2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

Prediction of Human Arm Target for Robot Reaching Movements

Chiara Talignani Landi[1], Yujiao Cheng[3], Federica Ferraguti[1], Marcello Bonfè[2],
Cristian Secchi[1], Masayoshi Tomizuka[3]

Abstract— **The raise of collaborative robotics has allowed to create new spaces where robots and humans work in proximity. Consequently, to predict human movements and his/her final intention becomes crucial to anticipate robot next move, preserving safety and increasing efficiency. In this paper we propose a human-arm prediction algorithm that allows to infer if the human operator is moving towards the robot to intentionally interact with it. The human hand position is tracked by an RGB-D camera online. By combining the Minimum Jerk model with Semi-Adaptable Neural Networks we obtain a reliable prediction of the human hand trajectory and final target in a short amount of time. The proposed algorithm was tested in a multi-movements scenario with FANUC LR Mate 200iD/7L industrial robot.**

I. Introduction

In the recent years, the raise of collaborative robotics has allowed to create new spaces where robots and humans work in proximity. Robots are no longer separated inside safety cages, but they are considered as collaborators and helpers inside a shared workspace. This close interaction allows to create a human-robot synergy that merges complementary strengths of both agents: the high precision and repetitiveness of machines and human high-level skills [1].

In this new scenario, humans and robots play a different role [2]. Since robots can reactively adapt to environment changes and replan their trajectory in a very short time, they can be considered as human followers.

As a consequence, to predict future human movements and to infer his/her final intention becomes of paramount importance. For collision avoidance purposes, if the robot knows the human future behavior in advance, it can replan the action more efficiently, still preserving human safety. Conversely, for collaborative or physical interactive tasks, the robot should behave coherently. In assembling scenarios, for example, if the human is approaching to grasp the object carried by the robot, the robot should pause and wait for him/her. Otherwise, if the human is approaching to manually apply a modification to the robot path [3], the robot should keep following the desired trajectory, waiting for the correction to be made.

[1]Department of Science and Methods for Engineering, University of Modena and Reggio Emilia, Italy. E-mail: {chiara.talignanilandi, federica.ferraguti, cristian.secchi}@unimore.it

[2]Department of Engineering, University of Ferrara, Italy. E-mail: marcello.bonfe@unife.it

[3]Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA. E-mail:{yujiaocheng, tomizuka}@berkeley.edu

Different approaches to predict human hand trajectory and infer the final target have been proposed, especially for collaborative tasks.

A modelling approach for 3D hand trajectories in reaching motions is presented by Faraway *et al.* [4]. They propose a method for fitting the trajectory data to the control points of a Bézier curve. However, the method is largely empirical. Tamura *et al.* [5] predict the target of the human hand movement among the objects placed on the table. They define the certainty of a target based on how much the hand reduces the distance with respect to it. This is a successful approach only if the objects are placed along a row in front of the human and their positions are known a priori. Mainprice *et al.* [6] generate a prediction of human workspace occupancy exploiting the swept volume of learnt human trajectories. This volume is computed by summing the likelihood for each class of human movements using a Gaussian Mixture Model. The robot trajectory is then planned in such a way to minimize the penetration cost, avoiding collisions. This method allows to plan robot trajectories in a conservative and safe way, without inferring human target though. In Kaeami *et al.* [7], the robot builds a belief over human intentions by observing human actions. A Markov Decision Process is used to finally predict human goals. However, human actions are defined a priori.

Although there are several ways to approach to an object and users are generally different, human arm movements tend to be quite similar. Different approaches in literature try to model and describe human arm behavior: the Minimum Torque Change Model [8] [9] and the Minimum Jerk Model [10] are commonly used. The Minimum Torque Change Model is an intrinsic-dynamic-mechanical representation whose objective is to minimize the time derivative of joint torque. However, the dynamics equation of the musculoskeletal system must first be specified.

Conversely, the Minimum Jerk Model is independent of the arm dynamics, arm posture, external forces and movement duration. The basic idea is that, for point-to-point movements, human arms tend to follow a path that minimizes the third-order derivative of the position (i.e. jerk). These trajectories are usually straight and characterized by a bell-shaped velocity. The model allows to infer human goals, by fitting the parameters of the Minimum Jerk solution to experimental observations and including in the parameter set the final human arm position. A Minimum Jerk implementation for collision avoidance and human safety is presented by Dihn *et al.* [11]. The end position of the human movement is derived by finding the closest object to the current hand

978-1-7281-4004-9/19/$31.00 ©2019 IEEE

velocity vector. Objects are fixed in the space and their position is known a priori. Bratt *et al.* applied Minimum Jerk in a catching ball task. Since the ball is thrown at high speed, the human approaches to the expected point of impact and then he/she makes a small correction. For this reason the whole human movement can be approximated with one Minimum Jerk trajectory and the point of impact can be inferred. The main limitation of this approach is that half of the motion has to be observed before a prediction can be made. Namiki *et al.* [12] propose an assistive control system for a master-slave-type humanoid robot, to increase the speed of a reaching-and-grasping motion. Minimum Jerk is used to predict human arm trajectory towards different objects, whose position is known a priori. Then a particle filter is used to infer human target.

