2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9560758

Learning from Demonstration for Real-Time User Goal Prediction and Shared Assistive Control

Calvin Z. Qiao*[1], Maram Sakr[1], Katharina Muelling, and Henny Admoni[2]

Abstract— **In shared autonomy, the user input is blended with the assistive motion to accomplish a task where the user goal is typically unknown to the robot. Transparency between the human and robot is essential for effective collaboration. Prior works have provided methods for the robot to infer the user goal; however, they are usually dependent on the distance between the robot and object, which may not be directly associated with the real-time user control intention and thus cause low control feelings. Here, we propose a real-time goal prediction method driven by assistive motion generated by learning from demonstration (LfD) allowing more reactive assistive behaviors. This LfD-generated assistive motion is blended with the user input based on goal predictions to achieve targeted tasks. The LfD policy was learned offline and used with different users. To evaluate our proposed method, we compared it with a state-of-the-art Partially Observable Markov Decision Process (POMDP) based method using a distance cost, and a direct control method (i.e., joystick). A pilot study (N = 6) was conducted to control a 6-DoF Kinova Mico robotic arm to carry out three tasks: (1) reaching-and-grasping, (2) pouring, and (3) object-returning with the three control methods. We used both objective and subjective measures in the comparative study. Results show that our method has the shortest task completion time, the lowest amount of joystick control inputs among all three control methods, as well as a significantly lower angular difference between the user input and assistive motion compared to the POMDP-based method. Besides, it obtains the highest subjective score in the user preference and perceived speed ratings, and the second-highest in the control feeling and the robot did what I wanted ratings.**

I. INTRODUCTION

The user input is explicitly mapped to the robot's final motion during direct robotic teleoperations. To reduce the user's burden, shared autonomy is used to provide intelligent assistance where the transparency between human and robot plays a crucial role in achieving efficient and seamless collaboration [1], [2]. Two perspectives, among many, have become growing interests for elevating the transparency: (1) how to predict the user goal, and (2) how to make the robot's assistive motion more human-like.

Prior user goal prediction research extensively uses distance-based cost functions. One study predicted the goal to be the nearest object around the end-effector in an industrial grasp-and-spray task [3]. Another study combined the distance with the agreement between the user input and the final motion executed by the robot [4]. Distance information was also analyzed and compared using two different forms of cost functions: the hinge loss versus the maximum-entropy-principle-based loss [5]. The maximum entropy principle restores the underlying user costs by updating a normalized distribution over the user control sequences [6]. This principle has been successfully incorporated into the Partially Observable Markov Decision Process (POMDP) framework to assist in minimizing the expected cost-to-go for an unknown user goal and approximating the assistive motion via hindsight optimization, which was validated using an empirically-derived distance cost in the experiment without losing generality [7]. Besides the distance information, gaze cue (e.g., center on the target object) has been adopted for the POMDP framework to update the probabilistic distribution over potential user goals [8]. More recently, a human-in-the-loop deep reinforcement learning method has been proposed to directly map environment observations and user inputs to the robot motion in contexts where the transition dynamics, user policy, and goal representation are unclear [9]. With this method, user goals can be implicitly constructed without defining the goal space.

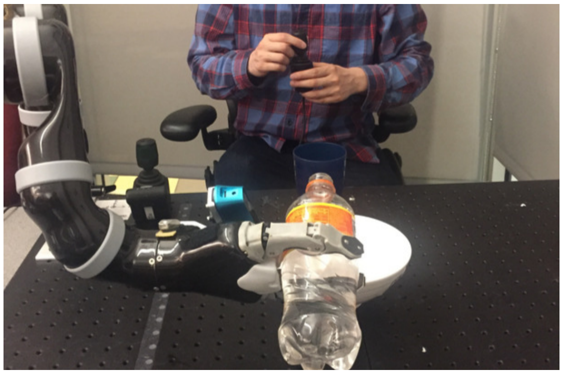

Fig. 1. The robot can assist users in daily tasks such as reaching, grasping, and pouring. In this figure, the user was completing a pouring task.

