In: *Proceedings of the 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, Naples, Italy, Aug 31-Sep 4, 2020.
doi:10.1109/RO-MAN47096.2020.9223477

Model Mediated Teleoperation with a Hand-Arm Exoskeleton in Long Time Delays Using Reinforcement Learning

Hadi Beik-Mohammadi[1,2,*], Matthias Kerzel[2], Benedikt Pleintinger[1], Thomas Hulin[1], Philipp Reisich[1], Annika Schmidt[1,3], Aaron Pereira[1], Stefan Wermter[2] and Neal Y. Lii[1]

***Abstract*— Telerobotic systems must adapt to new environmental conditions and deal with high uncertainty caused by long-time delays. As one of the best alternatives to human-level intelligence, Reinforcement Learning (RL) may offer a solution to cope with these issues. This paper proposes to integrate RL with the Model Mediated Teleoperation (MMT) concept. The teleoperator interacts with a simulated virtual environment, which provides instant feedback. Whereas feedback from the real environment is delayed, feedback from the model is instantaneous, leading to high transparency. The MMT is realized in combination with an intelligent system with two layers. The first layer utilizes Dynamic Movement Primitives (DMP) which accounts for certain changes in the avatar environment. And, the second layer addresses the problems caused by uncertainty in the model using RL methods. Augmented reality was also provided to fuse the avatar device and virtual environment models for the teleoperator. Implemented on DLR's Exodex Adam hand-arm haptic exoskeleton, the results show RL methods are able to find different solutions when changes are applied to the object position after the demonstration. The results also show DMPs to be effective at adapting to new conditions where there is no uncertainty involved.**

I. INTRODUCTION

Teleoperation provides the possibility for an operator to interact with a remote environment using an intermediate device. The intermediate device consists of two interconnected parts so-called input and avatar. Using the input device, the teleoperator remotely controls/commands the avatar through a communication channel. Although the avatar is assumed to be passive, due to the bilateral control scheme, it affects the input device using position or force feedback. The bilateral control scheme provides feedback to the operator, which augments the remote environment to be perceived by the operator's senses, such as haptic, visual, and auditory. In a long-distance teleoperation scenario, data transmission can take a significant time delay to reach the teleoperator. This delay introduces inconsistency or mismatch between input and avatar system that falsifies the provided feedback. For example, the teleoperator may receive the haptic feedback well after the avatar robot has already collided with an object in the remote environment, causing damage to the target robot system, as well as the remote environment. Furthermore, the visual information provided to the operator may also have significant discrepancies. For example, the object may move in the span of the communication delay, which can cause the task to be compromised. Therefore, relying solely on the data collected from the remote environment is undesirable.

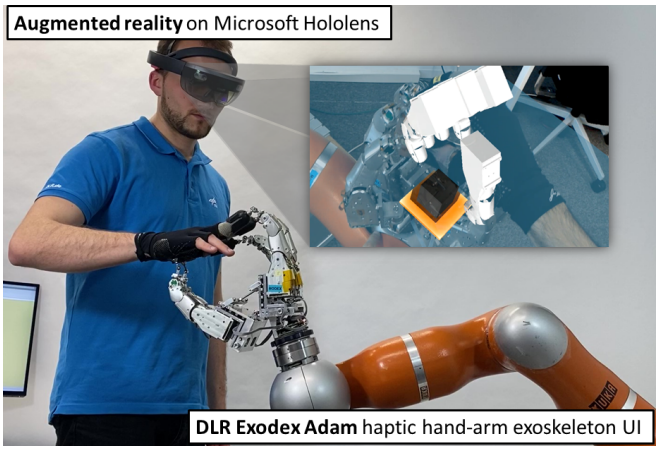

Fig. 1. A teleoperator using the proposed approach with the Exodex Adam hand-arm haptic interface and a Microsoft Hololens teaches the avatar robot to build a tower using three cubes. The teleoperator observes a replica of the remote environment and the avatar robot using the augmented reality.

Grasping or manipulation of objects by teleoperation can be realized in different ways. The robot configurations can be pre-recorded and used as a look-up table. The corresponding configurations can be selected based on the closest object position to the current position [1]. To reduce the inconsistency between configurations, end-to-end deep learning with pre-trained neural networks [2] can be used instead of hard-coded look-up tables. Furthermore, deep reinforcement learning can be used as an online solution where the agent learns to reach for grasp by interacting with the environment [3]–[5]. The idea of learning from demonstration (LfD) can increase the stability and performance of the grasping and in-hand manipulation [6], [7].

In practice, a teleoperation scenario may take place on different levels of abstraction, automation, and shared autonomy. Using a task-driven approach with gesture recognition, it has been demonstrated that in-hand manipulation can be effectively carried out for all six possible Degrees of Freedom (DOF), on free and partially constrained objects [8]. German Aerospace Center (DLR) and European Space Agency's (ESA) METERON SUPVIS Justin [9] was a teleoperation mission with high-level abstraction and complete task autonomy where an astronaut in orbit commanded a

[1] Institute of Robotics and Mechatronics, German Aerospace Center, Munich, Germany (Hadi.Beikmohammadi, Thomas.Hulin, Annika.Schmidt, Benedikt.Pleintinger, Philipp.Reisich, Aaron.Pereira, Neal.Lii)@DLR.de

[2] Knowledge Technology Institute, Dept of Informatics, Hamburg University, Hamburg, Germany (Kerzel, Wermter) @informatik.uni-hamnburg.de

[3] Technical University of Munich, Munich, Germany (An.Schmidt)@TUM.de

robot on the ground to execute a task. In this experiment, the task performance relied mainly on the avatar's built-in intelligence and the intuitiveness of the task representation for the teleoperator.

Teleoperators continue to demand immersive user experience with haptic feedback of force reflection to feel the object and the environment. Various work has been carried out to study force reflection in-hand telepresence, including an exoskeleton system that uses neural networks [10].

