Adaptive Shared Control with Human Intention Estimation for Human Agent Collaboration*

Negin Amirshirzad[1], Emre Ugur[2], Ozkan Bebek[2] and Erhan Oztop[4]

Abstract— **In this paper an adaptive shared control framework for human agent collaboration is introduced. In this framework the agent predicts the human intention with a confidence factor that also serves as the control blending parameter, that is used to combine the human and agent control commands to drive a robot or a manipulator. While performing a given task, the blending parameter is dynamically updated as the result of the interplay between human and agent control. In a scenario where additional trajectories need to be taught to the agent, either new human demonstrations can be generated and given to the learning system, or alternatively the aforementioned shared control system can be used to generate new demonstrations. The simulation study conducted in this study shows that the latter approach is more beneficial. The latter approach creates improved collaboration between the human and the agent, by decreasing the human effort and increasing the compatibility of the human and agent control commands.**

I. INTRODUCTION

Autonomous agents have found their way into our daily life and their encounters with humans are increasing rapidly. A common form of this encounter is in form of a collaboration where humans and agents are working together to achieve a common goal. This form of collaboration has important challenges compared to a multi-agent collaboration scenario [1], because human behavior and its creativity, irregularity, and unpredictability should be considered to achieve a successful collaboration. One way that humans and agents can collaborate is through sharing the control of any manipulator to accomplish a task. When the control is given both to a human and an intelligent system, an ideal strategy is to use the commands of both controllers to decrease the human effort in order to achieve a better performance than single agents. One of the simplest strategies is called traded control [2], where the control authority can be transferred completely either to the human or to the agent. This strategy is applied in aircraft autopilot systems where the control is given to the computer while cruising, and more complex situations such as taking-off and landing the control are handled by the human pilot. In earlier literature of human-agent shared control, a leader-follower strategy was used frequently [3], where the agent was taking the role of the follower and the human was taking the role of the leader. This strategy has become popular again because it can reduce human effort and increase task performance, and is now being widely used in robotic-assisted surgeries [4]. Another strategy is tactile shared control [5], in which the input of the agent is transmitted to the operator through the force of the user interface. Operators can obey or resist these forces. This strategy has been applied in [6] for wheelchair control. Among other popular strategies for shared control is to use the integration of human and agent commands. The arbitration can be done via a policy blending approach [7], [8], where a task weight coefficient is assigned between human and agent commands.

In general, shared control methods are supposed to reduce the workload and improve operation efficiency by combining manual teleoperation with autonomous assistance where human intervention is still indispensable. As human actions are often goal-directed [9], it is important for the agent to know the goal of the human in order to assist the human in accomplishing the corresponding goal. Therefore the goal of the human can explicitly be given to the agent [10]. In some scenarios the agent does not know the goal specifically but assumes that the human is following a predefined path or behavior [11], [12], [13]. Alternatively, the agent can estimate and update the human goal as the task execution continues and sufficient information for human goal estimation is collected [14], [15].

Recently, in shared control frameworks, more studies have focused on a more effective and safer human-agent interaction. In [16] a heuristic-based prospective method adapting the triplet Competence-Availability-Possibility-to-act (CAP) is proposed to discover possible conflicts of shared control between humans and autonomous system. In [17] an indirect shared control method is proposed to model the situation where the autonomous agent and human have different intentions, which can cause severe conflicts during the cooperation, using a Nash equilibria strategy the best response for the automated system is obtained, which can guide the automated controller to act more safely and comfortably. In another study [18] inspired by sensorimotor mechanisms of the primate brain, object affordances were utilized for both intention estimation and task execution, to generate an altruistic model. To assess how human partners interact with such an altruistic model a study with naive human subject was conducted, which resulted in engaging human partners to exploit the behavior.

In this paper a shared control framework is proposed,

* This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant No. 119E036.

[1] N. Amirshirzad is with the Department of Computer Science, Ozyegin University, Istanbul, Turkey negin.amirshirzad@ozu.edu.tr

[2]E. Ugur is with the Department of Computer Science, Bogazici University, Istanbulemre.ugur@boun.edu.tr

[3]O. Bebek is with the Department of Mechanical Engineering, Ozyegin University, Istanbul, Turkey ozkan.bebek@ozyegin.edu.tr

[4] E. Oztop is with SISREC, OTRI, Osaka University, Japan and Ozyegin University, Istanbul erhan.oztop@otri.osaka-u.ac.jp, erhan.oztop@ozyegin.edu.tr

where human and agent are coupled to control a manipulator for accomplishing a task. This framework has the potential to be used where full comprehensive sensing of the environment is impractical, and where quick decision making ability of the human when unseen or unexpected events may happen is needed to provide a level of robustness that is difficult to encode. In the next section, the proposed shared control framework is introduced and its components are explained from a general perspective. In the third section, the implementation and setup are explained, and then the results of experiments are reported. The fourth section concludes this study and provides discussions for the potential future work.