Using the Minimum Jerk model for human motion prediction requires to record the initial part, of suitable length, of the hand trajectory and then fit the parameters, including final position and time. By limiting the number of recorded points of such initial part, we can anticipate the detection of the human final target. As a consequence, the robot can also advance the planning of a collision-free trajectory if the human is not moving towards it. To reduce the number of recorded points, without affecting the long-term prediction accuracy, this paper will also exploit a short-term prediction of the future human arm trajectory, based on a different and data-driven model.

To address nonlinearity, stochasticity and individual differences related to the human motion estimation problem [13], Recurrent Neural Networks (RNNs) proved to be a powerful prediction tool [14]. RNNs allow to store past information inside hidden states, but due to this reason they are also computationally heavy to train. To take into account past history and individual differences in human motion, Semi-Adaptable Neural Networks (S-ANNs) can be used [15]. The parameters in the last layer of the Neural Network are adapted online using the Recursive Least Square Parameter Adaptation algorithm (RLS-PAA). The main advantage is that S-ANNs produce a reliable human prediction with a limited computational load.

In this paper we propose a human prediction algorithm that allows to infer if the human operator is moving towards the robot to intentionally interact with it, or he/she accidentally passing nearby to grasp a different object. The human hand position is tracked by an RGB-D camera and the initial hand movement is recorded. We adopt the Minimum Jerk model to infer the final target of the human arm movement, given the collected initial points. To anticipate the target detection, we add the short-term prediction of the human arm trajectory, based on a Neural Network, to the past collected points. To accommodate time-varying behaviors and individual differences in human motion, the parameters of the last layer of the Neural Network are adapted online, according to the Semi-Adaptable Neural Networks (S-ANN) presented in [15]. By combining the Minimum Jerk model with S-ANN we obtain a reliable prediction of the human hand trajectory and final target in a short amount of time. The prediction of the human final target allows to define if the human is going to intentionally interact with the robot. In this case, the robot should continue its trajectory, waiting for the human operator to manually execute the modification [3] or start the robot manual guidance [16]. If the final human target is a different object in the workspace, the robot should instead pause its movement to preserve human safety. The advantage of this method is to combine the accuracy of the S-ANN prediction with the long-term prediction of Minimum Jerk to obtain a reliable detection of human intention.

The main contributions of our work are:

- We only rely on the robot position to infer if the human is approaching to the robot itself. Hence, only prior knowledge of the robot trajectory is required, while an object detection system or prior knowledge of other objects positions are not necessary.
- We combine a long-term Model-Based approach (i.e. Minimum Jerk) with an accurate Data-Driven (i.e. S-ANNs) approach to obtain a reliable prediction of human hand trajectory.
- With respect to [15], we found a suitable size of the prediction horizon for the S-ANN output. This size is a trade-off between the S-ANN adaptation convergence, the accuracy of the final target detection and the time required to compute the latter.
- We detect online all the point-to-point hand movements of an arbitrary sequence, which represents a realistic working scenario.

II. Problem statement

In novel industrial scenarios, it is likely that a robot executes its task while a human operator is working nearby, either to collaborate at the same task or to perform a different operation. The focus of this work is to determine whether the human operator is going to approach to the robot, in order to physically interact with it. This kind of predictive approach on human intention allows to guarantee human safety and an effective collaboration.

If the human is intentionally approaching to the robot to start a physical interaction, the robot should not consider the human gesture as an unsafe incoming collision. Whereas if the human operator is doing random movements towards other objects, that can possibly be unsafe for him/her, the robot should modify its trajectory consequently.

The proposed human movement prediction allows to infer the final goal early in time, in order to promptly and accordingly modify robot's behavior.

III. Background on robot coaching

In a cooperative scenario, it can happen that the user needs to change a portion of the robot trajectory in a desired way. This is due, for example, to the presence of an obstacle placed along the robot's path or a change of the working area. This problem is known as "robot coaching". In our previous work [3], an admittance-controlled robot tracks the desired trajectory. When the user needs to change a portion of the trajectory, he/she grasps the end-effector of the robot,

or the tool attached to it. The controller detects this operation and the stiffness is changed to make the robot easily drivable by the user, who manually executes the correction. Finally, the controller detects when the user releases the end-effector, the stiffness is restored to a high value and the correction is optimally joined to the previous trajectory.

As previously mentioned, switch between a tracking mode and a compliant mode (and vice versa) is achieved through a stiffness variation in the admittance control. However, this operation can generate unsafe robot behaviors, due to the loss of passivity [17]. Energy tanks have been introduced to flexibly dealing with energy exchange [18]. They allow to store the system dissipated energy and use it to implement active behaviors, while keeping the system passive and safe.

Formally, the dynamics of an admittance controlled robot with time varying parameters can be represented by:

$$M_d(t)\ddot{\tilde{y}} + D_d(t)\dot{\tilde{y}} + K_d(t)\tilde{y} = F_{ext} \tag{1}$$

where $M_d(t)$, $D_d(t)$, $K_d(t) \in \mathbb{R}^{6\times6}$ are desired symmetric positive definite inertia, damping and stiffness matrices respectively. $\tilde{y}(t)$ is the pose error, with $\tilde{y}(t) = y(t) - y_d(t)$, $y(t)$ and $y_d(t) \in \mathbb{R}^6$ are the robot pose[1] and the desired pose. $F_{ext} \in \mathbb{R}^6$ is the external force measured by a 6-DOFs Force/Torque (F/T) sensor mounted on the end-effector.