On the other end of the consideration, learning-from-demonstration (LfD) techniques have been used in shared autonomy to make the robot's motion feel more legible, understandable, and human-like for the users [10], [11]. LfD is categorized, from the learning algorithm perspective, into two main categories: (1) direct and (2) indirect approach. In the former, demonstrations are used to learn a policy that directly maps the input (i.e., state) to the output (i.e., action); this is often referred to as Behavioural Cloning (BC) [12]. In the latter, reward functions are recovered from demonstrations, and then a policy is learned in a way that

[1] Department of Mechanical Engineering, The University of British Columbia
[2] Robotics Institute, Carnegie Mellon University
*Email: calvin.qiao@ubc.ca

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

maximizes these reward functions; this is often referred to as Inverse Reinforcement Learning (IRL) [13] or Inverse Optimal Control (IOC) [14].

One widely-used LfD framework in the direct approach category is the *Dynamic Movement Primitive (DMP)* [15]. This learning method is based on a system of second-order ordinary differential equations in which a forcing term can be "learned" to encode the desired trajectory. DMP has already proven effective for learning repeated tasks to reduce the user burden in shared teleoperation [16]. It has also been deployed in human-robot cooperation tasks for learning how to move in the same direction as the user to collaboratively move a tray of bottles [17]. Furthermore, it has been utilized for learning daily-living tasks in rehabilitation and assistive robotics scenarios [18], [19].

Despite the above-mentioned progress, limited effort has been made to link the user goal prediction to the LfD-generated assistive motion in shared autonomy. By allowing these two components to work together, the system transparency and overall experience may be further improved. Thus, we propose an LfD-based framework for user goal prediction and shared assistive control. To show the effectiveness of the proposed method, we compared it with one of the state-of-the-art methods in prediction-based shared assistive control and a direct control method (i.e., using a joystick). The paper's main contributions are summarized as follows:

1) We propose the use of DMP-based assistive motion for real-time user goal prediction and assistive control in shared autonomy.
2) We conduct a pilot study comparing our proposed methodology with the POMDP framework and direct robotic control for completing daily-living tasks.

II. Methods

In this section, we firstly describe the proposed DMP-based control method; then we summarize the other two control methods that we compared our method with.

A. DMP-Based Assistive Control Method

The DMP-based method is adopted to learn motion skills from demonstrations offline. Specifically, three types of motion skills are learned for the pilot study: (i) reach-and-grasp, (ii) pouring, and (iii) object-returning task by letting a researcher with a robotics background (one of the authors) operate a joystick device to demonstrate these tasks to a 6-DoF Kinova Mico robotic arm. The demonstrations are recorded as the coordinates of the end-effector's motion, which are subsequently used to construct DMP models for real-time online goal prediction and assistive motion generation/arbitration.

1) DMP Learning Model: Each degree of freedom (DoF) of the DMP motion can be modeled as a 2nd-order damped spring system with a goal attractor [20]:

$$\tau \ddot{y} = \alpha_z(\beta_z(g - y) - \dot{y}) + f \tag{1}$$

where y is the system state, g is the goal state, α_z and β_z are gain terms, τ is a temporal scaling factor, and f is a non-linear forcing term:

$$f(x, g) = \frac{\sum_{i=1}^{N} \psi_i w_i}{\sum_{i=1}^{N} \psi_i} x(g - y_0) \tag{2}$$

where y_0 is the initial system state, w_i is the weight of the radial basis function, and x is the canonical system phase which is described by the dynamical equation $\dot{x} = -\alpha_x x$. Radial basis functions are conventionally modeled as Gaussian-types.

