, \ldots, x_{t+T}\}$ with T the number of steps to predict.

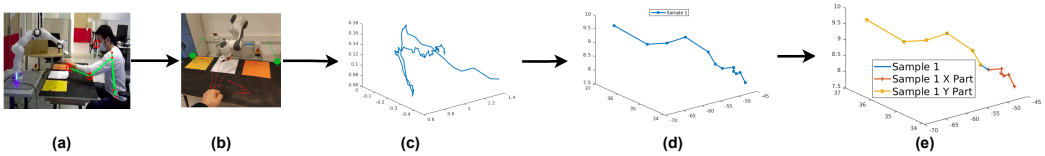


Figure 5. Demonstration of the environment for dataset generation, (**a**) shows the person working in shared environment with a cobot, (**b**) shows the possible goals on which the hand should be moving, (**c**) shows one sample hand motion trajectory generated over one minute, (**d**) shows the sub-trajectory over 12 observations, and (**e**) shows the trajectory divided into two sequences (x for training and y for prediction).

Modelling such a dynamics function is very challenging because the external factors are not measurable and unpredictable. Moreover, the dynamics of the human are highly nonlinear. However, neural network structure is efficient to learn such nonlinear mapping patterns. We define our prediction network structure in Figure 6a as an encoder–decoder model. The past observation data are encoded through several stacked Long Short-Term Memory (LSTM) layers to increase the depth of the network. The encoded information is passed into an LSTM decoder layer followed by a fully connected layer to produce the final multi-step prediction. The structure of an LSTM cell is shown in Figure 6b and the mathematical formulation is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{8a}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{8b}$$
$$\tilde{C}_t = tanh(W_C \cdot [h_t, x_t] + b_C) \tag{8c}$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{8d}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t + b_o] \tag{8e}$$
$$h_t = o_t * tanh(Ct) \tag{8f}$$

where $[W_f, b_f]$, $[W_i, b_i]$, $[W_C, b_C]$ and $[W_o, b_o]$ are learn-able weights and bias, f_t and i_t are forget gate and update gate, and h_{t-1} and h_t are previous and current hidden states, respectively. $\tilde{C}_t$ is the new candidate cell value. Thus, the new cell state C_t is updated by C_{t-1} and $\tilde{C}_t$ with associated forgetting weight and update weight. The new output and new hidden state are represented by o_t and h_t. In the end, the predicted positions are provided to the separating plane computation to formulate constraints Equations (1b) and (1c).