Furthermore, previous work [11] has evaluated and shown the effectiveness of haptic feedback for increasing the in-hand teleoperation performance, as compared to other feedback conditions. By considering the viscosity of the environment, a telepresence system can be extended to interact in a mixed media environment of fluid, gas, and solids up to the fingertips to realize an even more immersive user experience [12].

The bilateral teleoperation architecture usually contains 2-channel [13] or 4-channel [14] communication, which ensures the consistency between two models and the safety of the avatar. For example, if during a task an object in the remote environment hinders the avatar movement, instant force feedback can inform the operator to prevent colliding with an object or stop penetrating a hard surface. Instant force feedback requires high bandwidth communication channels with no time delay.

As mentioned, time delay can cause teleoperation to become unstable, particularly for high-DOF, long-delay systems. Kontur-2 [15] and METERON Haptics-2 experiments [16] have tackled this with 20 msec to 800 msec+ of time delays. In robot-assisted remote telesurgery, the operations are handled with delay as long as 100 msec+ [17].

Approaches such as passivity-based control [18]–[20] and predictive display [21], [22] can reduce the inconsistency, but the problems appear when the time delay increases or becomes variable. A unique way of approaching the time delay problem is introduced by Model Mediated Teleoperation (MMT) approach [23], [24]. The MMT approach uses a virtual replica of the remote environment on the input-side to provide instant feedback for the teleoperator and has been used in several areas, for example, surgical teleoperation [25] and space robotics [26].

MMT requires an accurate model of the remote environment and avatar device, but modeling a non-linear time-variant system like a remote environment is problematic. The MMT approaches attempt to recreate the remote environment model using different methods such as neural networks [27]. To tackle problems caused by the time delay, proving a form of timely visual and haptic feedback combined with adaptation to uncertainties is required. The avatar-side intelligence can be realized to compensate for errors and mistakes in the task demonstration; therefore, discarding the need for an accurate model of the remote environment on the input-side. Hence, providing instant haptic and visual feedback does not require the precise future state of the environment, and a simple approximation is sufficient.

Our approach adopts the architecture from the MMT scheme and combines it with Dynamic Movement Primitives (DMP) [28] and model-free Reinforcement Learning (RL) [6], [29], [30]. The avatar system uses DMPs to account for changes in the model by transforming the trajectory into non-linear dynamical systems. The reconstructed trajectory can be generated almost instantly. To improve the adaptation, the system should be able to adapt to uncertainties in the model that cannot be considered before trajectory execution. The policy search RL methods are integrated to explore and search for a viable solution in a limited time. Three RL methods are evaluated, Policy learning by Weighting Exploration with the Returns (PoWER) [30] based on expectation maximization, Episodic Natural Actor Critic (eNAC) [31] based on natural gradient, and Policy Improvement using Path Integrals (PI^2) [29] based on stochastic optimal control.

Various forms of feedback can be used to immerse the teleoperator in the remote environment. As Fig. 1 shows, our system offers haptic feedback via the haptic Exodex Adam interface [32], [33], in which all forces are calculated in the virtual environment. The teleoperator controls an anthropomorphic robot arm via the haptic interface, which is supposed to reproduce movements at the same time. Furthermore, a simulated robot farm using multiple instances of the avatar system and the remote environment is developed to facilitate the learning process. The rest of the paper is organized as follows: Sec. II presents the proposed RL enhanced MMT approach and architecture of the teleoperation system. Sec. III presents the design and the implantation of the necessary modules to provide a practical framework. In Sec. IV, the results are presented. In Sec. V a general discussion about the approach is provided. Finally, Sec. VI concludes the paper and lays out our future work.

II. RL Approach to MMT

The architecture of the proposed approach is designed in a modular way to facilitate the analysis and testing process. Fig. 2 illustrates the overall architecture of our MMT system. The avatar and the input side of the architecture are connected with a communication network that causes a delay in transmission.

A. Dynamic Movement Primitives (DMP)

Transferring knowledge between structurally different systems requires transformation and usually results in information loss. Similarly, in teleoperation, in most cases, avatar and input have fundamental differences that aggravate a lossless performance. Additionally, information can be lost or altered during the communication process. To evaluate the knowledge transfer and trajectory reconstruction, different properties are taken into account:

1) Quality of adaptation to new conditions
2) Dimensionality of the encoded space
3) Safe exploration and learning

DMPs can learn to reconstruct new trajectories with different temporal and spatial features using demonstration. Given the position $[x_0, x_1, ..., x_N]$, velocities $[\dot{x_0}, \dot{x_1}, ..., \dot{x_N}]$ and acceleration $[\ddot{x_0}, \ddot{x_1}, ..., \ddot{x_N}]$, DMP parameters θ can