Offline-learned DMP skills need to be adjusted for real-time assistive control to accommodate the likelihood of discontinuous control progress, e.g., users briefly pause their control in the middle of the task. Thus, instead of using the dynamical equation producing a continuous change in the canonical system phase, the phase of each object in the environment is estimated by the following two steps: (1) the time-to-go t of the task is estimated as:

$$t = \frac{||\vec{y}_s - \vec{y}_g||}{||\vec{y}_0 - \vec{y}_g||} \cdot T \tag{3}$$

where $\vec{y}_s$ is the robot's 6-DoF sensory state, $\vec{y}_0$ is the initial state, $\vec{y}_g$ is the goal state, and T is the time constant used in the offline DMP baseline training; (2) assuming the user makes rational decisions and aims at moving towards the goal, the canonical system phase x can then be estimated as:

$$x_{est.} = e^{\frac{-\alpha_z \tau t}{t + t_{elapsed}}} \tag{4}$$

where $t_{elapsed}$ is the elapsed time since the start of the task. By retrieving the forcing term f based on $x_{est.}$, we can generate the corresponding 6-DoF DMP system state consisting of 3 translations and 3 rotations for this particular time step and use the state's 1st-order derivatives as the assistive motion $\vec{u}_a^{dmp}(g)$ for this particular object.

2) User Goal Prediction: Offline-learned DMP models are used to predict the user goal by examining the agreement between the user input (i.e., joystick command) $\vec{u}_h$ and DMP-generated assistive motion $\vec{u}_a^{dmp}(g)$ during the control process. The probability of observing the current user input $\vec{u}_h$ given the user goal g is calculated as:

$$\pi^{dmp}(\vec{u}_h|g) = e^{-C_{user}^2(\vec{u}_h, \vec{u}_a^{dmp}(g))} \tag{5}$$

where C_{user} is the user cost defined as:

$$C_{user}(\vec{u}_h, \vec{u}_a^{dmp}(g)) = \arccos(\frac{\vec{u}_h \cdot \vec{u}_a^{dmp}(g)}{\|\vec{u}_h\| \cdot \|\vec{u}_a^{dmp}(g)\|}) \tag{6}$$

which essentially captures the cosine similarity between the user input and assistive motion. Jointly concatenating such probability at each time step, the probability of observing a history of user inputs $\xi^{0 \to t_{elapsed}}$ given the goal g is:

$$p^{dmp}(\xi^{0 \to t_{elapsed}}|g) = \prod_t \pi_t^{dmp}(\vec{u}_h|g) \tag{7}$$

Lastly via the Bayes' rule, the probability of an object being the goal g given the history of user inputs becomes:

$$p^{dmp}(g|\xi^{0 \to t_{elapsed}}) = \frac{p^{dmp}(\xi^{0 \to t_{elapsed}}|g)p(g)}{\sum_{g'} p^{dmp}(\xi^{0 \to t_{elapsed}}|g')p(g')} \tag{8}$$

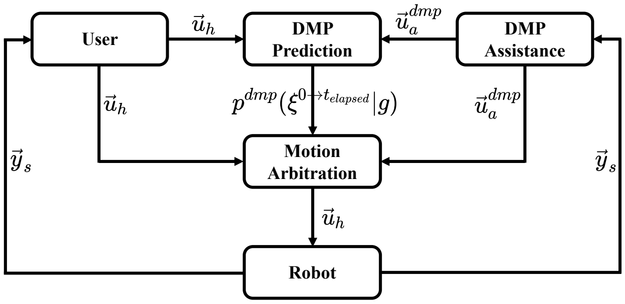

Fig. 2. Block diagram of the DMP-based assistive control method.

where the prior user goal information $p(g)$ is assumed to be uniformly distributed across objects in the environment prior to observing any user control input and is summed up to one.

3) Assistive Motion Generation and Arbitration: Typical motion arbitration techniques can be expressed as follows:

$$\vec{u}_r = (1-\alpha)\vec{u}_h + \alpha\vec{u}_a \tag{9}$$

where $\vec{u}_a$ is the assistive motion, $\vec{u}_h$ is the user input, α is the blending factor, and $\vec{u}_r$ is the final motion executed by the robot. The goal prediction is mapped to the blending factor via a transfer function (e.g, linear or Sigmoid-type):

$$\alpha = f_\alpha(p(g|\xi^{0\to t})) \tag{10}$$

The DMP-based method selects the assistive motion as the DMP-generated assistive motion corresponding to the object with the highest prediction confidence: $\vec{u}_a = \vec{u}_a^{dmp}(g_{dmp})$, while the blending factor used in our pilot study is linearly calculated as:

$$\alpha_{dmp} = \begin{cases} 0, & p^{dmp} < p_{min}, \\ \frac{p^{dmp}(g_{dmp}) - p_{min}}{p_{max} - p_{min}}, & p_{min} \le p^{dmp} \le p_{max}, \\ 1, & p^{dmp} > p_{max}. \end{cases} \tag{11}$$

where p_{dmp} is the prediction confidence for g_{dmp}, p_{min} (= 0.2) is the lower bound triggering a change in the blending factor, and p_{max} (= 0.9) is the upper bound terminating the change. The final robotic motion is normalized based on the magnitude of the user input such that the robot stays still when there is zero user input. The block diagram of the DMP-based method is shown in Fig. 2, and the associated assistive control algorithm is summarized in **Algorithm 1**.

Algorithm 1: DMP-based assistive control algorithm

```
$t_{elapsed} = 0$; $t_{pause} = 0$;
for $g$ in $\vec{g}$ do
    DMP$_g.x_{est.}$ = 0;
    $\vec{u}_a^{dmp}(g) =$ DMP$_g$.step($x_{est.}$);
end
$g_{dmp} = \arg\max_g p^{dmp}(g|\xi^{0\to t_{elapsed}})$;
while ($||\vec{y}_s - g_{dmp}|| > \delta_{dmp}$) do
    $t = \frac{||\vec{y}_s - \vec{y}_g||}{||\vec{y}_0 - \vec{y}_g||} \cdot T$;
    if ($||\vec{u}_h|| \neq 0$) then
        $t_{elapsed}$ += $\Delta t$; $t_{pause} = 0$;
        $x_{est.} = e^{\frac{-\alpha_z \tau t}{t + t_{elapsed}}}$;
    else
        $t_{pause}$ += $\Delta t$;
        if $t_{pause} > 1$ sec then
            $t_{elapsed} = 0$; $t_{pause} = 0$;
            for $g$ in $\vec{g}$ do
                DMP$_g.x_{est.} = 0$;
            end
        end
    end
    for $g$ in $\vec{g}$ do
        $\vec{u}_a^{dmp}(g) =$ DMP$_g$.step($x_{est.}$);
    end
    $g_{dmp} = \arg\max_g p^{dmp}(g|\xi^{0\to t_{elapsed}})$ ;
    $\vec{u}_r^{dmp} = (1 - \alpha_{dmp})\vec{u}_h + \alpha_{dmp}\vec{u}_a^{dmp}(g_{dmp})$;
    $\vec{u}_r^{dmp} = \vec{u}_r^{dmp} \cdot \frac{\|\vec{u}_h\|}{\|\vec{u}_r^{dmp}\|}$
end
```

B. POMDP-Based Assistive Control Method

This subsection summarizes the POMDP-based control method used for comparison in the pilot study [7].

1) User Goal Prediction: The maximum entropy principle is integrated into the POMDP framework to compute the probability of observing the current user input $\vec{u}_h$ given the user goal g and the robot's 6-DoF sensory feedback $\vec{y}_s$:

$$\pi^{pomdp}(\vec{u}_h|\vec{y}_s, g) = e^{V_g(\vec{y}_s) - Q_g(\vec{y}_s, \vec{u}_a, \vec{u}_h)} \tag{12}$$

where $V_g(\vec{y}_s)$ is the approximated value function given the robot's sensory feedback, and $Q_g(\vec{y}_s, \vec{u}_a, \vec{u}_h)$ is the approximated Q function given the robot's sensory feedback, assistive motion, user input, and goal. An empirical distance user cost is adopted for calculating the value and Q functions:

$$C_{user}(\vec{y}_s) = \begin{cases} K, & d > \delta_{pomdp}, \\ K\frac{d}{\delta_{pomdp}}, & d \le \delta_{pomdp}. \end{cases} \tag{13}$$

where the cost is constant when the robot is far away from the object whereas linear when close to the object. Similar to the DMP-based method, the probability of observing the user input history $p^{pomdp}(\xi^{0\to t}|g)$ can be obtained by concatenating $\pi^{pomdp}(\vec{u}_h|\vec{y}_s, g)$ at each time step and converted to goal predictions $p^{pomdp}(g|\xi^{0\to t})$ through the Bayes' rule.