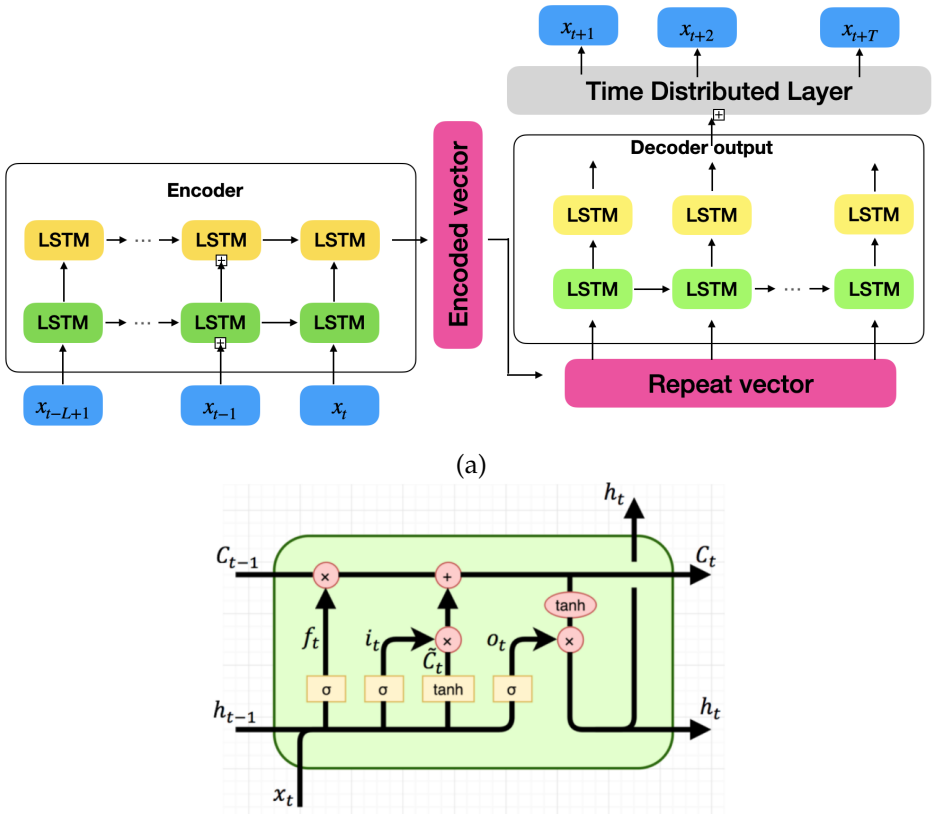


Figure 6. Proposed model for human hand motion prediction, (**a**) shows the architecture of the encoder–decoder LSTM neural network, and (**b**) shows the overview of LSTM cell architecture.

6. Experiments and Results

As stated in the Introduction, the collision avoidance by predicting human motion prediction has potential applications in the industry. However, there is still no benchmark dataset for experimentation and learning. Therefore, very few works have been reported. In this paper, we generate a dataset with real cobot coordination. The Franka Emika Panda cobot is used for the experimentation. In our previous work, we used a similar cobot for collision detection [6]. To generate the human hand trajectories, a human is asked to perform some tasks on a shared workspace with the cobot. The human hand moves to several different goals with some patterns with different speed and position; the patterns on which the hand should be moved is shown in Figure 5b, and the overall flow diagram for dataset generation is shown in Figure 5.

Let the learning data of K observation sequences be collected from the RGB-D camera, as demonstrated in Figure 5, S = $\{S^1, \ldots, S^K\}$, where $S^k = \{S^k_1, \ldots, S^k_{T_k}\}$. Each element $S^k_{t_k}$ denotes the position of the hand in Cartesian space. In this experimentation, the human's hand moves to several different goals with some patterns shown in Figure 5b. For each task, we generate five similar trajectories with minor changes in position and speed. Raw trajectory data are shown in Figure 5 (a partial trajectory), which can be used with relative positions for better learning [43].

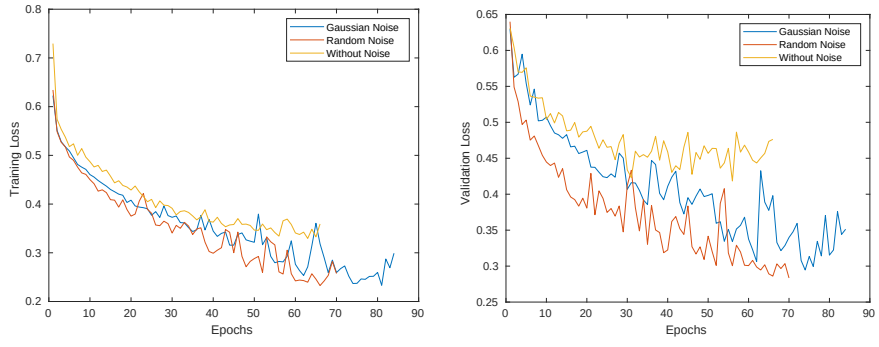
We transform the absolute coordinates of the hand's position into relative coordinates (relative displacements) to let coordinates become scene independent. In this way, the model will learn the motion displacement pattern instead of memorizing the trajectory. Secondly, we apply the data augmentation method to increase model generalisation capability. We use random rotation to each trajectory to make the network learn rotation-invariant patterns. We add Gaussian noise with mean 0 and small standard deviation to every point to make the network more robust to small perturbations and imprecision. Furthermore, we divide these trajectories into prediction windows. For example, we define one training

sample as $\chi = (x, y)$ with x and y tensor of shape two corresponding to time-step and features size, as shown in Figure 5e.

We demonstrate the proposed optimal collision-free trajectory planner with a 7-DoF manipulator cobot. Maximum joint speed and acceleration are respectively $\frac{\pi}{2}$ rad.s^{-1} and 10 rad.s^{-2}. We opt for a prediction horizon of length 0.25 s, with sampling time Δt = 0.05 s and N = 5, which covers the time necessary for the cobot to stop completely under all circumstances, in order to satisfy the terminal constraint, Equation (2e), and enable in this way the passive motion safety guarantee. A longer prediction time could provide improved collision avoidance, but this would be highly dependent on the precision of longer-term human motion prediction.

