Prescient Teleoperation of Humanoid Robots

Luigi Penco[1], Jean-Baptiste Mouret[2] and Serena Ivaldi[2]

***Abstract*— Teleoperated humanoid robots are ideally suited to act as human avatars in remote environments. Unfortunately, their deployment is hindered by the communication delays between the human input and the video feedback from the robot. Here, we introduce a direct teleoperation system in which the operator receives a synchronized video feed of real images, even when the communication channel imposes a 1 to 2-second delay. Our key idea is to leverage machine learning to allow the robot to execute commands before the operator performs them, so that the operator receives a delayed video stream that is almost indistinguishable from real-time feedback. In our experiments, the iCub humanoid robot (32 degrees of freedom) was successfully controlled to perform several whole-body manipulation tasks, including reaching different targets, picking up an object, and moving a box. This new technique may enable real-life avatars on long-range radio networks, from remote maintenance to space missions.**

I. Introduction

Teleoperated robots have the potential to replace humans in numerous hazardous scenarios, ranging from contaminated environments to outer space. For such robots to be effective, they must possess both intuitive controls and high versatility, enabling operators to adapt to various situations. Humanoid robots are among the most promising solutions to meet these challenges, as they offer the capability for bimanual manipulation, walking, crawling, or climbing, while also being inherently intuitive for human teleoperation due to their similar physical structure.

Substantial progress in whole-body control and motion capture now makes it possible to map a whole-body motion of a human to a robot, despite the difference in physical constraints and dynamics [1]–[5]. This *whole-body direct teleoperation* approach allows the operator to control the robot creatively, quickly, and intuitively, and has been demonstrated in several lab experiments [1] and even competitions (e.g., ANA Avatar Xprize) [4].

Whole-body direct teleoperation becomes impractical in the presence of delays because operators rely on the visaul feedback from the robot to adapt and adjust their commands. Unfortunately, delays are inherent to many of the most promising applications of teleoperated humanoids. For example, the roundtrip delay is approximately 0.81s when teleoperating a robot on Earth from the International Space Station [6], and it can be as long as 2.6 seconds with a laser communication link between the Earth and the Moon [7]. The DARPA Robotics Challenge [8] set the average delay to 1 second (with a maximum of 30 seconds), and the NASA Space Challenge [9] set it to 10 seconds (with a maximum of 20 seconds). These delays significantly impact the performance of the operator and limit the use of direct teleoperation in many real-life scenarios.

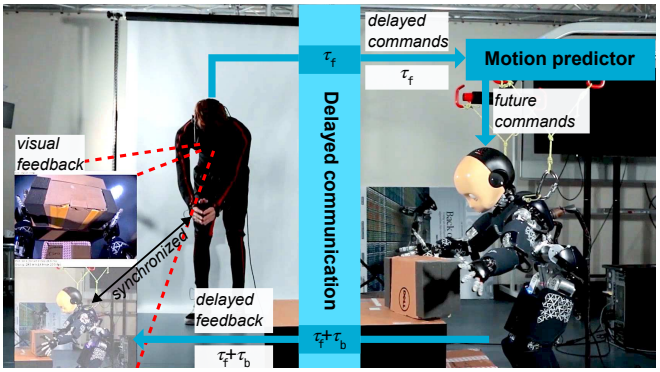

Fig. 1. **Concept of prescient teleoperation.** The robot executes the commands *before it receives them* thanks to a data-driven predictor. In this example, the robot drops a box onto a table before the human commands it to open its arms. When the predictions are accurate, by anticipating both the forward delay τ_f and the backward delay τ_b, the visual feedback appears to be synchronized with the operator. The robot sends two video stream, which are both displayed in the virtual reality headset: an internal view, from the head of the robot, and an external view. Video: https://youtu.be/_HnQw3ZZ3T8.

Here, we introduce a novel concept of direct teleoperation, where the operator receives a synchronized video feed of *real images* from the remote system (robot and environment/system cameras), even when the communication channel introduces a delay of 1 to 2 seconds. Our key idea is that if the robot executes the desired movement *before* the operator performs it, then the operator will observe a delayed video feed that will be almost indistinguishable from a real-time feed (Fig. 1). At each time-step, the robot analyses the data received from the operator, measures the communication time, estimates the communication time required to send the video feedback, and predicts the operator's most likely motions in the upcoming seconds. This prediction makes it possible to execute the command with enough anticipation so that the user receives a video feed that corresponds to the past motion of the robot but that aligns with the present time for the operator. We refer to this prediction-based feedback scheme as *prescient teleoperation*.

*This work was supported by the European Union H2020 and HE Research and Innovation Programs under GA No. 731540, No. 101070596 (projects AnDy, euROBIN), the European Research Council (ERC) under GA No. 637972 (project ResiBots), the Inria-DGA grant ("humanoïde résilient"), and the Inria "ADT" wbCub/wbTorque. Experiments were performed in the Creativ'Lab facilities, supported by the FEDER Sciarat.

[1]Luigi Penco is with the Florida Institute for Human and Machine Cognition. lpenco@ihmc.org

[2]Jean-Baptiste Mouret and Serena Ivaldi are with Inria Nancy — Grand Est, CNRS and Université of Lorraine. {jean-baptiste.mouret,serena.ivaldi}@inria.fr