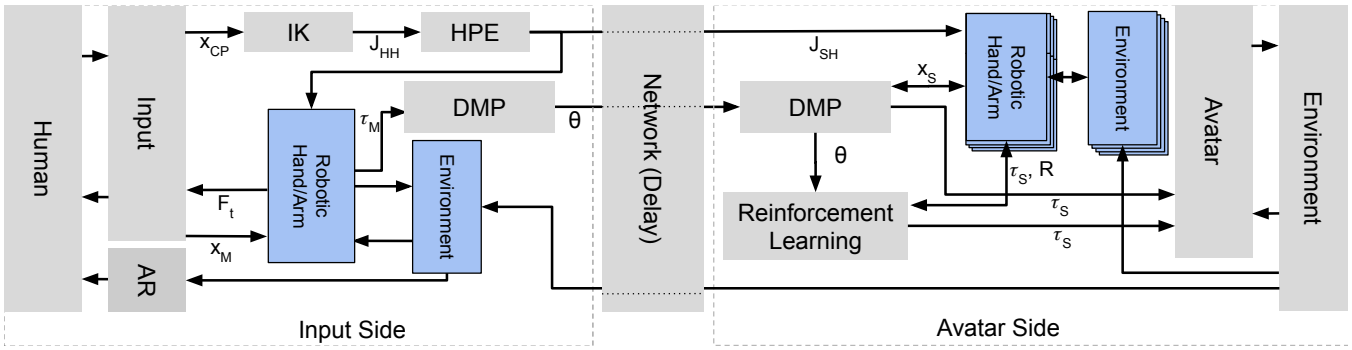

Fig. 2. The overall architecture of the RL enhanced MMT. The network (delay) splits the architecture into two main parts, the avatar and the input side. The *Human* teleoperator interacts with the *input* device. On the input side *LWR/Hand simulation* using physics simulation, *Environment* provides instant haptic feedback and also information for the Augmented Reality unit *AR*. The Inverse Kinematic *IK* block calculates the human hand posture given the contact points on the input device. Consequently, the avatar Hand Posture Estimator *HPE* approximates the corresponding avatar hand configuration. The *DMP* in the input side encodes the trajectory to basis functions while the *DMP* in the avatar side encodes them back, and evaluates and reconstructs the new trajectory based on the new environmental situation given by *Hand/LWR* and *Environment* simulator. In case of failure, the *Reinforcement Learning* unit improves the trajectory and finally deploys it on the *avatar* robot in the remote *Environment*.

be generated using one-shot learning. Each point on the trajectory has six elements, three for position and three for orientation. Since each DMP encodes one dimension, six DMPs are used to encode each demonstrated trajectory. The roll, pitch, and yaw of the end-effector are then converted to a rotation matrix by a set of transformations where the singularities are avoided. The DLR Five-Finger Hand (FFH), which is used as the avatar end-effector, is not considered in the DMP transformation due to the high number of DOF (21 DOF), which is outside of the scope of this paper.

As Fig. 2 shows, the avatar robot receives the final trajectory from two sources, DMP, and RL units. The DMP in the input side encodes the trajectory to a set of parameters and sends it to the avatar side where another DMP block decodes and rebuilds the trajectory given the updated values from the avatar robot and the environment.

Before executing the trajectory on the real robot, a simulation is used to evaluate the trajectory. This paper mainly focuses on grasping different geometrical objects (e.g. centric, parallel, arbitrary); the evaluation conditions are designed correspondingly. If the reconstructed trajectory leads to a successful grasp in simulation, thereafter, the real avatar robot deploys the trajectory. Otherwise, the DMP unit activates the RL to adapt and improve the trajectory until a successful grasp in simulation is achieved.

B. Reinforcement Learning

Although DMPs can adapt an old trajectory to new conditions, they may fail due to reasons such as the approaching angle for grasping with an asymmetric end-effector structure (e.g. anthropomorphic hand) while for a manipulator with a symmetric end-effector the approach angle does not make any difference. Furthermore, uncertainty in the object position is a common problem and causes collisions and inconsistency in grasping and eventually, failure of the task execution. Hence, a new demonstration by the teleoperator might be costly and also result in the same problem; therefore, our architecture detects such a failure before execution in the real environment and adapts the same trajectory to the new conditions using different reinforcement learning methods. This work presents the comparison between fast and robust algorithms such as Policy Improvement using Path Integrals (PI^2), Policy Learning by Weighting Exploration with the Returns (PoWER) and Episodic Natural Actor-Critic (eNAC).

PI^2 and PoWER are policy perturbation methods that explore the parameter space $\pi_{\theta+\epsilon}$ to generate trajectories that are similar to the demonstration but have slightly different features. These exploratory trajectories are essential for RL to visit new parts of the task-space to find a local optimal solution. Fig. 3 shows a detailed view of the RL block in Fig. 2, which outlines the general structure of policy perturbation methods. Both algorithms have a similar exploration behavior, but the differences are in the calculation of parameter update $d\theta$. The parameter update in PI^2 is calculated in a way that provides more freedom for designing the cost function. Whereas the cost function in the PoWER algorithm has to be an improper probability distribution, therefore, the returns should always be positive.

The eNAC algorithm [31] is an actor-critic policy gradient which uses the natural gradient to find the steepest path to the optimal solution. It is called episodic since the critic is evaluated at the end of the trajectory execution, but the algorithm is step-based since the actions are perturbed each frame. The algorithm has several benefits like intuitive exploration rate and using natural gradient but perturbing actions has disadvantages [34] such as,

- Introducing independent random small movements to a

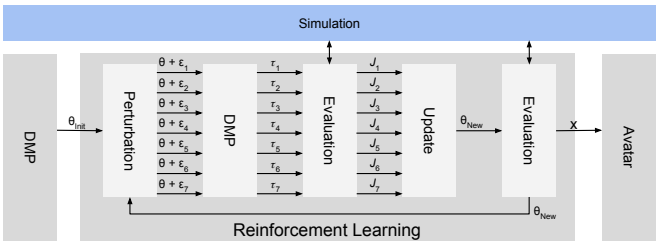

Fig. 3. The figure shows the general structure of policy parameter perturbation methods inspired from [6].

smooth trajectory increases the jerkiness,
- The system might behave as a low-pass filter and filter-out the high-frequency perturbations by averaging.
- Instant changes in robot motion without considering the inertia might damage the robot,
- The variance of the parameter vector might increase dangerously.

The details regarding the formulas and algorithms can be found in [29]–[31].

C. Learning under Uncertainty

Due to the uncertainty in object detection, the RL agent must be able to learn the new trajectory goal while optimizing the trajectory shape. The authors in [6] introduce an approach to extend the capability of PI^2 to adapt the target of the trajectory. The approach uses the cost of the whole trajectory since the goal does not change throughout the trajectory execution. The haptic feedback as secondary sensory feedback is used to compensate for the vision inaccuracy.