The safety distance is chosen equal to $d_{safe} = 20$ cm. The cobot completes a pick-and-place task between positions $G_{rA} = (0.5, 0.4, 0.2)$ m and $G_{rB} = (0.5, -0.4, 0.2)$ m expressed in the frame of the cobot base link. These two positions are shown as the green balls in Figure 5b. In addition, the human moves his hand following the trajectory patterns shown in Figure 5b. The neural network model is implemented in Tensorflow [44], and the total network parameters are 149,699. The encoder part consists of three stacked LSTM layers with 64 units for each layer. We add l_1 and l_2 regularisation with weights 1×10^{-4} and 1×10^{-5}, respectively. The decoder has the same structure as the encoder, and we change the final output layer by a time distributed by using fully connected layers to predict future relative displacements. The MSE (mean squared error) is used as loss function for our deep learning model, and the original data are augmented by adding random rotations and Gaussian noise, as explained above. The Gaussian noise is widely used for data augmentation. However, we experimented with Gaussian noise, random noise, and without any kind of noise. Figure 7 shows the learning loss over different epochs, (a) shows training loss, and (b) shows validation loss.

The model is evaluated based on MAE (mean absolute error), which is a widely used metric for evaluating the prediction based models [45–48]. The MAE is basically the sum of absolute difference between ground-truth 3D position and predicted position. Figure 8 shows the MAE for the proposed model; (a) shows the MAE for validation and training set with Gaussian noise; (b) and (c) show comparative curves for random noise and models without noise. It can be seen that the model without noise has a high error on the validation set, which clearly gives the impression of over-fitting. Surprisingly, the random noise gives an overall minimum error on the training and validation set, but the Gaussian based model works better on a cobot when these models were deployed for real-time testing.



(a) (b)

Figure 7. Training and validation loss of proposed model, (**a**) shows the training and validation MAE for the model with Gaussian noise, and (**b**) shows the training and validation MAE with random noise.

The qualitative visualization of deployed model is shown in Figure 9. In Figure 9a,a', the cobots move from goal G_{rA} to G_{rB}, and the collision-free trajectory is shown as successive frames in green. As the distance between the cobot and human is large enough, this trajectory is straight to the goal. The yellow spheres represent the predicted positions

of the human's hand in five time-steps. The green plane represents the separating plane (only the first predicted step is shown here, we have in total (N-1) planes). In Figure 9b,c and Figure 9b′,c′, the cobot deviates its trajectory in order to avoid the human motion. The predicted positions are shown according to the yellow sphere. Finally, the cobot attends the position of G_{rB} with successful collision avoidance, as shown in Figure 9d,d′.