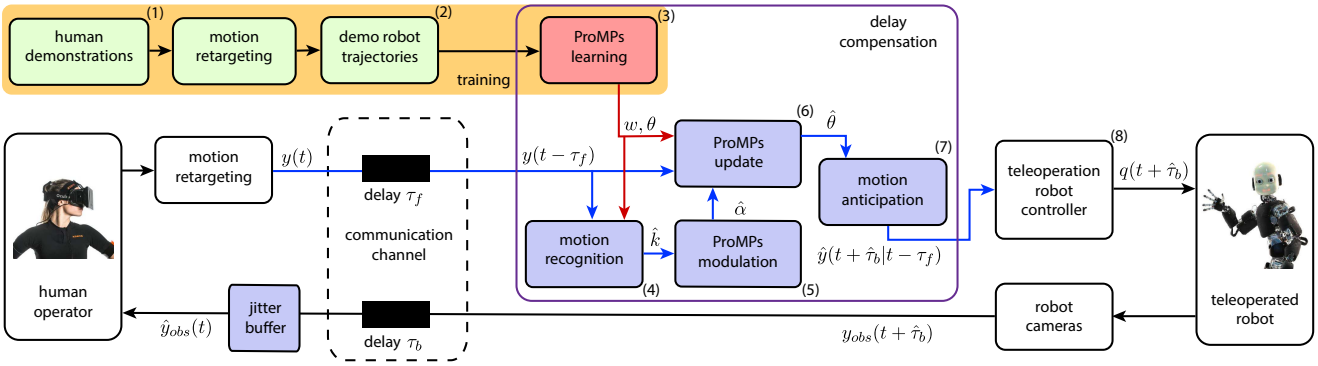

Fig. 2. **Flowchart of the prescient teleoperation system.** During the training phase, (1) the human operator teleoperates the robot without any significant delay (local network), and performs a variety of tasks. The retargeted human motions represent the robot trajectories demonstrations (2) that are later used to train the Probabilistic Movement Primitives (ProMPs). A ProMP is learned for every task (3). When teleoperating the robot in a non-ideal network, the ProMPs are used to predict the robot movement: (4) the system recognizes the current task, i.e. it identifies the most likely ProMP; (5) it estimates the speed of the teleoperated motion and (6) updates the selected ProMPs according to the delayed observed via-points; (7) it compensates for the delay by selecting the trajectory of the posterior ProMPs at the right timestep. This ensures synchronization between the user's movements and what they observe from the remote cameras on the robot's side. The resulting trajectories are then tracked by the whole-body controller (8), which calculates the joint commands for the robot.

This paper describes the system and presents several experiments to show the contribution of the different components, in terms of prediction, delay compensation, and tracking error. More detail and additional figures, not included in the manuscript for space limits, are reported in the appendices I-A to V (including supplementary Fig. S1-S8), available at `https://hucebot.github.io/www_prescient_humanoids/`. A video showing prescient teleoperation with the humanoid robot iCub is available online: `https://youtu.be/_HnQw3ZZ3T8`.

II. Previous Work

The initial studies with teleoperated robot manipulators and delays [10] suggested that users often adopt a "move-and-wait" strategy to avoid overcompensating for delayed perceived errors. However, subsequent experiments demonstrated that this strategy is ineffective even with time delays of approximately 0.3 seconds [11].

Due to the challenges posed by delays, many teleoperation systems focused on supervised autonomy. In this approach, the operator provides high-level goals, such as selecting an object on a screen and pressing a button to grasp it, and the robot autonomously carries out the motions to fulfill the goals [3], [5], [12]–[14]. This method requires careful planning, making it difficult to spontaneously generate new behaviors for new goals or adapt to changes. Moreover, the division of goals into discrete operations often results in reduced teleoperation speed and fluidity: the operator must select an operation, wait for its completion, and then activate the next one, resulting in a sequential process.

In direct teleoperation, most research on delays focused on designing a stable control loop for systems that provide haptic feedback, specifically in multilateral teleoperation [6], [15]–[19]. This line of work does not compensate for the delays; instead, it mitigates the instability issues arising from delayed force feedback. The operator still experiences delayed feedback and must move cautiously. To account for potential operator errors, impedance controllers are typically used by the robot [6], [20].

The challenge of delays in *visual* feedback during direct teleoperation is currently addressed through the use of *predictive displays* [21]–[24]. These displays present a non-delayed simulation of the robot and its environment. However, due to the inherent limitations of 3D modeling, disparities between the actual environment and the simulated one are unavoidable. This is particularly critical in unstructured scenarios, such as remote planets or damaged buildings, where mission-specific models are typically unavailable. Another related approach is the use of *predictor displays*, where the predicted trajectory of the system is overlaid on the visual feedback, simulated or not. This enables the operator to anticipate and better cope with delays [19], but human operators are charged with high cognitive load and can only tolerate moderate delays.

III. Prescient Teleoperation

Fig. 2 describes the overall system. We assume that the robot is controlled by our whole-body controller based on quadratic programming (see [2] or Appendix I-B) that takes as inputs a reference posture and the Cartesian positions of the hands, feet, etc. In prescient teleoperation, the primary concern revolves around determining how Cartesian and joint commands should be executed by the robot in order to effectively compensate for the delays.

A. Delay Estimation

To determine the amount of anticipation required for the commands, the robot needs to calculate the round-trip delay, which consists of a forward delay $\tau_f(t)$ in the communication from the operator to the robot and a backward delay $\tau_b(t)$ between the robot and the operator:

$$\tau(t) = \tau_f(t) + \tau_b(t)$$

Each one-way delay consists of two components: a deterministic component primarily caused by transmission and propagation time, and a stochastic component [25], often referred to as "jitter". We denote by $\tau_{f,D}$ the deterministic part of τ_f and by $\tau_{f,S}$ the stochastic part:

$$\tau_f(t) = \tau_{f,D} + \tau_{f,S} \quad (1)$$

$$\tau_b(t) = \tau_{b,D} + \tau_{b,S} \quad (2)$$

The robot computes the forward delay, including both the deterministic and stochastic components, by utilizing time stamps attached to the packets sent by the operator and clocks synchronized through NTP [26]. While the robot cannot predict the exact time required for a packet to reach the operator before transmission, it can obtain the average backward delay by querying the operator's station. However, it remains unaware of the stochastic portion of the backward delay until the packet is sent. To overcome this challenge, we adopt an upper-bound approach: video streaming systems incorporate a "jitter buffer" [27], [28] that accumulates, reorders, and, if necessary, discards video packets within a specific time window. When the buffer is set to a constant size, it effectively converts the stochastic delay into a deterministic upper bound delay. As a result, the robot only needs to be aware of the jitter buffer size and the deterministic backward delay to determine when the current frame will be displayed on the operator's computer.