If we consider a variation of the stiffness matrix, while keeping the inertia and damping constant, the variation of the system energy can be expressed as:

$$\dot{H} = \dot{\tilde{y}}^T F_{ext} + \left[-\dot{\tilde{y}}^T D_d \dot{\tilde{y}} + \frac{1}{2}\tilde{y}^T \dot{K}_d(t)\tilde{y}\right] \tag{2}$$

Since $D_d \geq 0$, the first term in the brackets is always negative and energy can be introduced only by the second term, when the stiffness increases.

If we augment the admittance model by adding the energy tank, it is possible to use the dissipated energy to implement a stiffness increase and to track back the original trajectory after the modification is done.

The maximum value of stiffness augmentation in a time interval $[t_0, t_{end}]$, according to the energy T stored in the tank, is:

$$\dot{k}_i(t) \leq \frac{2\left(T(t_0) - \delta\right)}{\|\tilde{y}_M\|^2 \left(t_{end} - t_0\right)} \quad \forall i = 1, \ldots, n \tag{3}$$

where k_i is each element on the stiffness matrix diagonal. δ represents the minimum energy value to avoid tank depletion and $\|\tilde{y}\|_M$ is the maximum tracking error that the robot can experience in its workspace.

In conclusion, a stiffness increase can be planned according to condition (3) in order to maintain the passivity of the overall system and switch to the tracking mode.

[1]It is assumed that a low-level controller is designed and tuned in such a way that the pose tracking error is negligible, namely $y \simeq y_d$, where y_d is the reference pose computed by the admittance control.

IV. Human Motion Prediction

A. Minimum Jerk Model

Flash and Hogan [10] stated that, for unconstrained point-to-point movements, human arms move along a smooth trajectory while minimizing the mean-square jerk. This statement can be formalized, considering a spatial trajectory [19] moving from an initial position to a final one in a given time interval, as model fitting problem requiring to minimize the following cost function:

$$C = \frac{1}{2}\int_{t_i}^{t_f} \left(\frac{d^3x}{dt^3}\right)^T \left(\frac{d^3x}{dt^3}\right) dt \tag{4}$$

where $x \in \mathbb{R}^3$ is the time-varying hand position.

Given a start time t_i, a final time t_f and boundary conditions at $x_i = x(t_i)$ and $x_f = x(t_f)$ (i.e. position, velocity and acceleration), the unique solution of the optimization problem is a fifth-order polynomial. If we assume that the movement starts and ends with zero velocity and acceleration (i.e. point-to-point movement), the polynomial solution results in:

$$x = x_i + (x_f - x_i)\left(10\tau^3 + 15\tau^4 + 6\tau^5\right) \tag{5}$$

where $\tau = (t - t_i)/(t_f - t_i)$.

In the following, we will consider the human grasping motion as a point-to-point movement even if the robot moves in the workspace. Indeed, the robot speed is necessarily limited during collaborative tasks, hence if the human approaches with the intent to physically interact, his/her hand would reach the robot with negligible velocity and acceleration. However, non zero velocity and acceleration at the initial and final points could also be taken into account using the generalized solution described in [19].

The trajectory described by (5) is a straight line between the initial and final positions, with a bell-shaped velocity profile. Given the parameters $r = [x_i^T, x_f^T, t_i, t_f]$, the human arm trajectory can be computed for each instant of time.

However, the prediction of the human arm target must be setup as a different problem: given an observed partial arm movement, how can we find the parameters r in such a way that a partial arm movement fits the Minimum Jerk polynomial (5).

We assume that the human arm movement is observed by a tracking system with a discrete-time output. Therefore, if the beginning of the trajectory is observed, the vector of collected positions in the Cartesian space is $\bar{\chi}_k = [\bar{x}_1, \ldots, \bar{x}_k]$, where $\bar{x}_k$ is the observed hand position at time t_k. Since the beginning of the motion can be easily detected (i.e. by monitoring the hand velocity), the parameters x_i and t_i are known.

Determining the final goal x_f at the corresponding time t_f is not trivial. In a shared human-robot workspace, different objects can surround the robot and their location is unknown. Hence, we cannot consider different final positions and evaluate the fitting for each one. This procedure would require an object identification system and it would be computationally expensive to be implemented online. However, the current

position of the robot is usually known, together with its trajectory. If we assume that the human wants to grab the end-effector, we can exploit this information by considering the end-effector Cartesian position as the parameter x_f. Hence, the only unknown parameter that has to be computed is the final time t_f.

Therefore, we fit the collected points with the Minimum Jerk trajectory that ends in the end-effector position x_f, by searching the value of t_f that minimizes the following cost function:

$$C(t_f) = \frac{1}{2}\sum_{k=i}^{n}\|x_k - \bar{x}_k\|^2 \tag{6}$$

where the values x_k are computed from (5) at the sampling instants t_k of the collected points $\bar{x}_k$.

Then, we can evaluate the fitting quality by computing the Root Mean Square (RMS) norm error between the collected points and those calculated by the fitted model. The fitting is performed in a tridimensional space.

In conclusion, if the human is moving towards the robot, the quality of the fitting is good and the RMS norm error is low. Otherwise, if the human is moving towards another object, the RMS norm error will be higher than a given threshold. The tuning of such a threshold will be described in Sec. VII.

V. Semi-Adaptable Neural Networks

To anticipate the determination of human target, a prediction of the human hand based on Neural Networks can be added. In this way, the predicted trajectory points are added to the collected points to fit the Minimum Jerk model.