2) Assistive Motion Generation and Arbitration: The assistive motion for an individual assumed goal object g is approximated as the derivative of the Q function via hindsight optimization in the POMDP framework:

$$\vec{u}_a^{podmp}(g) = \dot{Q}_g(\vec{y}_s, \vec{u}_a, \vec{u}_h) \tag{14}$$

The net assistive motion is then calculated as the sum of individual assistive motion weighted by the predictions:

$$\vec{u}_a^{pomdp} = \sum_g p^{pomdp}(g|\xi^{0\to t})\dot{Q}_g(\vec{y}_s, \vec{u}_a, \vec{u}_h) \tag{15}$$

In our pilot study, the POMDP-generated assistive motion is equally blended with the user input, meaning that the user control and assistive motion remain equal influences over the final robotic motion throughout the task:

$$\vec{u}_r^{pomdp} = \vec{u}_h + \vec{u}_a^{pomdp} \tag{16}$$

To make it comparable to the DMP-based method, the POMDP-generated final motion is also normalized based on the magnitude of the user input.

C. Direct Control Method

The direct control method directly uses a Kinova Mico robotic arm joystick to map the user input to the robot's end-effector final motion without involving any assistive motion. Moving the joystick's handle left-right moves the end-effector along the X-axis, front-back along the Y-axis, while rotating it clockwise-counterclockwise moves it up-down along the Z-axis. The user needs to press a button to switch to the angular-velocity control mode where left-right, front-back, and clockwise-counterclockwise handle movement is mapped to the corresponding angular velocity of each axis. The same button needs to be pressed again to switch to a mode in which rotating the handle clockwise-counterclockwise opens-closes the gripper's fingers.

III. Pilot Study

Our pilot study compared three control methods using both objective and subjective measures. Specifically, three tasks were tested: (1) reaching-and-grasping a bottle, (2) pouring the water from the bottle into a container, and (3) returning the bottle to the start position and releasing it.

A. Experimental Procedure

We pilot-tested six subjects recruited from the Carnegie Mellon University; half of them have prior robotics experiences while the other half have no experiences. Subjects were introduced to the experimental setup via a verbal briefing. Then, they were provided two minutes to practice each control method during which they used a Kinova joystick to control a 6-DoF Kinova Mico robotic arm to complete the experimented task sequence. In the experiment, we set up three objects on the table: one orange water bottle and two containers (one blue cup and one white bowl) as shown in Fig. 3. The blue cup acted as an interfering object to diversify the environment. Each trial consisted of three steps: (1) subjects were asked to reach and grasp the orange water bottle; (2) pour water from the bottle to the white bowl; and (3) move the bottle to its original location and release it. The timestamp, normalized user joystick input, and assistive motion were all recorded. Five trials were repeated for each control method, and the control method was referred to by number (e.g., "method 1"). Thus, subjects were not aware of what specific control method they were using. Additionally, we randomized the test sequence of control methods across different subjects to mitigate the carryover effects. Objective/subjective measures were both averaged across trials and subjects.

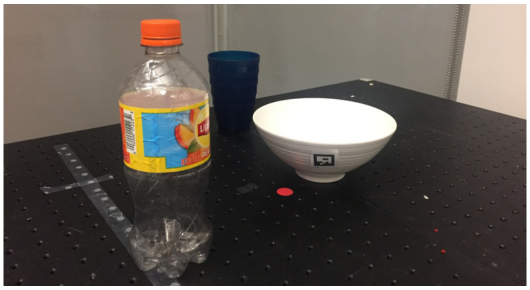

Fig. 3. The object configuration in the environment for the pilot study.