B. Delay Compensation with Predictor Anticipation

To operate with delays, the system predicts the most likely future trajectories at each time step based on the commands received from the operator so far (for instance, the 3D trajectories of the hands). More formally, given a predictor of future trajectories $P(\cdot)$, the commands sent up to time step t over a history of length n, and a retargeting function[1] $R(\cdot)$ that transforms the operator inputs x to robot inputs y (e.g., apply scaling factors), the predictor $P(\cdot)$ predicts the next k time steps:

$$y_{t+1}, \cdots, y_{t+k} = P(R(x_{t-n}), \cdots, R(x_t)) \quad (3)$$

At time-step t, the robot executes the command q_t that corresponds to the future input at time-step $\lfloor t + \tau_f + \hat{\tau}_b \rceil$ using is whole-body controller $WB(\cdot)$:

$$q_t = WB(y_{\lfloor t+\tau_f+\hat{\tau}_b \rceil}) \quad (4)$$

where $\lfloor \tau \rceil$ denotes the nearest integer.

Numerous generic machine learning techniques have been proposed to predict the future of time-series, especially with neural networks [29], [30]. Nevertheless, the robotics community has long focused on regression techniques based on the concept of motion primitives, which are particularly suited for robot trajectories. Our system is based on Probabilistic Movement Primitives (ProMPs) [31]. They represent trajectories as probability distributions, making them well-suited for capturing the variability observed in human demonstrations as motion primitives. Another advantage of working with distributions is that the properties of motion primitives can be translated into operations from probability theory [31]. In particular, our system utilizes the conditioning operator of ProMPs to adapt predictions based on incoming observations. This allows us to update the prediction, referred to as the posterior, for the ongoing movement at each time step, using the learned model of the associated primitive as the prior. In other words, ProMPs predict the mean trajectory of prior demonstrations when no conditioning data is available, but when data, e.g., as observations of the current movement, is present, they update the prediction to resemble the most likely trajectories from the training set.

At time-step t, a point $\boldsymbol{\xi}_t$ of a single trajectory is computed using a weight vector $\boldsymbol{w} \in \mathbb{R}^m$ and a vector of Gaussian basis functions Φ, assuming a trajectory noise variance $\boldsymbol{\epsilon}_\xi$:

$$\boldsymbol{\xi}_t = \boldsymbol{\Phi}_t \boldsymbol{w} + \boldsymbol{\epsilon}_\xi \quad (5)$$

where the vector $\boldsymbol{\Phi}_t \in \mathbb{R}^m$ corresponds to the m normalized radial basis functions evaluated at time t (see [31] or Appendix IV-A). Let us denote by $\boldsymbol{\Sigma}_\xi$ is the observation noise variance. The probability of observing a trajectory $\boldsymbol{y}$ given the weight vector $\boldsymbol{w}$ is:

$$p(\boldsymbol{y}|\boldsymbol{w}) = \prod_t \mathcal{N}(\boldsymbol{\xi}_t|\boldsymbol{\Phi}_t\boldsymbol{w}, \boldsymbol{\Sigma}_\xi), \quad (6)$$

To learn a ProMP for a given task (a set of demonstrations), we fit each demonstration (i.e., find $\boldsymbol{w}$) to this representation using ridge regression [31]. Then, we fit the distribution of weight vectors $p(\boldsymbol{w})$, assuming they follow a normal distribution $\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}})$.

The trajectory distribution $p(\boldsymbol{y})$ is obtained by marginalizing out the weight vector $\boldsymbol{w}$, i.e.

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}. \quad (7)$$

Since a whole-body trajectory is represented by N trajectories (x, y, z position of the center of mass, of the hands, etc.), we learn a ProMP for each of the N trajectories. These ProMPs all together encode a task, which means that each task is associated with a set of ProMPs. More detail on our implementation is reported in Appendix IV-A to IV-F.

To enable online prediction, ProMPs are learned from demonstrations for each task (e.g., one to take a box, one to release it, one to push a bottle, etc.). During the offline training phase, an operator teleoperates the robot in a local network, which we consider to be an approximation of an ideal network without any delay. For each motion, we record the trajectories of the center of mass ground projection, the waist height, the hand Cartesian positions, the arms posture (shoulder rotation, elbow flexion, forearm rotation), the neck posture (flexion and rotation) and the torso posture (flexion, rotation and abduction). A ProMP is learned for each of these trajectories (examples in Appendix: Fig. S2-S4). Hence, the robot knows a set of ProMPs for each task.

[1] In this work, we predict the retargeted values, but we could have equivalently predicted the input value and retargeted the prediction.

TABLE I

DATASETS USED TO TRAIN AND TEST PRESCIENT TELEOPERATION.

Dataset Multiple Tasks			
	Bottle reaching (Fig. S1)		
	TOTAL	Bottle on table	Bottle on box
#training	12	6	6
#testing	20	10	10
	Box pick & place (Fig. S2)		
	TOTAL	Picking	Placing
#training	42	18	24
#testing	21	9	12
Dataset Obstacles — Bottle on table with different obstacles (Fig. S3)			
#training	6		
#testing	9		
Dataset Goals — Bottle on table at different positions (Fig. S4)			
#training	7		
#testing	10		

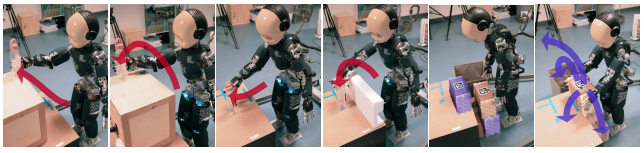

Fig. 3. **Tasks performed by the robot during the experiments (dataset "Multiple tasks").** In the first scenario (4 left images), the robot is teleoperated to reach a bottle at different locations and in different ways; in the second scenario (2 right images), the robot has to pick up a box from different locations and then placing it in another location.