To accommodate both the time varying behavior of human and individual differences among different people, we adopted the Semi-Adaptable Neural Networks (S-ANNs). In this model we only adapt the weights of the output layer online, leaving the weights of the remaining layers as obtained from offline training. The reason behind this choice is that the input of the last layer can be seen as human features, that are linearly combine to output the prediction. For different humans, these features can be combined differently, but also for the same human these features can vary in time.

The transition model of human joint motion can be formulated as

$$\chi(k+1) = f^*(\chi^*(k)) + w_k, \tag{7}$$

where $\chi(k+1) \in \mathbb{R}^{3M}$ denotes human's M-step future trajectory state of the hand joint at time steps $k+1, k+2, \ldots, k+M$ in a Cartesian coordinate system. $M \in \mathbb{N}$ is the prediction horizon. Denoting $x_k \in \mathbb{R}^3$ the Cartesian position of the joint at time step k, $\chi(k+1)$ is a stack of M future joint positions $x_{k+1}, x_{k+2}, ..., x_{k+M}$. $\chi^*(k) \in \mathbb{R}^{3N}$ denotes human's past N-step trajectory of the joint at time steps $k, k-1, \ldots, k-N+1$, constructed by stacking the position vectors $x_k, x_{k-1}, ..., x_{k-N+1}$. $w_k \in \mathbb{R}^{3M}$ is a zero-mean white Gaussian noise. The function $f^*(\chi^*(k)) : \mathbb{R}^{3N} \rightarrow \mathbb{R}^{3M}$ represents the transition of the human motion, which takes historical trajectory as inputs, and outputs the future positions of the joint.

The human motion transition model f^* is approximated by an n-layer neural network with ReLU (Rectified Linear Unit) activation function:

$$f^*(\chi^*(k), a) = W^T \max(0, g(U, \chi^*(k))) + \epsilon(s_k), \tag{8}$$

where $s_k = [\chi^*(k)^T, 1]^T \in \mathbb{R}^{3N+2}$ is the input vector, g denotes $(n-1)$ - layer neural network, whose weights are packed in U. $\epsilon(s_k) \in \mathbb{R}^{3M}$ is the function reconstruction error, which goes to zero when the neural network is fully trained. $W \in \mathbb{R}^{n_h \times 3M}$ is the weights of the last layer, where $n_h \in \mathbb{N}$ is the number of neurons in the hidden layer.

By stacking all the column vectors of W, we get a time varying vector $\theta \in \mathbb{R}^{3Mn_h}$ to represent the weights of last layer. θ_k denotes its value at time step k. To represent the extracted features, we define a new data matrix $\Phi_k = diag(\max(0, g(U, s_k))_1, ..., \max(0, g(U, s_k))_M)$, $\Phi_k \in \mathbb{R}^{3M \times 3Mn_h}$.

Using Φ_k and θ_k, (7) and (8) can be written as

$$\chi(k+1) = \Phi_k \theta_k + w_k. \tag{9}$$

Let $\hat{\theta}_k$ denotes the parameter estimate at time step k, and let $\tilde{\theta}_k = \theta_k - \hat{\theta}_k$ be the parameter estimation error. We define the *a priori* estimate of the state and the estimation error as:

$$\hat{\chi}(k+1|k) = \Phi_k \hat{\theta}_k, \tag{10}$$

$$\tilde{\chi}(k+1|k) = \Phi_k \tilde{\theta}_k + w_k. \tag{11}$$

In this paper, recursive least square parameter adaptation algorithm (RLS-PAA) with forgetting factor [15] is applied to asymptotically adapt the parameters in the neural network. The core idea of RLS-PAA is to iteratively update the parameter estimation $\hat{\theta}_k$ and predict $\chi(k+1)$ when new measurements become available. The parameter update rule of RLS-PAA can be summarized as:

$$\hat{\theta}_{k+1} = \hat{\theta}_k + F_k \Phi_k^T \tilde{\chi}(k+1|k), \tag{12}$$

where F_k is the adaptation gain updated by:

$$F_{k+1} = \frac{1}{\lambda_1(k)}\left[F_k - \lambda_2(k)\frac{F_k \Phi_k \Phi_k^T F_k}{\lambda_1(k) + \lambda_2(k)\Phi_k^T F_k \Phi_k}\right] \tag{13}$$

where $0 < \lambda_1(k) \leq 1$ and $0 < \lambda_2(k) \leq 2$. Typical choices for $\lambda_1(k)$ and $\lambda_2(k)$ are:

1) $\lambda_1(k) = 1$ and $\lambda_2(k) = 1$ for standard least squares gain.
2) $0 < \lambda_1(k) < 1$ and $\lambda_2(k) = 1$ for least squares gain with forgetting factor.
3) $\lambda_1(k) = 1$ and $\lambda_2(k) = 0$ for constant adaptation gain.

VI. Proposed Prediction Algorithm

The proposed approach combines Minimum Jerk fitting with Semi-Adaptable Neural Networks (S-ANNs) to obtain a long-term human arm prediction, inferring if the human is moving towards the robot or not.

Algorithm 1: Proposed prediction algorithm

1 Collect the initial portion $\bar{\chi}_k$ of the observed human hand trajectory
2 Add the predicted points $\hat{\chi}(k+1|k)$ given by the S-ANN
3 Perform the Minimum Jerk trajectory fitting, considering x_f in (5) as the future robot position
4 Evaluate the RMS norm error between the points in $\bar{\chi}_k \cup \hat{\chi}(k+1|k)$ and the points fitted by the Minimum Jerk
5 **Output:** If RMS norm error $<$ threshold then the human is approaching the robot, else the human is approaching a different object.