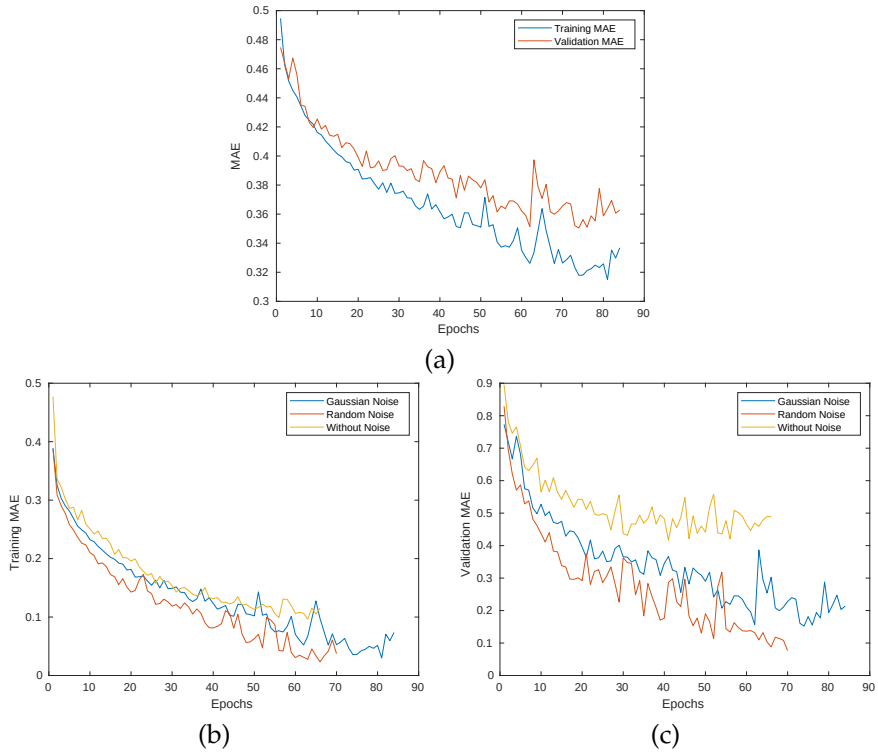


Figure 8. The MAE of proposed model, (**a**) shows the training and validation MAE for the model with Gaussian noise, (**b**,**c**) show the comparative validation and training MAE with a different configuration of noise.

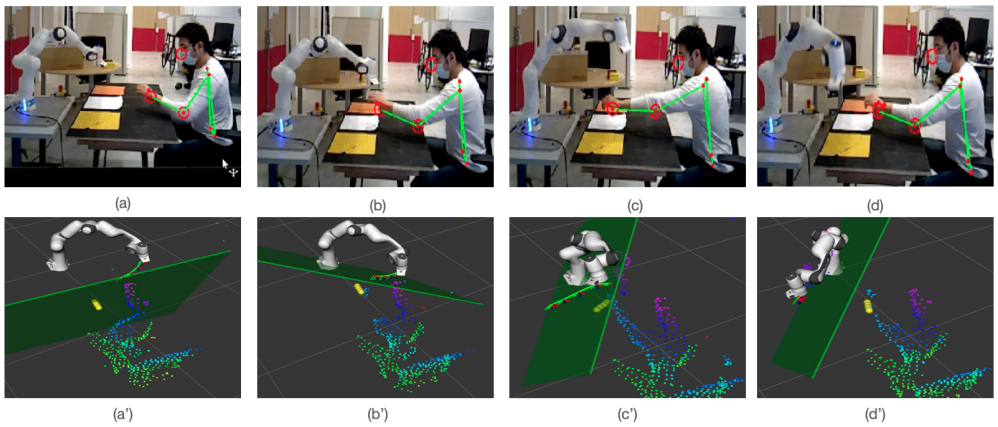


Figure 9. The qualitative visualization of collaborative environment between cobot and human, (**a**) shows the distance between cobot and hand, which is much larger than safety distance (20 cm), (**b**) shows the behavior of the cobot when the human hand is intentionally placed for possible collision, the cobot deviates from its initial trajectory to avoid the collision, (**c-d**) show that the cobot achieves its goal without stopping or hurting the human while keeping a safe distance. The sub-figures from (**a′**–**d′**) show the corresponding visualization of the same data in RVIZ.

To ensure human safety in a collaborative shared workspace, we evaluated different cases on which we ensured that the cobot achieves his goal by avoiding the possible collision

with humans, as explained in previous experiments, and we also tested a situation which created a deadlock, and it is not achievable for the cobot to complete the task. Figure 10 shows the simulation on which the human hand is placed for longer duration, which creates no exception for completing the task. In this case, the motion generator keeps the cobot still at the desired safe distance.

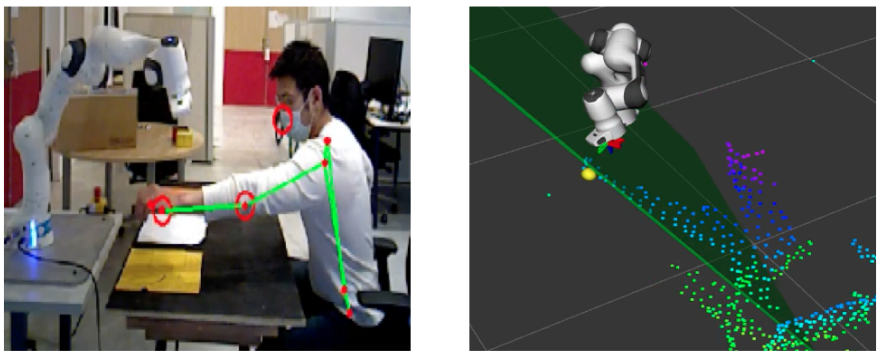


Figure 10. A case study on which collision can not be avoided, (**a**) human blocks cobot motion intentionally, (**b**) the trajectory generator ensures that the cobot is at rest to avoid collision.