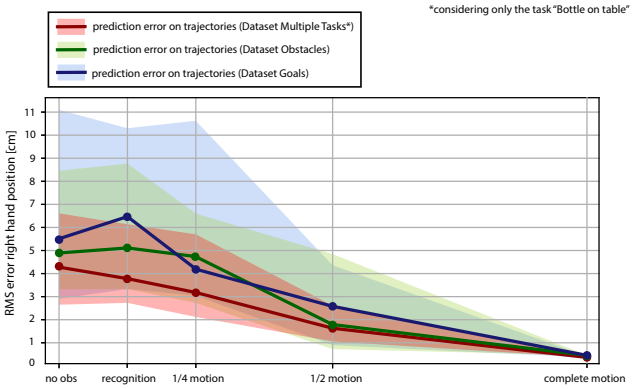

Fig. 4. **Comparison of the prediction error on different datasets.** The RMS of the error is computed based on the 10 testing trajectories of the task of reaching the bottle on the table from Fig. S1 (dataset Multiple Tasks), the testing trajectories from the dataset Obstacles (Fig. S3), and the dataset Goals (Fig. S4). The bold line represents the mean RMS error. The transparent region around it represents the maximum and minimum of the error. The prediction error is computed as the Euclidean distance between the predicted trajectory and the reference trajectory. The plot reports the error given by the mean trajectories of the ProMPs learned from the demonstrations, from the prediction updated after observing the first portion of motion used to infer the task, after observing a fourth of the motion, after observing half of the motion, and after observing the whole motion.

For inference (prediction), we identify the task $\hat{k}$ by computing the distance to the mean of each ProMP [32]:

$$\hat{k} = \arg\min_{k\in[1:K]} \left[\sum_{n=1}^{N} \sum_{t\in T_{obs}} |\boldsymbol{y}_n(t-\tau_f(t)) - \Phi_{n,t-\tau_f(t)} \boldsymbol{\mu}_{n,\boldsymbol{w}_k}| \right], \tag{8}$$

where K is the number of tasks in the dataset and $T_{obs} = \{t_1, ..., t_{n_{obs}}\}$ is the set of timesteps associated to the n_{obs} early observations. To account for different motion speeds, we additionally search for the best time-modulation by generating variants of each ProMP with different time-modulation values (see [32] or Appendix IV-D). This enables us to also predict the most likely duration of the current motion for the predicted task.

To predict the remaining of the trajectory, we condition the ProMP on the past trajectory using Bayes' theorem [31], [32], which "fine-tunes" the prediction to the actual data (instead of simply following the mean of the ProMP):

$$p(\boldsymbol{w}_{\hat{k}}|\boldsymbol{x}_t^*) \propto \mathcal{N}(\boldsymbol{y}_t^*|\boldsymbol{\Phi}_{\hat{\alpha}t}\boldsymbol{w}_{\hat{k}}, \boldsymbol{\Sigma}_y^*) p(\boldsymbol{w}_{\hat{k}}). \tag{9}$$

Detail on the operations can be found in the literature [31], [32] and in the Appendices IV-A to IV-G.

When the robot has not received enough data to recognize the ProMP, it executes the delayed command. Once the ProMP is recognized, the robot transitions from delayed commands to compensated commands using a blending technique similar to that used in shared autonomy research [33] (detail in Appendix IV-F and Fig. S6).

IV. EXPERIMENTS

We implemented this system with the humanoid robot iCub [34], which has 32 degrees of freedom (excluding the hands and eyes) and is position-controlled. The whole-body motion of the operator is measured with a motion capture suit (Xsens MVN), which provides both joint angles and Cartesian positions of the operator's body parts.

At each time-step (100 Hz), the robot uses its whole-body controller [2]. This controller solves a hierarchical constrained quadratic problem with inequality and equality constraints [35], aiming to minimize the tracking error, i.e. the deviation from the desired references. It considers priorities and constraints such as the kinematic model, joint velocities, and zero moment point bounds. The parameters and structure of the whole-body controller were optimized in our previous work using a multi-objective stochastic algorithm (see [2] or Appendix I-B).

We considered several teleoperation scenarios, which resulted in three different experiments and datasets (Table I), each divided into training and testing sets (Appendix II). All the datasets are variants of two tasks: reaching a bottle (scenario 1, Fig. 3, Fig. S1) and manipulating a box with both hands (scenario 2, Fig. S2). The test trajectories of the first dataset consist of various repetitions of these tasks, capturing the inherent movement variability of the operator. The second dataset, called Obstacles (Fig. S3), features the same scenarios as in the first dataset, but the test trajectories involve different ways of reaching a bottle while avoiding obstacles. These obstacles were not considered during training, allowing the robot to adapt its task execution to different environments. The third dataset, called Goals (Fig. S4), focuses on reaching a bottle placed at different positions. The test trajectories correspond to positions that were not used for training.

To facilitate result reproducibility and minimize time-consuming experiments, we used the real robot only for the

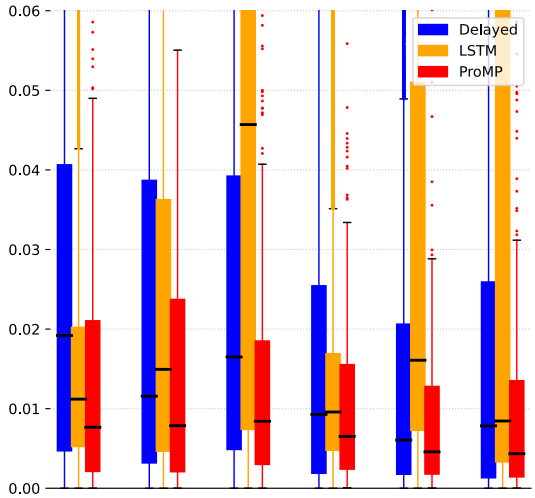

Fig. 5. Average difference between the point predicted 2-seconds ahead (200 time-steps) and the ground truth, for each degree coordinate of each hand, for each trajectory of the dataset "Multiple tasks". The box extends from the lower to upper quartile values of the data, with a line at the median. For the LSTM, the 10 replicates of the learning process (with different seeds) are considered independent data (i.e., the variance of the prediction error comes both from the different trajectories and the different seeds). The "Delayed" trajectory corresponds to the original trajectory delayed by 1 second, that is, to what the robot would have done without any prediction system. All comparisons are statistically significant ($p < 10^{-6}$, Mann-Whitney U-test).