The proposed procedure is described in Alg. 1. It is worth noting that, since in (Line 2) we add the predicted points from the S-ANN, the robot position used as the Minimum Jerk parameter x_f has to be determined consequently, to obtain a coherent fitting. Hence, if the last point of the S-ANN prediction is, for example, 0.5 s in the future, the corresponding future position of the robot has to be considered. Since we assume that the robot trajectory is known in advance, as is common in industrial scenarios, this data can be easily obtained.

To distinguish whether the human is approaching to the robot or to a different object, a proper threshold on the prediction model fitting quality has to be defined a priori. Its value depends on how wide the grasping area on the robot is. In our case we consider that human operator always approach to the tool placed on the end-effector, to grab it and apply the manual modification. A discussion on the threshold value is done in Section VII. If the human is moving towards a different object, the robot behaves to guarantee human safety (i.e. it stops, it pauses or it replans the trajectory to avoid the collision).

A. Sub-movements detection

In a cooperative scenario, the human operator executes several sequential movements inside the workspace: a few of them can be directed towards the robot for collaborating, others can be directed somewhere else. To properly fit data with Minimum Jerk model, it is important to detect the beginning and ending of each sub-movement.

For point-to-point movements the initial and final velocities and accelerations are zero. Hence, a velocity-based threshold logic can detect when the human is starting and stopping the movement. Since skeletal data coming from vision sensors are usually noisy, a smoothing differentiation filter, based on the Savitzky-Golay algorithm [20], was used to compute the hand velocity while reducing the noise.

VII. Experimental Results

The experiments are performed on a FANUC LR Mate 200iD/7L as shown in Fig.1. To track human right hand position, we used a Kinect V2 RGB-D camera and the skeleton tracking software based on the Kinect for Windows SDK. The robot low-level controller is deployed in Simulink Real-Time on a target PC with Intel i5-3340 Quad-Core CPU. The Kinect sensor is connected to the Windows host PC, that executes the algorithm of Sec. VI and the fitting error as described in Sec. IV-A. Both the host and the target PC communicate with an external PC through a UDP socket. On the external PC the admittance control and the manual correction software components are implemented using Orocos real-time framework. Both the low-level controller and the Orocos components run at a 125 Hz frequency, whereas the Kinect frames are updated every 0.033 s.

A three-layer neural network was trained with 100 movements going towards different objects in the workspace, either random ones or the robot end-effector. The number of nodes in the input and output layers is respectively 9 and 30: 3 points (x, y, z coordinates) are used as a past history and 10 points are predicted in the future.

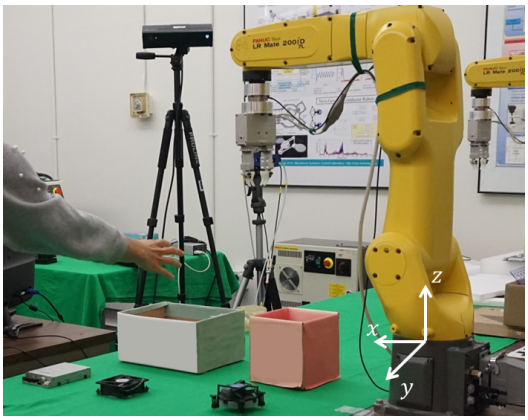


Fig. 1. Experimental setup. The Kinect tracks the human right hand. The robot is a Fanuc LR Mate 200iD/7L.

A. Adaptable Neural Network prediction points

After several experimental trials, we found that 30 points are necessary to perform a reliable Minimum Jerk fitting. Since the Kinect updates every 33 ms, the collection of 30 points would require almost 1 s. Point-to-point hand movements are usually within a similar time interval (i.e. one to few seconds), hence we would like to anticipate as much as possible the detection of the human goal, to guarantee his/her safety. For this reason we added the Adaptable Neural Network, in such a way that the number of collected data from Kinect can be reduced and the decision making process can start earlier. However, reducing the number of predicted data can cause a loss of accuracy and the target identification can be negatively affected. The final solution should be a trade-off between accuracy and time.

Neural Networks are usually not suitable for long-term predictions. Moreover, if we want to adapt the last layer parameters, an error computation between the predicted points and the collected ones (i.e. ground truth) should be performed. By choosing 3 points as a past history and 10 points as the number of predictions, we can exploit the advantages of parameters adaptation still reducing the detection time to 0.66 s (i.e. 20 points collected from Kinect and 10 point predicted). In Table I the error computation sequence (i.e. (11)) is depicted. The table shows that, by

TABLE II

ERROR COMPARISON BETWEEN 5 AND 10 PREDICTED POINTS

Number of predicted points	**5**	**10**
RMS Error [m]	0.061	0.074
	0.037	0.050
	0.069	0.046
	0.055	0.048
	0.041	0.088
	0.059	0.057
	0.063	0.072
	0.060	0.039
	0.065	0.069
	0.062	0.037
Mean [m]	**0.058**	**0.057**
Standard Deviation [m]	**0.009**	**0.016**

considering 20 recorded points and 10 predicted, the Neural Network adaptation is performed 8 times (from point 13^{th} to point 20^{th}).