Formulation

1) Exploration Rate Σ: As PI^2 and PoWER approaches suggest, the perturbation step introduces exploration into the parameter space. The exploration may cause serious disruptions in the agent's behavior and unpredictable movements, which can adversely affect the avatar robot and the remote environment simultaneously. Due to the counter-intuitive definition of the parameters θ, which define the final shape of the trajectory by modifying the DMPs, an intuitive understanding of the applied noise (exploration) Σ is required before performing actions on the avatar. A high exploration rate in parameter perturbation methods may highly diverge the trajectory and cause physical damage to the robot.
As mentioned, the eNAC algorithm uses the action-space to explore the new policies; therefore, the exploration rate Σ is an intuitive measure of distance. A low exploration rate leads to a smoother trajectory, while a high exploration rate may cause damage to the robot by big instant changes. In parameter and action-space perturbation methods, a trade-off can be found that speeds up learning while keeping the avatar device and remote environment safe.
Furthermore, initially, the RL agent requires higher exploration to increase the chance to find a solution within the time limit. For that reason, when a solution has been found, the rate of the exploration must decrease to stabilize the convergence and finalize the trajectory shape. Therefore, the exploration rate Σ decreases at each time step using the following equation:

$$\gamma = max(\frac{\text{Update}_{\text{max}} - i}{\text{Update}_{\text{max}}}, 0.1). \quad (1)$$

$$\Sigma_i = \gamma \Sigma_{\text{init}}. \quad (2)$$

Where i is the current number of updates and $\text{Update}_{\text{max}}$ is the maximum number of updates. The decay rate γ is responsible for linearly decreasing the exploration rate. On the other hand, the exploration rate of the goal defines the perturbation of the target position. The learning exploration rate for each approach was defined experimentally in the simulation (see table I).

2) Learning Rate α: In the critic step of the eNAC algorithm, a learning rate is deployed to control the final influence of parameter updates on the final trajectory. A small learning rate leads to late convergence, while a high learning rate results in sudden changes in the trajectory and failure due to high divergence from the initial trajectory.

3) Cost Function: The cost function in policy search algorithms defines whether the agent should get a reward. It also defines the scale of the reward, depending on the quality of the action. A cost function maps several aspects of action in one single value with different proportions depending on the purpose of the learning. For example, the cost function can be a combination of:

- Acceleration or velocity of the end-effector
- Distance from the target
- The scale of the exploration noise
- The number of frames that fingertips are in proximity of an object
- The number of fingers involved in the grasp
- The object displacement during grasp

Different combinations of these parameters have a different interpretation, and they result in different trajectories. For example, integrating the acceleration into the cost function results in a less jerky trajectory [6]. Moreover, The simulation involves characteristics that are unrealistic, like continuous access to the precise object position. In a real environment, the object can be obscured and hard to detect and locate; therefore, using cost items like the object position is not plausible. As a final solution, two properties are experimentally selected; acceleration and noise scale for the control and trajectory cost. And for the final cost, the number of fingers that have touched the object is included. The cost function is defined as below [6],

$$J(\tau_i) = \Phi_{t_N} + \int_{t_i}^{t_N} (10^{-11}(\ddot{x_t}^2 + \frac{1}{2}\theta_t^T R \theta_t))dt. \quad (3)$$

where Φ is the final cost indicating the quality of the grasp. The quality of the grasp is determined by the number of fingers involved in the grasp $N_{Fingers}$,

$$\Phi(\tau_i) = 1 - \frac{N_{Fingers}}{5}. \quad (4)$$

The final cost Φ is zero when all fingers are involved in the grasp, but depending on the object size, the maximum number of fingers might differ.

TABLE I
THE TABLE SHOWS THE EXPLORATION RATE DEFINED FOR EACH APPROACH

	PI^2	PoWER	eNAC	Goal
Σ	300	300	0.01	0.04

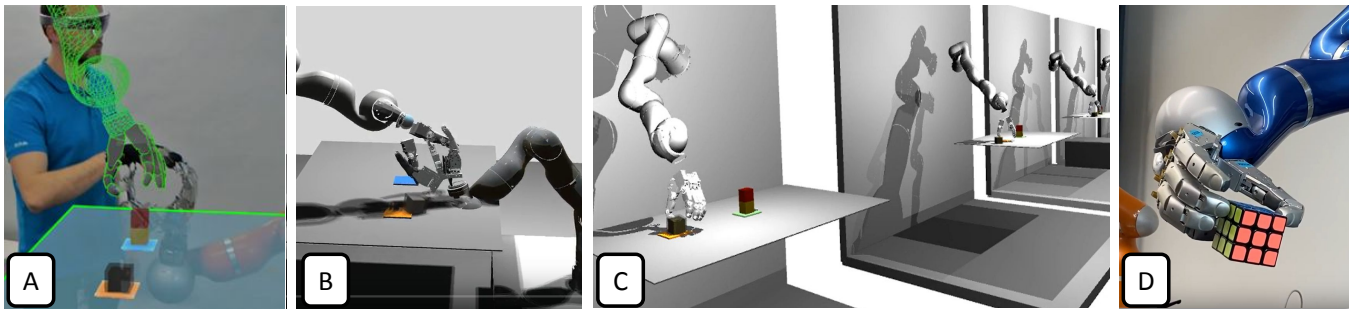

Fig. 4. The learning procedure for grasping a cube. In A, the teleoperator uses the Exodex Adam haptic interface to teach the avatar robot to grasp a cube. In B, a simulation of avatar robot and input setup processes all the physical properties of the environment like forces e.g. gravity and lets the human interact with objects in real-time. In C, multiple avatar units try different approaches to find a solution in case the environment has changed during data transmission. In D, finally, the evaluated trajectory in the simulation is used to grasp the object in the real avatar setup.