II. Framework

Below is the general description of the proposed adaptive shared control framework which is illustrated in the Fig.1. This framework can be adopted for different teleoperation tasks for human-agent collaboration, and its components (particularly the feedforward and feedback interfaces) can be chosen based on the specifications of the given task.

A. Feedforward interface

The main job of the feedforward interface is to communicate information from human to the system. In this framework, this unit is responsible for taking the input of the human operator via an interface and converting it to the control commands to run the manipulator.

B. Feedback interface

The feedback interface unit is responsible for communicating information from the system to the human operator. In shared control systems, this feedback interface is usually in form of visual information about the environment and manipulator [15]. The human operator can observe the scene either directly or through a live video recording. Since teleoperating tasks involve interacting with a physical setup, taking advantage of the sense of touch can also provide an effective force-feedback for the human operator [19].

C. Autonomous controller

The autonomous controller component of this framework has two main jobs; estimating the human intention and generating control commands accordingly. Agents can interact and collaborate closely with humans while reasoning and learning about human intentions. Enabling the agents to learn the human intention is a challenging but a critical issue to address.Usually the agents do not have extensive knowledge about the environment and they assess it with perception, prediction and creating a general model of the environment. Predicting human intention accurately in all circumstances does not seem feasible because of the complex nature of human behaviors. Human behaviors are very stochastic, not only different human operators can have different strategies for accomplishing the same task, but also the same human operator can apply different strategies when repeating that same task.Nonetheless, assisting the human operator and an efficient collaboration, without knowing the human intent is not possible, so the system first needs to estimate the human intention through some measurement and probabilistic inference. Some of the common methods for human motion and intention prediction use visual information[20], and some require wearing sensing devices[21]. Without the help of wearing sensing devices or visual systems, collecting human demonstrations and observing human behavior when performing a task can facilitate human intention estimation.

D. Shared controller

The shared controller unit is responsible for taking the autonomous command and human command and blending them. This unit is adaptive and able to give the appropriate portion of the control at the right time to the right party while keeping the task trajectory smooth and seamless. This shared controller unit uses a blending parameter that is obtained based on the autonomous controller confidence in estimating the human intention.In the proposed adaptive shared control framework, as the task starts by the human operator sending the control command, the agent estimates the human intention by observing the task state; and based on this estimation, the corresponding autonomous command is generated. The confidence in the estimation of human intention determines a blending parameter. The blending parameter is used for combining the control commands and decides how much weight should be assigned to the human command and autonomous command when calculating the shared command that drives the manipulator. The human intention estimation and the blending parameter update processes run at each step until the task is accomplished. The implementation detail of this adaptive shared control framework is explained in detail in the following section.

III. Implementation and Experiments

A. Learning from weighted demonstration

The autonomous controller works based on learning from demonstrations. In order to teach to the autonomous controller, an extended model of Conditional Neural Movement Primitives (CNMPs)[22] is used. CNMPs are originally used for learning from demonstrations and are built on top of Conditional Neural Processes (CNPs) [23]. After learning, CNMPs can be conditioned on single or multiple desired trajectory points at specific time-steps, to produce trajectory distributions that satisfy the given conditions. Since CNMPs work was formulated for learning from perfect demonstrations, their demonstration data quality affected the estimations directly. On the other hand, assuming that different demonstrations of a task is given to the network to learn, some of these demonstrations could be better than other ones, and each of the given demonstrations can yield a different performance level. In the original formulation of CNMPs, given a condition, the network aims to generate a trajectory that satisfies the corresponding condition, without considering the performance factor. Different from the original CNMPs, in cases where different demonstrations have different levels of performances, we propose to take into account the performance factor and lead the network towards