B. Performance Measurements

Three measures were used for objective assessments: (1) the task completion time, (2) total amount of user joystick inputs (normalized between $[0, 1]$ at each time step), and (3) angular difference between the user input and assistive motion. The angular difference metric was only used for evaluating the DMP-/POMDP-based methods where assistance is available. Subjective measures were rated using a survey consisting of the four statements below (adjusted from [7]):

1) "I felt in control."
2) "The robot did what I wanted."
3) "I was able to complete the tasks quickly"
4) "I prefer to use this control method."

The second statement, compared to the first one, emphasized more on whether the robot helps achieve high-level task goals. Subjects were asked to give a rating from 1 to 7 for each statement based on the 7-point Likert scale [21].

C. Hypotheses

We proposed and assessed the following three hypotheses.

H1 The direct control method consumes a much longer time with higher amount of user inputs than other methods.

H2 The DMP-based method has a much lower angular difference between the user input and assistive motion than the POMDP-based method.

H3 The DMP-based method has a higher control feeling than the POMDP-based method.

IV. Results

A. Objective Measures

Table I shows the average objective measures for the three control methods. Overall, the DMP-based method outperforms the other two as shown in Table I, and Fig. 4, 5, 6. Compared to the POMDP-based method, it is faster by 17.9% (39 vs. 46 sec), consumes less user inputs by 9.9% (1101 vs 1211 total joystick inputs), and has a significantly lower angular difference by 30.9% (1.39 vs. 1.82 rad).

Friedman test shows a significance in the completion time ($p < 0.001$) and total amount of user inputs ($p < 0.001$) between the three methods. Conover Posthoc analysis identifies that the direct control method results in a much longer task completion time and higher amount of joystick inputs compared to the DMP-based ($p < 0.001$) and POMDP-based ($p < 0.001$) method. Thus, we find support for **H1**. No

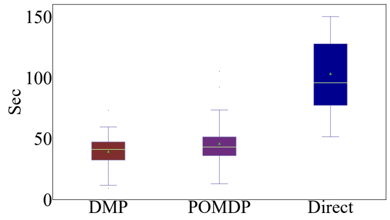

Fig. 4. Task Completion Time

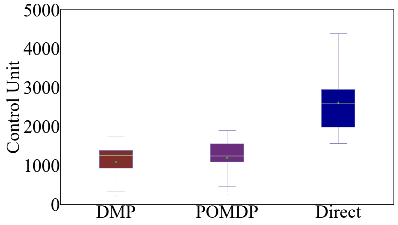

Fig. 5. Total Joystick Inputs

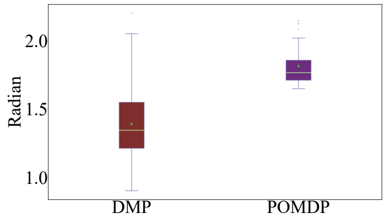

Fig. 6. Angular Difference

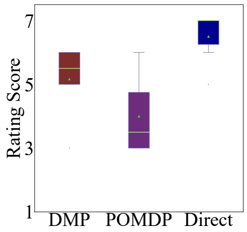

Fig. 7. Rating: Control

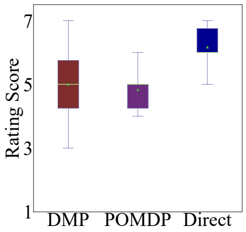

Fig. 8. Rating: Want

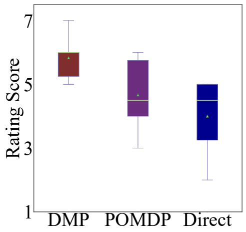

Fig. 9. Rating: Speed

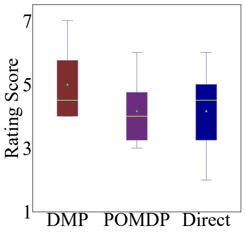

Fig. 10. Rating: Preference

Black dots are the data outliers; light-green triangles are the mean values; light-green lines are the median values.

significant differences are detected in the task completion time and the total amount of joystick inputs between the DMP-based and POMDP-based method. For the angular difference, the DMP-based method is significantly lower ($p < 0.001$) than the POMDP-based method supporting **H2**.