Multiple Tasks dataset (Fig. 3). For the other two datasets, we utilized an accurate dynamic simulation. It is important to note that the real and simulated robots employ the same whole-body controller and follow the same joint trajectories. These trajectories are "played" in an open-loop manner, whether in simulation or on the robot.

Exp. 1. Evaluation of the predicted trajectories

Using testing data from the Multiple Tasks dataset, we initially evaluated the predictive capabilities of conditioned ProMPs in forecasting the future motion of the operator, independently of any consideration for teleoperation (Fig. 4). The results demonstrate that the prediction error decreases over time as more observations become available to update the prediction. For Cartesian trajectories (e.g., hands), the error reduces to approximately 0.5 cm after observing half of the motion (see Table S1). Similarly, notable improvement is observed in the postural trajectories. These results highlight that continuously updating the prediction allows the operator to influence the predicted motion so that it matches better their intentions. Put differently, the system is not simply recognizing the motions and then executing the mean of the learned demonstrations: it is accurately predicting the future trajectories given the data received so far and is adapting to a specific trajectory. We obtained similar errors with the other datasets (Fig. 4).

To further assess the quality of the predictions, we compared the ProMPs to traditional LSTM (Fig. 5 and appendix IV-H), a widely used method for time series forecasting [30]. The results show that the ProMPs produce significantly better predictions than the LSTM baseline for both 1-second ahead and 2-seconds ahead predictions (Fig 5, Fig. S7-S8). In addition, the difference in performance between ProMPs and LSTM becomes more prominent when the prediction horizon is increased: while the prediction quality of the LSTM substantially decreases when predicting 2-seconds compared to 1 second, the ProMPs keep almost the same quality (Fig. S7-S8). Overall, ProMPs demonstrate their effectiveness as predictors for long-term (in this case, a few seconds) predictions in our datasets.

Exp. 2. Prescient teleoperation experiments

We conducted evaluations of the system on the iCub robot, considering a mean time-varying round-trip delay of 1.5 seconds. This delay consisted of a stochastic forward delay, following a normal distribution with a mean of 750ms and a standard deviation of 100ms, and a constant backward delay of 750ms (Fig. 6).

To assess the quality of compensation, we compared the compensated trajectory to the non-delayed trajectory for 20 testing motions in the bottle-reaching scenario of the Multiple Tasks dataset, as well as for 21 testing motions in the box handling scenario of the same dataset (see Table S2). In the box handling scenario, the results demonstrate that the error for all considered references (especially the hands) is approximately 1cm or less with compensation, whereas, without compensation, the error is roughly three times higher (around 3cm for the hands). Similarly, in the bottle reaching task, the compensated trajectory exhibits an error of approximately 1 to 1.4cm for the hands, compared to an error of about 4cm for the hands and 1cm for the center of mass when no compensation is applied. The angular errors exhibit a similar trend. While an error of around 1cm is often acceptable to accomplish a task, such as grasping an object, an error of 3 to 4cm significantly increases the likelihood of missing the object, in addition to causing the operator frustration and disorientation.

We also evaluated the performance of the compensation as the communication delay increases (Fig. 7). To do this, we compared the compensated trajectory to the non-delayed trajectory, for the right hand, in the task of reaching the bottle on the table of the dataset Multiple Tasks (Fig. S1). During the synchronization, the error is roughly proportional to the delay (Fig. S6), which adds up to the prediction errors. In this case, we observed a mean error of approximately 2.5cm for a 1-second delay, but over 10cm for a 3-second delay, primarily because the transition time constitutes a significant portion of a short trajectory (30% for a 3-second delay and a 10-second trajectory). Excluding the synchronization time, the results reveal that the compensated tracking error is below 2cm for delays around 0.5s, approximately 2.5cm for delays around 1.5s, and increases to about 5cm for delays of 3s and 4s. Qualitatively, the system becomes challenging to use for the operator with a delay exceeding 2 seconds, resulting in an average error of approximately 3cm after the transition and approximately 7cm when considering the transition. We recorded similar tracking errors with the other datasets (Fig. 7b).