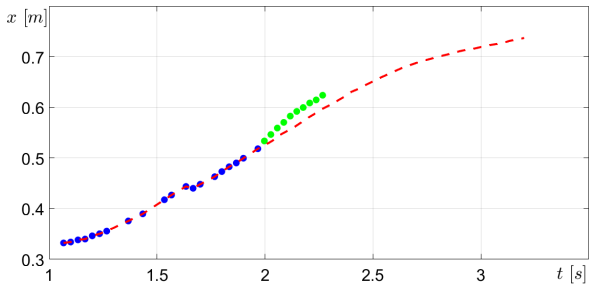
To show the reliability of choosing 10 predicted points, we compare the RMS error in two cases. The human operator is tracked while is moving towards the robot end-effector for 10 times. In the first experiment, out of the 30 points needed for the Minimum Jerk fitting, 25 are collected from Kinect and 5 points are predicted from the Neural Network. In the second experiment, 20 points are collected from the Kinect and 10 predicted. As show in Table II, the computed mean value of the RMS error is similar and the movement towards the end-effector is always detected correctly. The small difference in the Standard Deviation shows that, even in the case of 10 predicted points, the intention of going towards the robot is correctly detected. In fact, since we chose an error threshold of 0.10 m, in both cases the error values are under this threshold.

In conclusion, we chose to use 10 predicted point as a trade-off to exploit the advantages of the Adaptable Neural Network yet anticipating the detection of the human intention.

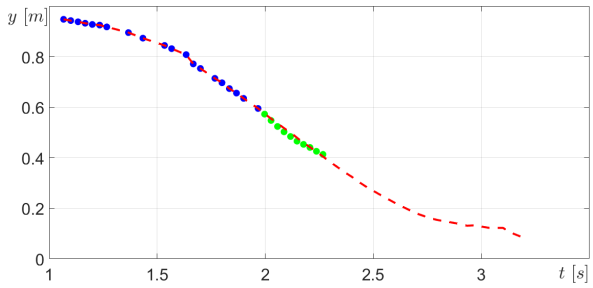
B. Comparison between recorded data and Adaptable Neural Network prediction

As mentioned previously, we experimentally found that 30 points are required to obtain a suitable trajectory fitting using the Minimum Jerk. However, a total of almost 1 s would be necessary to predict the final human target. In a human-robot collaborative scenario, human safety is of paramount importance. In this context, anticipating human final intention gives more time to the system to behave accordingly, i.e. to replan the trajectory or to temporarily stop. For this reason we chose to consider 20 recorded points from the Kinect and then to add 10 predicted points from the Neural Network. By adopting this strategy, the overall prediction time is reduced to 0.66 s.

To show the reliability of the Neural Network prediction, we compare the RMS error obtained in two different cases. In the first one, 30 recorded points from Kinect were used to fit the Minimum Jerk trajectory. In the second one, only 20 points were used and then we add 10 predicted points, referred to the last recorded point and derived from the Adaptable Neural Network.



(a) x-direction



(b) y-direction

Fig. 2. Comparison between recorded points (dashed red line) and recorded points (blue points) with predicted ones (green points) from the Adaptable Neural Network.

Figure 2 shows the trajectories along x and y directions (similar behavior along z). The first 20 points are the same in both cases (dashed red line and blue dots). As can be clearly seen, the predicted points (green dots) follow the trend of the recorded ones.

The difference between the two RMS errors is 6 mm, hence we can consider the ANN prediction accurate enough to substitute the recorded points. For the sake of completeness, all the recorded trajectory is depicted in Figure 2 (red dashed line), but only 30 points were used to compute the RMS error.

C. Comparison between different movements

The objective of the presented work is to infer if the human is approaching to the robot or is going towards another object. To evaluate the performance of the proposed method, two types of movement are compared.

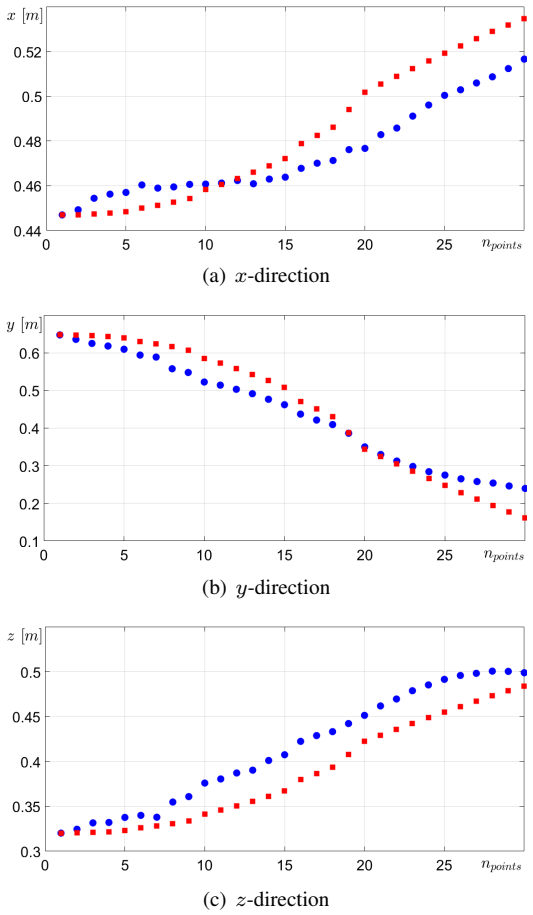
In the first one, the human is approaching to the tool attached to the robot end-effector, to grasp it and manually modify the trajectory. As depicted in Fig. 3, the Minimum Jerk (red squares) fits the reference data (blue dots). As described in Sec. VII-B, the reference data are given by 20 recorded points of human right hand, and then 10 predicted points from the Adaptable Neural Network are added. The computed RMS error is 0.05 m.