4) Roll-out and Update number: The number of roll-outs determines the number of perturbed trajectories which must be generated and evaluated for calculating the next update. The number of updates defines the maximum number of times that the policy parameters get updated. The higher number of roll-outs leads to robust learning but slower convergence [6]. Maximum 100 updates with seven roll-outs have been used in the experiments due to hardware limitations. To increase the learning stability, the best two trajectories are kept in the memory to be used in the next update.

III. Teleoperation System Implementation

To provide a framework to evaluate the system, different modules are designed and implemented. Fig. 1 and Fig. 4.A show the input device, DLR Exodex Adam, a novel haptic interface designed by the MODEX lab at German Aerospace Center [32], [33], [35]. This is paired with a Microsoft Hololens as the AR to provide a visualization of the virtual environment as an overlay on the operator's vision. The augmented 3D models of the table, objects, and robots are processed in Unity game engine[1] and visually anchored using Vofuria engine[2]. The avatar setup uses a robot arm and a customized version of the DLR Five-Finger Hand (FFH) [8]. The differences between The FFH and DLR hand is in the position of the thumb, which is opposing other fingers to facilitate grasping and in-hand manipulation [36]. The avatar robot arm is a custom-configured LWR 4+, with physical offset modifications on joints 1, 3, and 6, to facilitate the anthropomorphic configuration. The Links and Nodes (LN) library is used as middleware to manage the processes and the communication between all the components/agents.

High-level task demonstration, such as grasping in a teleoperation scenario, requires a highly compliant interface that provides high flexibility and weightless movement [37]. In addition to compliance, proper haptic feedback helps the teleoperator to understand the physical properties of the remote environment by interacting with the objects. To do so, a gravity compensation and a regular torque controller were deployed on the input device to provide compliance and flexibility on the arm and hand (fingers). On the avatar-side, the robot requires a position to follow the input; this position, called anchor, can be any point on the input device. For instance, the human hand palm position or the input's end-effector position would are proper options. The anchor is a virtual link that the avatar robot uses to follow the input movements.

A pipeline of Inverse Kinematic (IK) [37] and joint-to-joint mapping has been used to estimate the human hand posture using the avatar hand. The IK is used since there is no direct way to read human joint values. It utilizes the contact points calculated using forward kinematic on the input device. Then, the joint-to-joint mapping uses the human hand joint configuration to compute the avatar robot joint values using optimization. Since MMT extensively uses a simulation environment, therefore using a simulation with a proper physics engine is necessary. The chosen physics engine must have the ability to:

- Calculate penetrations on complex objects
- Simulate rigid body dynamic behavior

Unity is a cross-platform game engine with a powerful built-in physics engine. Although Unity is a powerful tool to simulate complicated objects with dynamic behavior but due to limited processing power, the capabilities are restricted. For example, Unity does not support non-convex rigid body collision detection due to computation overhead and low demand in the market.

Although Unity supports simulation for different tangential forces for instance friction, the positional noise coming from the input robot makes the surface friction impractical. Surface friction is necessary for grasping; this tangential force is the main reason that the object stays within the hand during the movements. In the simulation, due to the noise from the real input device, the surface friction fails because of constant attachment and detachment of the fingertips and the object surface. This fluctuation causes the object to slide out of the hand right after picking up off the ground. To solve this problem using Unity a new approach is proposed so-called the diaphragm. The diaphragm guarantees a smooth attachment without any contact issue. The diaphragm is a novel idea which uses an arbitrary virtual object surrounding the target object to adapt contact forces between the hand and the objects. It has the same shape as the object but 20 percent

[1]Unity game engine available at `https://unity.com`

[2]Vofuria engine available at `https://engine.vuforia.com/engine`

Fig. 5. The learning cost over 100 updates. Each update contains seven roll-outs performed in the simulation. The targets related to this plot were placed about 50 centimeters away from the demonstration. The figure shows PI^2 and PoWER has a very steep decrease in the cost of the trajectory while eNAC has a slower convergence and less stability with high fluctuations.

bigger in size with a transparent appearance. Once one of the robot fingers enter the diaphragm, the physics engine takes over the low-level control. Although the diaphragm facilitates the grasping, still due to high noise in object position, it may slip out of the hand, to address this issue an external controller is deployed on the object position. This external controller makes the grasp more persistent by using a kinematic control and eliminating all the forces, including gravity. The depth and normal vector of the penetration calculated in simulation are sent to the input robot, where the god object approach [38] is used to provide haptic feedback for the operator.

IV. Results

The first set of results compare different algorithms in a scenario where the avatar robot grasps a box. Fig. 5 compares the cost of trajectories generated by different methods in a condition where the object is located almost 40 centimeters away from the demonstration in the XY plane (table top). The initial cost relates to the trajectory generated by DMPs, and later, each algorithm attempts to reduce costs within a limited time. The figure shows that all methods successfully decrease the cost and stabilize the learning with a low cost where the grasping happens. The PI^2 and PoWER rapidly decrease the cost, but PI^2 shows higher stability by less fluctuation while reaching a lower value in the steady-state. The eNAC algorithm has a gradual decrease and unusual fluctuations in the steady-state. Moreover, eNAC's steady-state shows a higher final cost than the other approaches.

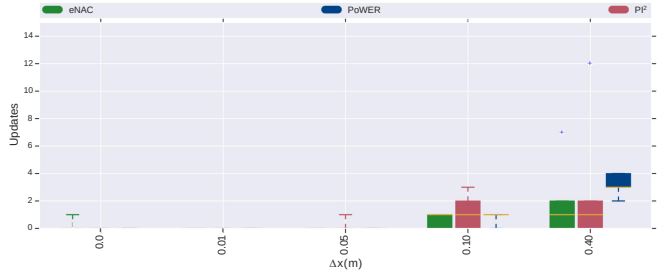

Fig. 6. The update number required for different approaches to find a solution. The horizontal axis indicates the distance from the demonstrated target, and the vertical axis shows the number of updates. The positional differences are applied to the X direction.