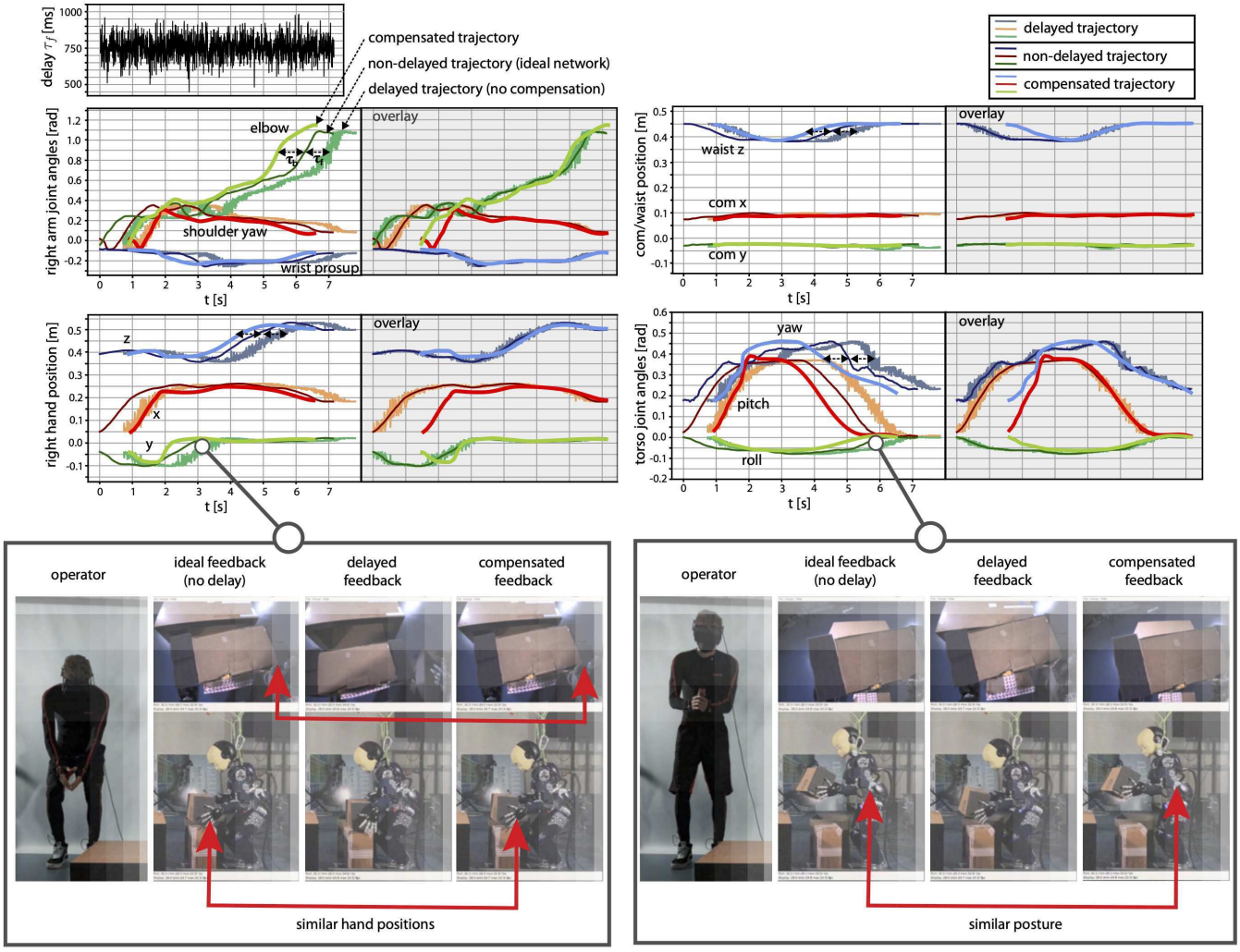

Fig. 6. **Teleoperation with compensation of a mean round-trip delay of 1.5s.** The robot is picking up a box in front of it at mid-height. The forward delay follows a normal distribution with 750ms as mean and 100ms as standard deviation, while the backward delay is 750ms (top). After the synchronization time (first 2 seconds), the compensated trajectory anticipates the delayed trajectory by $\tau_b + \tau_f$ to take into account both the delay from the operator to the robot and from the robot to the operator, this is why the robot anticipates *more* than the non-delayed trajectory. Once we shift all the trajectories to compare them ("overlay"), the commands (delayed trajectories) and the compensated trajectories align very well, which shows that the robot executes the right trajectory but shifted in time to anticipate the motion.

Exp. 3. Conforming the teleoperation to the intended motion

Humans can often perform a task in many ways because they are highly redundant. For example, when picking up a box, an operator can choose to bend their back without utilizing their legs, bend their legs while minimizing back movements, or adopt any combination in between. Ideally, the robot should execute tasks in the preferred manner of the operator. To explicitly explore this motion adaptation, we teleoperated the robot in simulation to reach a bottle placed on a table using three different approaches during the training phase (Fig. S3a, dataset "Obstacles"). During testing, we assessed the effectiveness of our compensation approach for the same task but with different obstacles present (Fig. S3b). As with the previous dataset, the prediction steadily improves as more observations become available, increasingly aligning with the observed operator commands. (Fig. S3c). The results show that the position error is comparable (if higher, only by about half a centimeter) to the error from the bottle-reaching scenario, where the same obstacles were used for both training and testing.

Exp. 4. Conforming the teleoperation to new goals

In many scenarios, the operator may need to perform motions that the system was not specifically trained for. To investigate this challenge, we assessed how the robot adapts to new object locations. For this purpose, we utilized the third dataset (dataset "Goals") and teleoperated the robot in simulation to reach a bottle positioned on a table at various locations (Fig. S4a). We then tested the approach by reaching the same bottle but at different positions (Fig. S4b). As one would expect, the prediction is less accurate compared to the previous datasets where the goal position remains constant. However, the error decreases to approximately 1cm after observing half of the motion (Fig. 4). When teleoperating