Conversely, the example of a movement going towards another object is depicted in Fig. 4. The object is located along the same y direction, but at a different x location and height z. In Figure 1, it is the left box position. As can be seen in Fig. 4, the fitting is inaccurate and the final RMS

TABLE I

ERROR COMPUTATION FOR 20 COLLECTED POINTS AND 10 PREDICTED POINTS

time step	collected point	past points (input layer)	predicted points (output layer)	error computation
1	$\bar{x}_1$			
2	$\bar{x}_2$			
3	$\bar{x}_3$	$[\bar{x}_1,\bar{x}_2,\bar{x}_3]$	$[\hat{x}_4,...,\hat{x}_{13}]$	
4	$\bar{x}_4$	$[\bar{x}_2,\bar{x}_3,\bar{x}_4]$	$[\hat{x}_5,...,\hat{x}_{14}]$	
...	...	...	...	
13	$\bar{x}_{13}$	$[\bar{x}_{11},\bar{x}_{12},\bar{x}_{13}]$	$[\hat{x}_{14},...,\hat{x}_{23}]$	$[\bar{x}_4 - \hat{x}_4,..., \bar{x}_{13} - \hat{x}_{13}]$
...	...	...	...	...
19	$\bar{x}_{19}$	$[\bar{x}_{17},\bar{x}_{18},\bar{x}_{19}]$	$[\hat{x}_{20},...,\hat{x}_{29}]$	$[\bar{x}_{10} - \hat{x}_{10},...,\bar{x}_{19} - \hat{x}_{19}]$
20	$\bar{x}_{20}$	$[\bar{x}_{18},\bar{x}_{19},\bar{x}_{20}]$	$[\hat{x}_{21},...,\hat{x}_{30}]$	$[\bar{x}_{11} - \hat{x}_{11},...,\bar{x}_{20} - \hat{x}_{20}]$

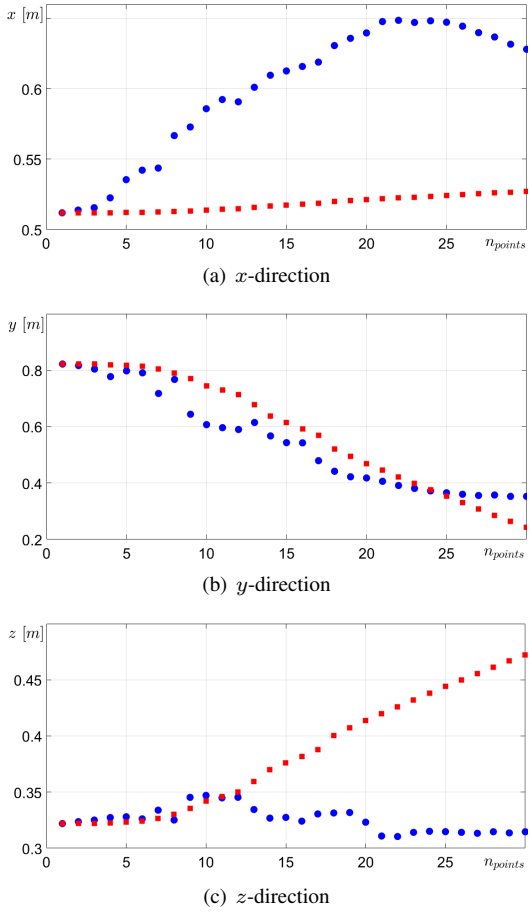


(a) x-direction

(b) y-direction

(c) z-direction

Fig. 3. Comparison between Minimum Jerk fitting (red squares) and the reference data (blue dots) going towards the end-effector.



(a) x-direction

(b) y-direction

(c) z-direction

Fig. 4. Comparison between Minimum Jerk fitting (red squares) and the reference data (blue dots) going towards an object, that is different from the robot end-effector.

error results in 0.15 m, which exceeds the error threshold of 0.10 m. Hence, the human is moving the hand towards an object different from the robot.

It is worth noting that in this paper we consider the hand posotion as the most relevant human joint to be tracked. Conversely, in [15] the tracking is focused on the human wrist. Our different choice is motivated by the fact that the fitting accuracy improves significantly. The improvement reflects the human movement to approach an object with the hand, not the wrist. Therefore, the position mismatch between the hand point and the wrist point causes an error in the detection of the final target. The distance between these two points can't be compensated only along one direction, since the arm and the wrist are rotated by an unknown angle.

D. Sub-movements and error threshold

In a cooperative scenario, it is likely that the human operator works side by side with the robot. In the video and the setup of Fig. 1, the human is placing objects inside the two boxes. We make the assumption that the human approaches to the robot to manually apply a modification to the trajectory. Hence, the robot must continue its path, following the desired trajectory. Conversely, if the human is

placing an object inside a box, the robot pauses to avoid the collision.