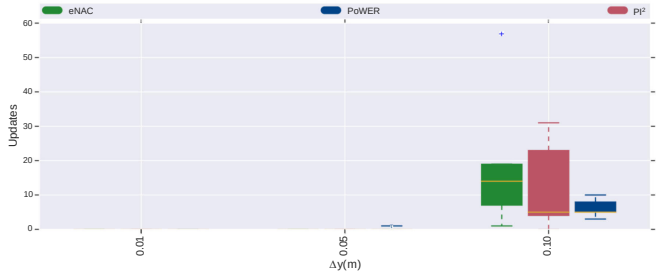

Fig. 7. The update number required for different approaches to find a solution. The horizontal axis indicates the distance from the demonstrated target, and the vertical axis shows the number of updates. The positional differences are applied to the Y direction.

Fig. 6 and Fig. 7 compare the number of required updates to achieve a successful grasp when the object is moved in the XY Plane. Moreover, zero updates indicate the DMPs have compensated for the changes, and the RL was not used. The box charts indicate the results gathered from five different learning sessions for each algorithm. The figures show the changes in Y are harder to compensate for the DMPs due to workspace limitation and robot hand kinematics.

As shown, the RL is required when the changes are greater than ten centimeters on the X-axis, where the median of the required updates is one. When the change reaches 40 centimeters, the PoWER algorithm needs more updates, and the outliers show eNAC and PI^2 encounter occasional difficulties to find a solution. When applying changes in the Y-axis, the PoWER algorithm has a better performance by having a smaller inner fence and lower median. The PI^2 has the same average but has a wider boundary. Additionally, eNAC has the same issue plus a higher median and also an outlier, which shows the algorithm needed more than 50 updates to find a proper solution.

Fig. 8 compares the number of required updates for grasping a cylinder with different approaches. The result shows a small deviation from the original position of the object during demonstration results in a DMP failure, which made this task the hardest among the others. The results also show PI^2 has the best performance while eNAC requires the highest number of updates.

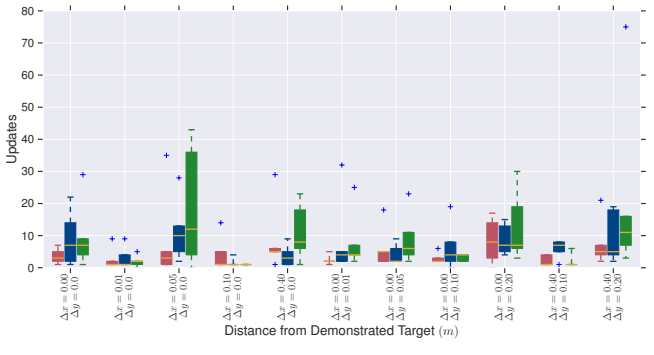

Fig. 8. Comparison between different approaches in grasping the cylinder. The result shows a small deviation from the original position of the object during demonstration results in a DMP failure. PI^2 shows the best performance, while eNAC requires the highest number of updates.

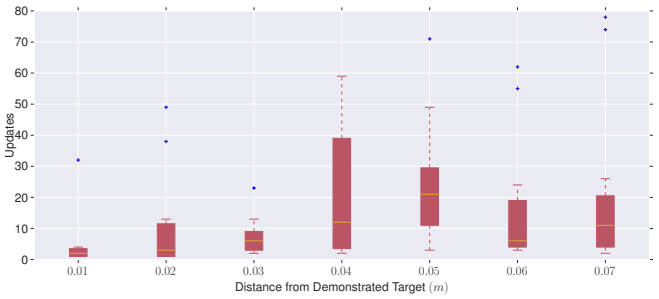

Fig. 9. The number of necessary updates for PI^2 shape and goal learning under uncertainty. The vertical axis determines the number of updates, and the horizontal axis determines the magnitude of uncertainty. The magnitude increases to seven centimeters from the initial object position. The result shows an increase in the number of updates due to increasing the uncertainty, but an anomaly has happened after five trials, and the number of total updates decreases, which can be attributed to the exploration rate.

Fig. 9 compares the number of updates required to achieve a successful grasp under uncertainty. As mentioned, although DMPs are powerful tools, but they fail with small uncertainty in the object position. By increasing the uncertainty, the number of required updates increases as expected. The medians in the figure show that the number of required updates decreases when the uncertainty reaches six centimeters. The drop might be due to the goal exploration rate. So the end-effector does not hit the object, since initially, it is not in the exploration range. Therefore, smaller exploration rates were evaluated, but the results were not satisfying, and the RL agent mostly failed to reach the object in 100 trials. The failure is due to the low probability of finding the object since the haptic feedback is used as a reward, so as long as the object is not touched by at least one of the fingers, the agent does not receive any reward. For example, When the uncertainty is six centimeters, but the exploration rate is two centimeters the agent needs at least three trials toward the object with maximum step size to gain reward from touching the object, which is quite unlikely to happen.

Fig. 10 shows the trajectories generated under different uncertainties in object position. As the trajectories illustrate, the agent has successfully found several solutions for each trial with different uncertainties. The final stage of each trajectory shows where the end effector has grasped the object. The final values of the trajectories, in comparison with the final values of the demonstration, show the approach has successfully found a solution.