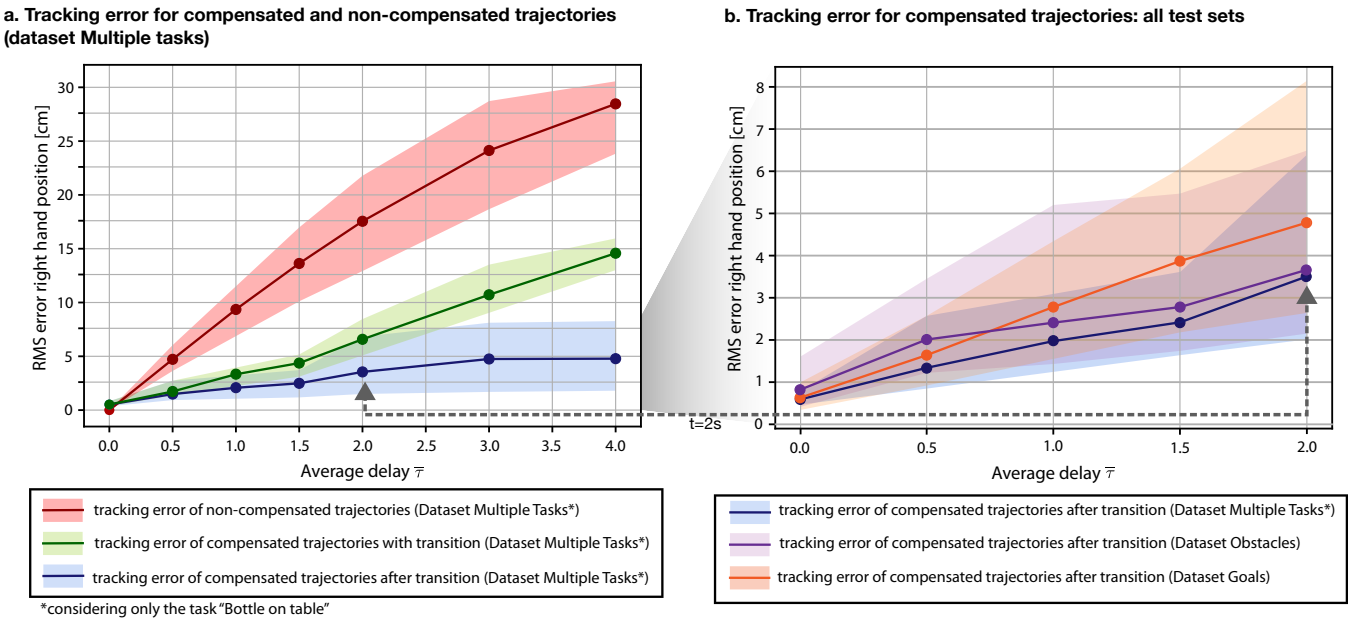

Fig. 7. **Scalability of the delay compensation with respect to increasing time delays. a. Tracking error of the compensated trajectories for the right-hand position with respect to the non-delayed ones compared to the tracking error of the corresponding non-compensated (delayed) trajectories with respect to the same non-delayed ones.** The tracking error of the compensated trajectories is considered both including the transition from the delayed phase to the synchronization phase (Fig. S6), which adds a non-compensable error, and without transition. The RMS of the error is computed from the 10 testing motions of the task of reaching the bottle on the table from dataset Multiple Tasks (Fig. S1) and its mean value is reported as a bold line. The transparent region around it, represents the maximum and minimum of the error. The tracking error is computed as the Euclidean distance between the evaluated trajectory and the reference trajectory. The compensated trajectories are temporally realigned with the non-delayed trajectories for computing the error, which is evaluated with different round-trip delays $\tau(t)$: 0s, around 0.5s, 1s, 1.5s, 2s, 3s and 4s. The time-varying forward delay follows a normal distribution with mean $\overline{\tau_f} = \overline{\tau}/2$ and standard deviation equal to $\frac{2}{15}\overline{\tau}_f$. The backward delay is set equal to $\overline{\tau}_f$. **b. Tracking error of the compensated trajectories for the right-hand position with respect to the non-delayed ones (after the transition phase) for all the datasets.**

the robot with a mean delay of 1.5s, the average tracking error on the hand position is 4cm (Fig. 7b), which is approximately a centimeter higher than the error with a 2-second delay in datasets where the bottle remains in the same position for both training and testing. The operator found it challenging to teleoperate the robot with such compensation errors. However, the approach can still be used effectively with novel goals for lower delays, typically between 0.5s and 1s, where similar accuracy to the other datasets is often achieved (Fig. 7B).

V. Conclusion and Discussion

Whole-body teleoperation offers an intuitive and flexible approach to operate humanoid robots exploiting their entire body, as long as the operator can rely on synchronized feedback. Through our use of machine learning to anticipate operator commands, we have demonstrated the ability to compensate for delays ranging from 1 to 2 seconds, which are typically observed in round-trip communication times between Earth and space [6] and across continents on the Internet [36].

While we utilized ProMPs as our chosen method for predicting future motions, other data-driven techniques for trajectory or time-series prediction, such as LSTM [37]–[40], could also be employed. However, these alternatives may require more training data and could potentially result in less smooth movements (see Fig. S7 and S8). In essence, the concept of *prescient teleoperation* can be implemented using any predictor, including neural networks, and the system's performance will improve with more accurate and advanced predictors.

All the experiments described in this study were conducted by an experienced user who is familiar with the teleoperation system and iCub. While novice users may not possess the same level of proficiency with the proposed system, it is unlikely that they would be entrusted with operating an expensive humanoid robot in a high-stakes mission, such as intervention in a damaged chemical plant or a remote lunar base. Just as drone pilots undergo rigorous training before their first mission, we expect that future humanoid operators would undergo extensive training as well. For this reason, in this work, we did not compare novice and expert user performance and we did not conduct a user study on usability in general. In future work, we will conduct user studies to examine the impact of prediction and delay compensation on human cognition, behavior, and performance, to explore the requirements and limits of how experienced users can effectively utilize a prescient teleoperation system in real-world scenarios.

Appendix

The appendices are available online at `https://hucebot.github.io/www_prescient_humanoids/`. They include more detail on the methods and additional figures.

REFERENCES

[1] L. Penco *et al.*, "Robust real-time whole-body motion retargeting from human to humanoid," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2018, pp. 425–432.

[2] L. Penco, E. M. Hoffman, V. Modugno, W. Gomes, J. Mouret, and S. Ivaldi, "Learning robust task priorities and gains for control of redundant robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2626–2633, 2020.

[3] M. Zucker, S. Joo, M. X. Grey, C. Rasmussen, E. Huang, M. Stilman, and A. Bobick, "A general-purpose system for teleoperation of the drc-hubo humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 336–351, 2015.

[4] S. Dafarra, K. Darvish, R. Grieco, G. Milani, U. Pattacini, L. Rapetti, G. Romualdi, M. Salvi, A. Scalzo, I. Sorrentino, *et al.*, "icub3 avatar system," *arXiv preprint arXiv:2203.06972*, 2022.

[5] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," *IEEE Transactions on Robotics*, 2023.

[6] M. Panzirsch *et al.*, "Exploring planet geology through force-feedback telemanipulation from orbit," *Science Robotics*, vol. 7, no. 65, p. eabl6307, 2022.

[7] D. M. Boroson, B. S. Robinson, D. V. Murphy, D. A. Burianek, F. Khatri, J. M. Kovalik, Z. Sodnik, and D. M. Cornwell, "Overview and results of the lunar laser communication demonstration," in *Free-Space Laser Communication and Atmospheric Propagation XXVI*, vol. 8971. SPIE, 2014, pp. 213–223.

[8] C. Atkeson *et al.*, *What Happened at the DARPA Robotics Challenge Finals*. Springer Tracts in Advanced Robotics, 2018, pp. 667–684.

[9] "National aeronautics and space administration (nasa)-centennial challenges program-space robotics challenge phase 2," https://t.ly/t10-t.