A Savitzky-Golay differentation filter [20] is used to estimated the hand velocity from Kinect positions. The Savitzky-Golay filter is non-causal and its output estimation is inherently delayed by a number of time steps that are half of the size of its input buffer. Such a delay can be compensated by properly choosing data points for the trajectory fitting. For example, if a new movement is detected at time t_s but the filter introduced a delay of five time steps, the first point to be considered is the one at t_{s-5}.

The velocity thresholds to detect a sub-movement stopping and the starting are empirically set to 0.15 m/s^2 and 0.2 m/s^2. The RMS norm error threshold to discriminate the interaction intent is set equal to 0.10 m . However, When the human is retracting the arm or is reaching an object far from the robot, the robot should not pause. Hence, if the error is greater that 0.40 m (value experimentally found), the robot does not modify its behavior and it keeps following its trajectory.

VIII. Conclusions

In this paper we proposed a prediction method to infer if the human is approaching to the robot. The human right hand position is tracked by a Kinect and the beginning of the arm movement is captured. The recorded points are used to fit a Minimum Jerk model whose final position is the robot end-effector. Then, a fitting quality evaluation is used to distinguish if the human is approaching to the robot or is moving towards other objects in the workspace. To anticipate the detection and improve the human operator safety, an Adaptable Neural Network is used, to predict future positions of the human hand based on previous training.

Future works aim at implementing a safe and collision-free trajectory replanning, to avoid robot pauses and to guarantee a higher efficiency. In addition, a distinction between general movements and movements directly pointed to an object [5] should be added. General movements are usually limited to a restricted working area and hands turn around frequently. This behavior means that the human is not reaching an object. Hence these kind of movements do not fit the Minimum Jerk model and safety countermeasures can be directly applied, to avoid collisions.

References

[1] A. Levratti, G. Riggio, C. Fantuzzi, A. De Vuono, and C. Secchi, "Tirebot: A collaborative robot for the tire workshop," *Robotics and Computer-Integrated Manufacturing*, vol. 57, pp. 129–137, 2019.

[2] C. Liu and M. Tomizuka, "Designing the robot behavior for safe human robot interactions," in *Trends in Control and Decision-Making for Human Robot Collaboration Systems*, Y. Wang and F. Zhang, Eds. Springer, 2017, pp. 241–270.

[3] C. T. Landi, F. Ferraguti, C. Fantuzzi, and C. Secchi, "A passivity-based strategy for coaching in human-robot interaction," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[4] J. J. Faraway, M. P. Reed, and J. Wang, "Modelling three-dimensional trajectories by using bézier curves with application to hand motion," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 56, no. 5, pp. 571–585, 2007.

[5] Y. Tamura, M. Sugi, J. Ota, and T. Arai, "Prediction of target object based on human hand movement for handing-over between human and self-moving trays," in *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication (ROMAN)*, 2006, pp. 189–194.

[6] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 299–306.

[7] A.-B. Karami, L. Jeanpierre, and A.-I. Mouaddib, "Human-robot collaboration for a shared mission," in *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*, 2010, pp. 155–156.

[8] Y. Uno, M. Kawato, and R. Suzuki, "Formation and control of optimal trajectory in human multijoint arm movement," *Biological cybernetics*, vol. 61, no. 2, pp. 89–101, 1989.

[9] E. Nakano, H. Imamizu, R. Osu, Y. Uno, H. Gomi, T. Yoshioka, and M. Kawato, "Quantitative examinations of internal representations for arm trajectory planning: minimum commanded torque change model," *Journal of Neurophysiology*, vol. 81, no. 5, pp. 2140–2155, 1999.

[10] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[11] K. H. Dinh, O. Oguz, G. Huber, V. Gabler, and D. Wollherr, "An approach to integrate human motion prediction into local obstacle avoidance in close human-robot collaboration," in *Proceedings of IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 2015, pp. 1–6.

[12] T. A. Namiki, Y. Matsumoto, T. Maruyama, and Y. Liu, "Vision-based predictive assist control on master-slave systems," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[13] Z. Peng, T. Genewein, and D. A. Braun, "Assessing randomness and complexity in human motion trajectories through analysis of symbolic sequences," *Frontiers in human neuroscience*, vol. 8, p. 168, 2014.

[14] P. Ghosh, J. Song, E. Aksan, and O. Hilliges, "Learning human motion models for long-term predictions," in *3D Vision (3DV), 2017 International Conference on*, 2017, pp. 458–466.

[15] Y. Cheng, W. Zhao, C. Liu, and M. Tomizuka, "Human motion prediction using adaptable neural networks," *arXiv preprint arXiv:1810.00781*, 2018.

[16] C. T. Landi, F. Ferraguti, C. Fantuzzi, and C. Secchi, "Tool compensation in walk-through programming for admittance-controlled robots," in *Proceedings of IEEE Industrial Electronics Society (IECON)*, 2016.

[17] F. Ferraguti, N. Preda, A. Manurung, M. Bonfe, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini, and C. Secchi, "An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1073–1088, 2015.

[18] C. Secchi, S. Stramigioli, and C. Fantuzzi, "Position drift compensation in port-hamiltonian based telemanipulation." in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 4211–4216.

[19] N. Fligge, J. McIntyre, and P. van der Smagt, "Minimum jerk for human catching movements in 3d," in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, 2012, pp. 581–586.

[20] J. Luo, K. Ying, P. He, and J. Bai, "Properties of savitzky–golay digital differentiators," *Digital Signal Processing*, vol. 15, no. 2, pp. 122–136, 2005.