The proposed MPC modules give competitive performance for various industrial tasks in shared environments. However, the proposed model ensures human safety and collision avoidance only if one hand is used. It is obvious that the perception module can not see the second hand of the human due to occlusion. The limitation of the proposed module can easily be removed by installing more than one RGB-D cameras to ensure the occlusion free perception.

7. Conclusions and Future Work

In this paper, we integrated a perception module into our previous safe MPC scheme to generate optimal collision-free trajectories online. In our previous MPC module, we detected the collision and ensured that, during collision, the cobot is at rest. In this work, we extended our previous MPC scheme; instead of detecting collision, we aimed to prevent it by predicting the human hand motion. Based on the prediction trajectories, the cobot changes its motion and maintains a safe distance from the human hand. Taking into account the future motion of a human's hand can significantly help the motion generator to plan a collision-free trajectory. In this case, the task of the cobot is interrupted intentionally by the human, and MPC can not generate a collision free trajectory; then, the motion generator lets the cobot wait at a safe distance. However, using one camera leads to a problem of occlusion. In this case, the human second hand position can not be detected reliably even with state-of-the art algorithms. Our next goal is to extend our perception module with multiple cameras to ensure occlusion-free perception to our MPC. Furthermore, we want to generalize the hand's prediction task to whole-body motion prediction where humans can work with autonomous mobile robots.

Author Contributions: Conceptualization, P.Z., O.A., and P.W.; methodology, P.Z., P.W., and O.A; software, P.Z.; validation, P.Z., P.Z., O.A., and J.B.; formal analysis, P.Z., P.W., O.A., and J.B.; investigation, P.Z., P.W., O.A., and J.B.; resources, O.A.; data curation, P.Z.; writing—original draft preparation, P.Z., P.W., O.A., and J.B.; writing—review and editing, P.Z., P.W., O.A., and J.B.; visualization, P.Z., and J.B.; supervision, O.A., and P.W.; project administration, P.W.; funding acquisition, O.A. All authors have read and agreed to the published version of the manuscript.