B. *Subjective Measures*

Table II shows the average subjective ratings of the survey. Overall, the DMP-based method achieves the highest score in the user preference (Fig. 10) and perceived speed (Fig. 9) ratings, while obtaining a moderate score between direct and POMDP-based methods in both the control feeling (Fig. 7) and the robot did what I want ratings (Fig. 8).

Friedman test finds a significance in the control feeling rating ($p < 0.05$). Pairwise posthoc analysis shows the following comparative results: DMP vs. POMDP ($p = 0.35$), DMP vs. direct ($p = 0.08$), POMDP vs. direct ($p < 0.05$), partially providing evidence for **H3**. A significance is also found in the rating of the robot did what I wanted ($p < 0.05$). Posthoc analysis reveals that the direct control method significantly differs from the other two assistive methods: DMP vs. direct ($p = 0.047$), POMDP vs. direct ($p = 0.027$).

Marginal significance is discovered for the rating of the perceived speed ($p = 0.055$) while no significance for the overall preference ($p = 0.28$). By a relaxed posthoc analysis ($\alpha = 0.055$), we notice that the statistical significance between the DMP-based and direct control method ($p = 0.053$) is near the threshold.

TABLE I
AVERAGE OBJECTIVE MEASURES

	Time	User Input	Angular Difference
DMP-based	**39 sec**	**1101**	**1.39 rad**
POMDP-based	46 sec	1211	1.82 rad
Direct control	103 sec	2612	N/A

TABLE II
AVERAGE SUBJECTIVE MEASURES

	Control	Want	Speed	User Preference
DMP-based	**5.2**	**5**	**5.8**	**5**
POMDP-based	4	4.8	4.7	4.2
Direct control	6.5	6.2	4	4.2

V. DISCUSSION

The DMP-based method has led to the shortest completion time, lowest control effort among all methods, plus a higher agreement between the user and assistive control than the POMDP-based method, which may arise from its unique ability to respond more explicitly to real-time user inputs making the prediction/assistive motion more compatible with the user's joystick control intention. Such unique features potentially explain its higher control feeling rating over the POMDP-based method. Instead of coordinating between the user control and assistive motion, distance cost only takes into account the proximity between the robot and object and is unable to capture such control consistency in the task. In fact, one subject commented that in one trial using the POMDP-based method, the robot seemed to get stuck when approaching very close to one object (not the intended one), which may be due to the converged prediction probability relying only on the distance information. As such, the subject's control feeling was substantially reduced resulting in a less satisfactory overall experience, which is in line with prior observations reported in [7] while contrary to some other works showing correlations between satisfaction and objective control efficiency [22], [23]. The direct control method is extremely responsive and sensitive to user control inputs; however, it does not include any assistance. As a result, it requires a substantially longer completion time and greater control effort to finish the task, which raises the mental and physical burdens on the user side and deteriorates their overall experiences as well. Therefore, a suitable shared assistive control method needs to be developed to balance the objective control efficiency and subjective control feelings.

We believe that our proposed DMP-based assistive control method may mitigate the above-mentioned dilemma, and the results of this paper open up several directions for future research. First, a more extensive user study with more subjects is needed to further validate our results. Second, the offline learning approach can be improved by using the Probabilistic Movement Primitives (ProMPs) for providing a trajectory distribution subsequently employed in the real-time goal prediction and shared assistive control [24]. Third, validating the generalizability of the proposed approach in more complex tasks (e.g, scooping and feeding the food to the user) and environments (e.g., having multiple obstacles around the goal object), which potentially makes the user significantly detour into another motion direction rather than staying close to the demonstrated one. Such task and environment complexity changes may cause the trajectory to be highly nonlinear affecting the canonical phase estimation.

VI. Conclusion

In this paper, we have proposed a DMP-generated goal prediction and shared assistive control method and compared it with the POMDP-based method (distance cost) and direct control method. Instead of relying on the robot's distance information relative to the object, the proposed method is able to adjust its goal prediction and assistance by analyzing the similarity between the user input and assistive motion, thus enabling a more responsive system behavior to the real-time control inputs from the user. Based on the pilot study's results, we have found that the direct control method has a significantly longer completion time and greater amount of user inputs than the other two assistive methods supporting **H1**. The DMP-based method produces a much lower angular difference between the user input and assistive motion supporting **H2**, and receives the highest rating in the user preference and perceived speed. A significance is also identified in the control feeling rating with the DMP-based method being the second-highest providing evidence for **H3**. To further improve the current work, future studies may focus on exploring the resultant effect of increasing the sample size, using more advanced learning from demonstration techniques, and testing whether this method can adapt to more complex and diverse tasks and environments.