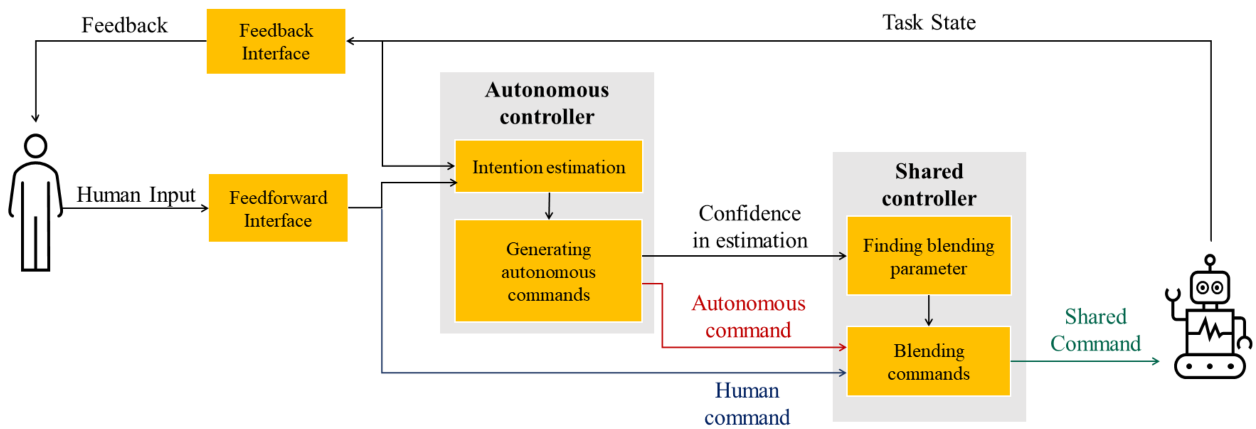

Fig. 1: The human and the agent share the control to accomplish a task via a manipulator. the agent estimates the human intention and generates control commands accordingly. Human commands and autonomous commands are then combined using a blending parameter that is obtained based on how confident the agent is in its estimation of the human intention.

the estimating trajectories that perform better while making sure that the conditions are still satisfied. When training a CNMP network, at each training iteration a demonstration is selected form the demonstrations data set. From that demonstration a number of observation points and a target point are sampled. Each observation is in a form of a tuple consisting of the time step and the features of the demonstration for the corresponding time step. The network outputs the mean and the variance of features of the demonstration at the target point allowing the calculation of a loss with respect to the ground truth feature. The loss is then minimized by stochastic gradient descent [24].

In this study, the demonstrations are evaluated before being used for training the network. Demonstrations can be evaluated based on different factors, and the evaluation may be task specific. After the demonstrations are evaluated, their evaluation score is used to weigh the demonstration impact on the network estimations. Using the evaluation score, a normal distribution is formed, which is centered around the demonstrations with higher scores. As stated in Algorithm 1, training starts by selecting a demonstration using the formed normal distribution, from the selected demonstration, some observation and query points are sampled randomly, the network then estimates The mean μ and variance σ^2 at the query point, having the ground truth of the query, the loss value is calculated and minimized by gradient descent.

Considering that demonstrations with higher scores have more chance of being selected during training, it is expected that they will have a greater influence on network estimations. In order to examine the impact of weighting the demonstrations and compare the outputs of CNMP with the modified network, we generated three movement trajectories as shown in Fig. 2. Two models were designed to learn and generate these trajectories from a start point to a target point while avoiding the obstacles on the way. The trajectories with longer distance to the obstacles were more desired hence

Algorithm 1 Neural Network Training

```
Demonstrations ← Load the demonstrations
Scores ← Evaluation(demonstrations)
Pd ← NormalDistribution(demonstrations, scores)
Initialize threshold Lt and number of observations m
while loss ≤ Lt do
    Select a Demonstration D using Pd
    n = randint(m)
    pick n observation point from D in form of [t_i; f(t_i)]
    Pick a query in form of [t_q; f(t_q)]
    Estimate(μ, σ^2)
    Pn = N(μ, σ^2)
    loss = −log(Pn(f(t_q)))
    loss is minimized by stochastic gradient descent
```

they got higher scores. After the models are trained, both models generated trajectories given one single observation as a condition point. It is expected that both models should generate a trajectory that satisfies the given condition. Fig. 2 shows the generated trajectories by these two models. The results show that given a common condition (where all the given demonstrations satisfy that condition), the CNMP model generates an average trajectory, while the modified model generates a more desired trajectory.

B. Human Intention Estimation

We also propose to use the aforementioned neural network model for human intention estimation by directly conditioning it in real-time based on human behavior and the current state of the task. Consider a scenario in which the human operator starts moving the manipulator to a desired position and orientation at time t. The neural network model is already trained, and and when being conditioned on the current state of the task (that is led by human and autonomous command), it generates a trajectory that

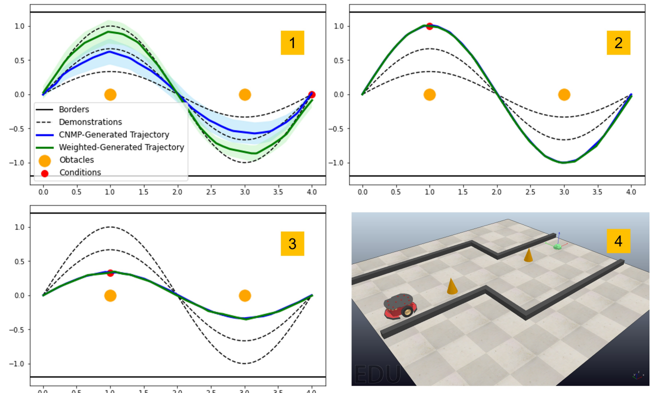

Fig. 2: Generated trajectory; x(m) and y(m) position; by CNMP (blue line) vs weighted model (green line). The light blue and light green color show the standard deviation of the estimation. sub figure (1) shows the generated trajectories when being conditioned on a common point, which shows when both models are uncertain, the weighted model generates a more desired trajectory.In (2) and (3), models are conditioned on an uncommon point (satisfying only one seen trajectory) and generate similar trajectories. Simulation scene is depicted in (4).