Funding: XXX

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable

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

References

1. Mukherjee, D.; Gupta, K.; Chang, L.H.; Najjaran, H. A Survey of Robot Learning Strategies for Human-Robot Collaboration in Industrial Settings. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102231.
2. Romero, D.; Stahre, J.; Wuest, T.; Noran, O.; Bernus, P.; Fast-Berglund, Å.; Gorecky, D. Towards an operator 4.0 typology: A human-centric perspective on the fourth industrial revolution technologies. In Proceedings of the International Conference on Computers and Industrial Engineering (CIE46), Tianjin, China, 29–31 October 2016; pp. 29–31.
3. Loughlin, C.; Albu-Schäffer, A.; Haddadin, S.; Ott, C.; Stemmer, A.; Wimböck, T.; Hirzinger, G. The DLR lightweight robot: Design and control concepts for robots in human environments. *Ind. Robot Int. J.* **2007**, *34*, 376–385.
4. De Luca, A.; Flacco, F. Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration. In Proceedings of the 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), Rome, Italy, 24–27 June 2012; pp. 288–295.
5. Lasota, P.A.; Shah, J.A. Analyzing the effects of human-aware motion planning on close-proximity human–Robot collaboration. *Hum. Factors* **2015**, *57*, 21–33.
6. Zheng, P.; Wieber, P.B.; Aycard, O. Online optimal motion generation with guaranteed safety in shared workspace. In Proceedings of the ICRA, Paris, France, 31 May–31 August 2020.
7. Osokin, D. Real-time 2d multi-person pose estimation on cpu: Lightweight openpose. *arXiv* **2018**, arXiv:1811.12004.
8. Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E.; Hays, M.; Zhang, F.; Chang, C.L.; Yong, M.G.; Lee, J.; et al. Mediapipe: A framework for building perception pipelines. *arXiv* **2019**, arXiv:1906.08172.
9. Wang, Y.; Ye, X.; Yang, Y.; Zhang, W. Collision-free trajectory planning in human–robot interaction through hand movement prediction from vision. In Proceedings of the 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), Birmingham, UK, 15–17 November 2017; pp. 305–310.
10. Psarakis, L.; Nathanael, D.; Marmaras, N. Fostering short-term human anticipatory behavior in human–robot collaboration. *Int. J. Ind. Ergon.* **2022**, *87*, 103241.
11. Liu, Q.; Li, M.; Yin, C.; Qian, G.; Meng, W.; Ai, Q.; Hu, J. CNN-Based Hand Grasping Prediction and Control via Postural Synergy Basis Extraction. *Sensors* **2022**, *22*, 831.
12. Widmann, D.; Karayiannidis, Y. Human Motion Prediction in Human-Robot Handovers based on Dynamic Movement Primitives. In Proceedings of the 2018 European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018; pp. 2781–2787.
13. Wang, J.; Tan, S.; Zhen, X.; Xu, S.; Zheng, F.; He, Z.; Shao, L. Deep 3D human pose estimation: A review. *Comput. Vis. Image Underst.* **2021**, *210*, 103225.
14. Sarbolandi, H.; Lefloch, D.; Kolb, A. Kinect range sensing: Structured-light versus Time-of-Flight Kinect. *Comput. Vis. Image Underst.* **2015**, *139*, 1–20.
15. Xu, T.; An, D.; Jia, Y.; Yue, Y. A review: Point cloud-based 3d human joints estimation. *Sensors* **2021**, *21*, 1684.
16. Zhu, Y.; Dariush, B.; Fujimura, K. Kinematic self retargeting: A framework for human pose estimation. *Comput. Vis. Image Underst.* **2010**, *114*, 1362–1375.
17. Ye, M.; Yang, R. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014; pp. 2345–2352.
18. Yuan, X.; Kong, L.; Feng, D.; Wei, Z. Automatic feature point detection and tracking of human actions in time-of-flight videos. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 677–685.
19. Xu, T.; An, D.; Wang, Z.; Jiang, S.; Meng, C.; Zhang, Y.; Wang, Q.; Pan, Z.; Yue, Y. 3D Joints Estimation of the Human Body in Single-Frame Point Cloud. *IEEE Access* **2020**, *8*, 178900–178908.
20. Shotton, J.; Girshick, R.; Fitzgibbon, A.; Sharp, T.; Cook, M.; Finocchio, M.; Moore, R.; Kohli, P.; Criminisi, A.; Kipman, A.; et al. Efficient human pose estimation from single depth images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 2821–2840.
21. Zhou, Y.; Dong, H.; El Saddik, A. Learning to estimate 3d human pose from point cloud. *IEEE Sensors J.* **2020**, *20*, 12334–12342.
22. Shahtalebi, S.; Atashzar, S.F.; Patel, R.V.; Mohammadi, A. HMFP-DBRNN: Real-Time Hand Motion Filtering and Prediction via Deep Bidirectional RNN. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1061–1068.
23. Haque, A.; Peng, B.; Luo, Z.; Alahi, A.; Yeung, S.; Fei-Fei, L. Towards viewpoint invariant 3d human pose estimation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 160–177.
24. Wang, K.; Xie, J.; Zhang, G.; Liu, L.; Yang, J. Sequential 3D human pose and shape estimation from point clouds. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7275–7284.
25. Li, S.; Chan, A.B. 3d human pose estimation from monocular images with deep convolutional neural network. In Proceedings of the Asian Conference on Computer Vision, Singapore, 1–5 November 2014; pp. 332–347.
26. Martinez, J.; Hossain, R.; Romero, J.; Little, J.J. A simple yet effective baseline for 3d human pose estimation. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2640–2649.