[10] W. R. Ferrell, "Remote manipulation with transmission delay," *IEEE Trans. on Human Factors in Electron.*, vol. 6, no. 1, pp. 24–32, 1965.

[11] W. R. Ferrell and T. B. Sheridan, "Supervisory control of remote manipulation," *IEEE Spectrum*, vol. 4, no. 10, pp. 81–88, 1967.

[12] M. Stilman, K. Nishiwaki, and S. Kagami, "Humanoid teleoperation for whole body manipulation," in *IEEE International Conference on Robotics and Automation*, 2008.

[13] B. Omarali, B. Denoun, K. Althoefer, L. Jamone, M. Valle, and I. Farkhatdinov, "Virtual reality based telerobotics framework with depth cameras," in *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020.

[14] A. Naceri, D. Mazzanti, J. Bimbo, Y. T. Tefera, D. Prattichizzo, D. G. Caldwell, L. S. Mattos, and N. Deshpande, "The Vicarios virtual reality interface for remote robotic teleoperation," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 4, pp. 1–16, 2021.

[15] J.-H. Ryu, D.-S. Kwon, and B. Hannaford, "Stable teleoperation with time-domain passivity control," *IEEE Transactions on robotics and automation*, vol. 20, no. 2, pp. 365–373, 2004.

[16] M. Panzirsch, H. Singh, T. Krüger, C. Ott, and A. Albu-Schäffer, "Safe interactions and kinesthetic feedback in high performance Earth-to-Moon teleoperation," in *IEEE Aerospace Conference*, 2020.

[17] J. Artigas, R. Balachandran, C. Riecke, M. Stelzer, B. Weber, J.-H. Ryu, and A. Albu-Schaeffer, "Kontur-2: force-feedback teleoperation from the international space station," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1166–1173.

[18] N. Lii *et al.*, "The robot as an avatar or co-worker? an investigation of the different teleoperation modalities through the KONTUR-2 and METERON SUPVIS Justin space telerobotic missions," in *Int. Astronautical Cong.*, 2018.

[19] T. B. Sheridan, "Space teleoperation through time delay: Review and prognosis," *IEEE Trans. on Robotics and Automation*, vol. 9, no. 5, pp. 592–606, 1993.

[20] P. Evrard, N. Mansard, O. Stasse, A. Kheddar, T. Schauß, C. Weber, A. Peer, and M. Buss, "Intercontinental, multimodal, wide-range tele-cooperation using a humanoid robot," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2009, pp. 5635–5640.

[21] L. Peñin, K. Matsumoto, and K. U. G. Kenkyūjo, *Teleoperation with Time Delay: A Survey and Its Use in Space Robotics*. National Aerospace Laboratory, 2002.

[22] M. Hernando and E. Gambao, *Advances in Telerobotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, ch. Teleprograming: Capturing the Intention of the Human Operator, pp. 303–320.

[23] A. K. Bejczy, W. S. Kim, and S. C. Venema, "The phantom robot: predictive displays for teleoperation with time delay," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1990.

[24] P. Mitra and G. Niemeyer, "Mediating time delayed teleoperation with user suggested models: Implications and comparative study," in *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2008, pp. 343–350.

[25] O. Gurewitz, I. Cidon, and M. Sidi, "One-way delay estimation using network-wide measurements," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2710–2724, 2006.

[26] D. Mills, J. Martin, J. Burbank, and W. Kasch, "Network time protocol version 4: Protocol and algorithms specification," Internet Requests for Comments, RFC 5905, June 2010.

[27] B. Oklander and M. Sidi, "Jitter buffer analysis," in *Proc. Int. Conf. on Comp. Comm. Networks, ICCCN*, 09 2008, pp. 1 – 6.

[28] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," Internet Requests for Comments, RFC Editor, STD 64, 2003.

[29] A. R. S. Parmezan, V. M. Souza, and G. E. Batista, "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model," *Information sciences*, vol. 484, pp. 302–337, 2019.

[30] B. Lim and S. Zohren, "Time series forecasting with deep learning: A survey," *Philosophical Transactions of the Royal Society A.*, vol. 379, pp. 20 200 209–20 200 209, 2021.

[31] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, p. 529–551, 07 2018.

[32] O. Dermy, A. Paraschos, M. Ewerton, J. Peters, F. Charpillet, and S. Ivaldi, "Prediction of intention during interaction with iCub with probabilistic movement primitives," *Frontiers in Robotics and AI*, vol. 4, pp. 45–45, 2017.

[33] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[34] L. Natale, C. Bartolozzi, D. Pucci, A. Wykowska, and G. Metta, "iCub: The not-yet-finished story of building a robot child," *Science Robotics*, vol. 2, no. 13, p. eaaq1026, 2017.

[35] A. Rocchi, E. M. Hoffman, D. G. Caldwell, and N. G. Tsagarakis, "Opensot: a whole-body control library for the compliant humanoid robot coman," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6248–6253.

[36] T. Høiland-Jørgensen, B. Ahlgren, P. Hurtig, and A. Brunstrom, "Measuring latency variation in the internet," in *Proc. Int. Conf. emerging Networking EXperim. and Techn.*, 2016, pp. 473–480.

[37] X. Zhao, S. Chumkamon, S. Duan, J. Rojas, and J. Pan, "Collaborative human-robot motion generation using LSTM-RNN," in *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2018, pp. 1–9.

[38] M. Anvaripour, M. Khoshnam, C. Menon, and M. Saif, "FMG- and RNN-Based estimation of motor intention of upper-limb motion in human-robot collaboration," *Frontiers in Robotics and AI*, vol. 7, pp. 183–183, 2020.

[39] E. Corona, A. Pumarola, G. Alenya, and F. Moreno-Noguer, "Context-aware human motion prediction," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 6992–7001.

[40] J. Bütepage, H. Kjellström, and D. Kragic, "Anticipating many futures: Online human motion prediction and generation for human-robot interaction," in *IEEE Int. Conf. Robotics and Automation*, 2018, pp. 4563–4570.