References

[1] P. Michelman and P. Allen, "Shared autonomy in a robot hand teleoperation system," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, vol. 1. IEEE, 1994, pp. 253–259.

[2] V. Alonso and P. De La Puente, "System transparency in shared autonomy: A mini review," *Frontiers in neurorobotics*, vol. 12, p. 83, 2018.

[3] C. E. Mower, J. Moura, A. Davies, and S. Vijayakumar, "Modulating human input for shared autonomy in dynamic environments," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2019, pp. 1–8.

[4] D. Gopinath, S. Jain, and B. D. Argall, "Human-in-the-loop optimization of shared autonomy in assistive robotics," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 247–254, 2016.

[5] S. Jain and B. Argall, "Probabilistic human intent recognition for shared autonomy in assistive robotics," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 9, no. 1, pp. 1–23, 2019.

[6] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[7] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization for teleoperation and teaming," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 717–742, 2018.

[8] H. Admoni and S. Srinivasa, "Predicting user intent through eye gaze for shared autonomy," in *2016 AAAI Fall Symposium Series*, 2016.

[9] S. Reddy, A. D. Dragan, and S. Levine, "Shared autonomy via deep reinforcement learning," *arXiv preprint arXiv:1802.01744*, 2018.

[10] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2013, pp. 301–308.

[11] A. Pichler, S. C. Akkaladevi, M. Ikeda, M. Hofmann, M. Plasch, C. Wögerer, and G. Fritz, "Towards shared autonomy for robotic tasks in manufacturing," *Procedia Manufacturing*, vol. 11, pp. 72–82, 2017.

[12] M. Bain and C. Sammut, "A framework for behavioural cloning." in *Machine Intelligence 15*, 1995, pp. 103–129.

[13] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, 2000, p. 2.

[14] P. Moylan and B. Anderson, "Nonlinear regulator theory and an inverse optimal control problem," *IEEE Transactions on Automatic Control*, vol. 18, no. 5, pp. 460–465, 1973.

[15] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1398–1403.

[16] A. Pervez, H. Latifee, J.-H. Ryu, and D. Lee, "Motion encoding with asynchronous trajectories of repetitive teleoperation tasks and its extension to human-agent shared teleoperation," *Autonomous Robots*, vol. 43, no. 8, pp. 2055–2069, 2019.

[17] T. Kulvicius, M. Biehl, M. J. Aein, M. Tamosiunaite, and F. Wörgötter, "Interaction learning for dynamic movement primitives used in cooperative robotic tasks," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1450–1459, 2013.

[18] C. Lauretti, F. Cordella, E. Guglielmelli, and L. Zollo, "Learning by demonstration for planning activities of daily living in rehabilitation and assistive robotics," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1375–1382, 2017.

[19] J. Nielsen, A. S. Sørensen, T. S. Christensen, T. R. Savarimuthu, and T. Kulvicius, "Individualised and adaptive upper limb rehabilitation with industrial robot using dynamic movement primitives," in *ICRA 2017 Workshop on Advances and challenges on the development, testing and assessment of assistive and rehabilitation robots: Experiences from engineering and human science research*, vol. 1, 2017, p. 40.

[20] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[21] W. M. Vagias, "Likert-type scale response anchors. clemson international institute for tourism," *& Research Development, Department of Parks, Recreation and Tourism Management, Clemson University*, 2006.

[22] 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.

[23] K. Hauser, "Recognition, prediction, and planning for assisted teleoperation of freeform tasks," *Autonomous Robots*, vol. 35, no. 4, pp. 241–254, 2013.

[24] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, no. 3, pp. 529–551, 2018.