27. Scott, S.H. Optimal feedback control and the neural basis of volitional motor control. *Nat. Rev. Neurosci.* **2004**, *5*, 532–545.
28. Flash, T.; Hogan, N. The coordination of arm movements: An experimentally confirmed mathematical model. *J. Neurosci.* **1985**, *5*, 1688–1703.
29. Uno, Y.; Kawato, M.; Suzuki, R. Formation and control of optimal trajectory in human multijoint arm movement. *Biol. Cybern.* **1989**, *61*, 89–101.
30. Sylla, N.; Bonnet, V.; Venture, G.; Armande, N.; Fraisse, P. Human arm optimal motion analysis in industrial screwing task. In Proceedings of the 5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, Sao Paulo, Brazil, 12–15 August 2014; pp. 964–969.
31. Pereira, A.; Althoff, M. Overapproximative human arm occupancy prediction for collision avoidance. *IEEE Trans. Autom. Sci. Eng.* **2017**, *15*, 818–831.
32. Bishop, C.M.; Nasrabadi, N.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4.
33. Luo, R.; Mai, L. Human Intention Inference and On-Line Human Hand Motion Prediction for Human-Robot Collaboration. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Venetian Macao, Macau, 4–8 November 2019; pp. 5958–5964.
34. Wang, J.; Fang, Z.; Shen, L.; He, C. Prediction of Human Motion with Motion Optimization and Neural Networks. In Proceedings of the 2021 3rd International Symposium on Robotics & Intelligent Manufacturing Technology (ISRIMT), Changzhou, China, 24–26 September 2021; pp. 66–70.
35. Mainprice, J.; Berenson, D. Human-robot collaborative manipulation planning using early prediction of human motion. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 299–306.
36. Ding, H.; Reißig, G.; Wijaya, K.; Bortot, D.; Bengler, K.; Stursberg, O. Human arm motion modeling and long-term prediction for safe and efficient human–robot-interaction. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011 ; pp. 5875–5880.
37. Zhang, J.; Liu, H.; Chang, Q.; Wang, L.; Gao, R.X. Recurrent neural network for motion trajectory prediction in human–robot collaborative assembly. *CIRP Ann.* **2020**, *69*, 9–12.
38. Bouraine, S.; Fraichard, T.; Salhi, H. Provably safe navigation for mobile robots with limited field-of-views in dynamic environments. *Auton. Robot.* **2012**, *32*, 267–283.
39. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: Realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 172–186.
40. Müller, L.R.; Petersen, J.; Yamlahi, A.; Wise, P.; Adler, T.J.; Seitel, A.; Kowalewski, K.F.; Müller, B.; Kenngott, H.; Nickel, F.; et al. Robust hand tracking for surgical telestration. *Int. J. Comput. Assist. Radiol. Surg.* **2022**, *17*, 1477–1486.
41. Shin, J.; Matsuoka, A.; Hasan, M.A.M.; Srizon, A.Y. American Sign Language Alphabet Recognition by Extracting Feature from Hand Pose Estimation. *Sensors* **2021**, *21*, 5856.
42. Nowak, J.; Fraisse, P.; Cherubini, A.; Daures, J.P. Point Clouds With Color: A Simple Open Library for Matching RGB and Depth Pixels from an Uncalibrated Stereo Pair. In Proceedings of the 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Karlsruhe, Germany, 23–25 September 2021; pp. 1–7.
43. Zamboni, S.; Kefato, Z.T.; Girdzijauskas, S.; Norén, C.; Dal Col, L. Pedestrian trajectory prediction with convolutional neural networks. *Pattern Recognit.* **2022**, *121*, 108252.
44. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. {TensorFlow}: A System for {Large-Scale} Machine Learning. In Proceedings of the 12th USENIX symposium on operating systems design and implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
45. Li, C.; Chen, X. Video prediction for driving scenes with a memory differential motion network model. *Appl. Intell.* **2022**. https://doi.org/10.1007/s10489-022-03813-9.
46. Gupta, U.; Bhattacherjee, V.; Bishnu, P.S. StockNet—GRU based stock index prediction. *Expert Syst. Appl.* **2022**, *207*, 117986.
47. Islam, Z.; Abdel-Aty, M.; Mahmoud, N. Using CNN-LSTM to predict signal phasing and timing aided by High-Resolution detector data. *Transp. Res. Part C Emerg. Technol.* **2022**, *141*, 103742.
48. Cao, W.; Li, S.; Zhong, J. A dual attention model based on probabilistically mask for 3D human motion prediction. *Neurocomputing* **2022**, *493*, 106–118.