V. Discussion

DMPs can help cope with challenging problems such as the ball-in-cup game, obstacle avoidance, and grasping [7], [39], [40]. The results from other research [6] have also shown the method to perform well in a real uncertain environment. In this work, the complexity and challenge of the problems stem from the integration of DMPs into a teleoperation framework. Furthermore, since uncertainty in many cases results from the use of physical hardware/avatar devices, it is essential to evaluate the approach under real conditions. The average delay time of the operation process depends on the demonstration time, network delay, environmental changes (real-time DMP) and the uncertainty of the task (non-real-time RL). The user, however, does not experience this delay as haptic feedback is generated in real-time based on the local model of the avatar environment. This paper aims to evaluate the system in a simulation environment and assess the ability to use sim-to-real where learning happens in simulation and the final results are executed in the real system to speed up the process. The smooth transfer from simulation to real-world will depend on several factors, e.g., correct camera calibration. Our approach takes advantage of combining MMT and learning from demonstration. Furthermore, the algorithms and approaches that have been used in this study were all policy search methods that explore the state-space locally. Therefore, it is very likely that the generated solutions are around the local minimum. The deep RL approaches use a global exploration method, which can improve the generality of the final solution [3], [4].

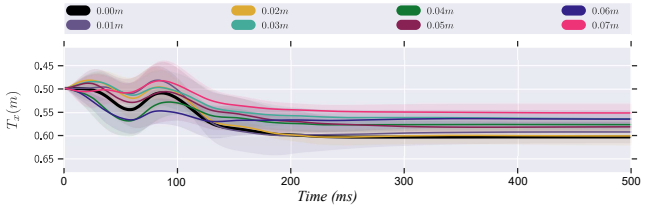

Fig. 10. The trajectories generated under different uncertainty in object position. The vertical axis shows the translation of the end-effector pose in X axis, and the horizontal axis determines the time.

However, deep RL uses neural networks as policy and value functions, and this requires a lot of training data. Therefore, if the avatar robot spends a long time gathering data, the teleoperation may fail, or the inconsistency between demonstration and the environment may increase. Augmented Reality (AR) is used with the intention to give the operator visual awareness of the input device while performing the tasks. However, some operators/subjects have expressed that this caused some distraction. With this in mind, a VR interface may be considered to completely occlude the input device from the operator's line of sight.

VI. Conclusion

We present a novel approach to Model-Mediated Teleoperation for grasping and manipulation, and show that integrating reinforcement learning and DMPs in the control can cope with challenging problems in a long-distance teleoperated grasping scenario under long time-delays. We also show PI^2 has the best performance to adapt to new conditions under high uncertainties in the model. Our approach is realized and examined on DLR's Exodex Adam to operate in a simulated environment through Microsoft Hololens to help assess the ability to use sim-to-real to speed up the process. We believe that apart from looking for learning from demonstration, future research should look for different methods such as learning from imitation and from videos to facilitate the learning process. Regardless, future research could continue to explore the online Deep Reinforcement Learning (DRL) methods, and investigating the effect of using a virtual reality representation might prove important.

REFERENCES

[1] H. Beik Mohammadi, N. Xirakia, F. Abawi, I. Barykina, K. Chandran, G. Nair, C. Nguyen, D. Speck, T. Alpay, S. Griffiths, et al. Designing a personality-driven robot for a human-robot interaction scenario. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4317–4324. IEEE, 2019.

[2] H. Beik Mohammadi, M. Görner, S. Wermter, M. Kerzel, M. A. Zamani, and M. Eppe. Neural End-to-End Learning of Reach for Grasp Ability with a 6-DoF Robot Arm. In *Machine Learning in Robot Motion Planning workshop at International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[3] H. Beik-Mohammadi, M. A. Zamani, M. Kerzel, and S. Wermter. Mixed-reality deep reinforcement learning for a reach-to-grasp task. In *Artificial Neural Networks and Machine Learning – ICANN 2019*, pages 611–623, Sep 2019.

[4] M. Kerzel, H. Beik Mohammadi, M. A. Zamani, and S. Wermter. Accelerating deep continuous reinforcement learning through task simplification. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2018.

[5] M. Kerzel and S. Wermter. Learning of neurobotic visuomotor abilities based on interactions with the environment. In Tamim Asfour and Michael Beetz, editors, *Proceedings of the DGR Days 2017*, pages 14–15, Nov 2017.

[6] F. Stulp, E. Theodorou, and S. Schaal. Reinforcement learning with sequences of motion primitives for robust manipulation. *Robotics, IEEE Transactions on*, 28:1360–1370, 12 2012.

[7] J. Lundell. Dynamic movement primitives and reinforcement learning for adapting a learned skill. Master's thesis, Luleå University of Technology, Department of Computer Science, Electrical and Space Engineering, 2016.

[8] N. Y. Lii, Z. Chen, M. A. Roa, A. Maier, B. Pleintinger, and C. Borst. Toward a task space framework for gesture commanded telemanipulation. *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 925–932, 2012.

[9] P. Schmaus, D. Leidner, A. Schiele, B. Pleintinger, R. Bayer, and N. Y. Lii. Preliminary insights from the meteron supvis justin space-robotics experiment. *IEEE Robotics and Automation Letters*, 3(4):3836–3843, July 2018.

[10] M. Fischer, P. van der Smagt, and G. Hirzinger. Learning techniques in a dataglove based telemanipulation system for the dlr hand. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, volume 2, pages 1603–1608. IEEE, 1998.

[11] N. Y. Lii, Z. Chen, B. Pleintinger, C. H. Borst, G. hirzinger, and A. Schiele. Toward understanding the effects of visual- and force-feedback on robotic hand grasping performance for space teleoperation. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3745–3752, October 2010.

[12] A. Schmidt. Viscosity perception of virtual fluids rendered by a hand exoskeleton. Master's thesis, Delft University of Technology, Faculty of Mechanical, Maritime and Materials Engineering, 2018.

[13] A. Frisoli, E. Sotgiu, D. Checcacci, F. Simoncini, S. Marcheschi, C. A. Avizzano, and M. Bergamasco. Theoretical and experimental evaluation of a 2-channel bilateral force reflection teleoperation system. 2004.

[14] E. Delgado, P. Falcon, M. Diaz-Cacho, and A. Barreiro. Four-channel teleoperation with time-varying delays and disturbance observers. *Mathematical Problems in Engineering*, 2015:1–11, 08 2015.

[15] R. Balachandran, M. Jorda, J. A. Esclusa, J.H. Ryu, and O. Khatib. Passivity-based stability in explicit force control of robots. In *IEEE International Conference on Robotics and Automation ICRA*, June 2017.

[16] A. Schiele, T. Krüger, S. Kimmer, M. Aiple, J. Rebelo, J. Smisek, E. Exter, E. Matheson, A. Hernandez, and F. Hulst. Haptics-2 – a system for bilateral control experiments from space to ground via geosynchronous satellites. 10 2016.

[17] J. Marescaux, J. Leroy, F. Rubino, Michelle Smith, M. Vix, M. Simone, and D. Mutter. Transcontinental robot-assisted remote telesurgery: feasibility and potential applications. *Annals of surgery*, 235(4):487, 2002.

[18] Emmanuel Nuño, Luis Basañez, and Romeo Ortega. Passivity-based control for bilateral teleoperation: A tutorial. *Automatica*, 47(3):485 – 495, 2011.

[19] R. J. Anderson and M. W. Spong. Bilateral control of teleoperators with time delay. *IEEE Transactions on Automatic Control*, 34(5):494–501, May 1989.

[20] B. Hannaford and Jee-Hwan Ryu. Time-domain passivity control of haptic interfaces. *IEEE Transactions on Robotics and Automation*, 18(1):1–10, Feb 2002.

[21] Mark Brudnak. Predictive displays for high latency teleoperation. 08 2016.

[22] A. K. Bejczy, W. S. Kim, and S. C. Venema. The phantom robot: predictive displays for teleoperation with time delay. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 546–551 vol.1, May 1990.

[23] C. Passenberg, A. Peer, and M. Buss. Model-mediated teleoperation for multi-operator multi-robot systems. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4263–4268, Oct 2010.

[24] X. Xu, B. Cizmeci, C. Schuwerk, and E. G. Steinbach. Model-mediated teleoperation: Toward stable and transparent teleoperation systems. *IEEE Access*, 4:425–449, 2016.

[25] C. Liu, J. Guo, and P. Poignet. Nonlinear model-mediated teleoperation for surgical applications under time variant communication delay. *IFAC-PapersOnLine*, 51(22):493 – 499, 2018. 12th IFAC Symposium on Robot Control SYROCO 2018.

[26] M. Panzirsch, H. Singh, M. Stelzer, M. J. Schuster, C. Ott, and M. Ferre. Extended predictive model-mediated teleoperation of mobile robots through multilateral control. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1723–1730, June 2018.

[27] K. Warwick, editor. *Industrial Digital Control Systems*. Control, Robotics amp; Sensors. Institution of Engineering and Technology, 1988.

[28] S. Schaal. *Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics*, pages 261–280. Springer Tokyo, Tokyo, 2006.

[29] E. Theodorou, J. Buchli, and S. Schaal. A generalized path integral control approach to reinforcement learning. *J. Mach. Learn. Res.*, 11:3137–3181, December 2010.

[30] T. Rückstieß, F. Sehnke, T. Schaul, D. Wierstra, Y. Sun, and J. Schmidhuber. Exploring parameter space in reinforcement learning. *Paladyn, Journal of Behavioral Robotics*, 1:14–24, 03 2010.

[31] J. Peters, S. Vijayakumar, and S. Schaal. Natural actor-critic. In J. Gama, R. Camacho, P. B. Brazdil, A. M. Jorge, and L. Torgo, editors, *Machine Learning: ECML 2005*, pages 280–291, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[32] N. Y. Lii, A. Balzer, G. Stillfried, Z. Chen, B. Pleintinger, and M. Grebenstein. Handexoskeleton and robotic arm with such a hand exoskeleton, de102017220936a1, patented in germany, November 2017.

[33] N. Y. Lii, G. Stillfried, Z. Chen, M. Chalon, B. Pleintinger, and A. Maier. exoskeleton, de102017220996a1, patented in germany, November 2017.

[34] F. Stulp and O. Sigaud. Robot Skill Learning: From Reinforcement Learning to Evolution Strategies. *Paladyn*, 4(1):49–61, 2013.

[35] N. Y. Lii and M. Neves. Eingabesystem, de102017220990, patented in germany, May 2019.

[36] M. A. Roa, Z. Chen, I. Staal, J. Muirhead, A. Maier, B. Pleintinger, C. Borst, and N. Y. Lii. Towards a functional evaluation of manipulation performance in dexterous robotic hand design. In *Int. Conf. on Robotics and Automation - ICRA*, Proc. IEEE Int. Conf. on Robotics and Automation - ICRA 2014, pages 6800–6807, June 2014.

[37] A. Pereira, G. Stillfried, T. Baker, A. Schmidt, A. Maier, B. Pleintinger, Z. Chen, T. Hulin, and N. Y. Lii. Reconstructing human hand pose and configuration using a fixed-base exoskeleton. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3514–3520, May 2019.

[38] M. Ortega-Binderberger, S. Redon, and S. Coquillart. A six degree-of-freedom god-object method for haptic display of rigid bodies. *IEEE Virtual Reality Conference (VR 2006)*, pages 191–198, 2006.

[39] D. Park, H. Hoffmann, P. Pastor, and S. Schaal. Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pages 91–98, Dec 2008.

[40] T. Matsubara, S. Hyon, and J. Morimoto. Learning parametric dynamic movement primitives from multiple demonstrations. *Neural Networks*, 24(5):493 – 500, 2011.