satisfies the given condition for accomplishing the task. This generated trajectory is considered as the estimate of human intention. The estimate of human intention μ and the variance of the estimation σ^2 are obtained by the neural network. Based on these estimations, the autonomous control commands will be generated.

C. Adaptive shared controller policy

Having the human and autonomous control commands, the adaptive shared controller should blend the commands and drive the manipulator. When estimating human intention, human's desired trajectory μ and the variance of estimation σ^2 are generated. At each time-step of the task t, the model estimates $\mu_t(i = t+1 : T)$ and $\sigma_t^2(i = t+1 : T)$, for the next time-steps $(i = t+1 : T)$, where T is the last time-step of the task. The mean of estimation's variance starting at time-step $t+1$ continuing till task completion time T taken as $\sigma_t^2(i = t+1 : T)$ can be used to define a confidence factor c_f. The confidence factor $c_f(t)$ is calculated with (1) and indicates how confident the autonomous controller is in estimation of the human intention at time t.

$$c_f(t) = 1/(K\frac{\sum_{i=t+1}^{T}\sigma_t^2(i)}{T-t}) \tag{1}$$

Where K is a scalar that was tuned manually. When the value of $c_f(t)$ is large, meaning that the autonomous controller is confident in its estimation of human intention, the human effort can be decreased and the autonomous controller can take a larger portion of control. For this, the weight of human control command can be obtained by $w(t) = 1 - c_f(t)$, where $0 \leq w(t) \leq 1$ is the adaptive blending parameter that indicates the weight of human control command at t. The blending parameter is sensitive to the value of K, and K is tuned based on σ of the estimation, such that c_f is close to 0 when the σ reaches its maximum value. Ultimately the shared command that drives the manipulator at t is calculated by (2):

$$C_{sh}(t) = w(t)C_h(t) + (1 - w(t))C_a(t) \tag{2}$$

Where $C_{sh}(t)$, $C_h(t)$, $C_a(t)$ are shared command, human command and autonomous command at time-step t, respectively.

D. Experiment

To examine this shared control setup, a simple task was designed and simulated in CoppeliaSim Edu [25]. In this task a mobile robot (Pioneer p3dx) should reach a given target point in a determined fixed time (15 secs in the experiments reported) while avoiding the obstacles along the way. The simulation scene is shown in Fig.2. For this task human input was taken via a computer mouse as a feedforward interface. Mouse displacements in x and y axis were used to first calculate the linear and angular velocity of the mobile robot, and the obtained velocities were converted to right and left wheel motor commands. The feedback to human was visual through the computer screen.

The neural network in the autonomous controller was first trained using a fixed set of demonstrations (3 in the reported experiments) as shown in Fig.2 and evaluations corresponding to the demonstration were given to the learning algorithm (see Algorithm. 1. The evaluations assessed how well the trajectories cleared the obstacles, by monitoring the closest distance to the obstacles. The network was designed to estimate x and y position of the mobile robot, given single or multiple observations as conditions. Observation(s) are given to the network in form of a tuple $o = [t, x(t), y(t)]$, and then the network estimates the mean and the variance of $x(q)$ and $y(q)$ for any query point.(In this experiment the whole range of time-step $(q = 0 : T)$ was used for queries. As the human behavior may change along the way, the intention estimation is done at every time-step continuously. μ_t is the human intention estimation at t and $\mu_t(t+1)$ depicts the desired mobile robot position at the next time-step $t+1$. Having the confidence factor $c_f(t)$ and blending parameter $w(t)$ obtained by (1) and (**??**), the shared command C_{sh} is formed with (2). The steps taken in this shared control loop are shown in Algorithm. 2.

Algorithm 2 Shared Control Loop

```
Model ← Load the trained model
Initialize K
while t ≤ T do
    R_x, R_y ← Read robot state
    C_h ← GetHumanCommand(Interface)
    condition = [t , R_x , R_y]
    μ, σ ← Model(condition)
    C_a ← GetAutoCommand(μ)
    c_f ← Confidence(σ, K)
    w ← GetBlendingParameter(c_f)
    C_sh = wC_h + (1 − w)C_a
    Send the C_sh to robot
```

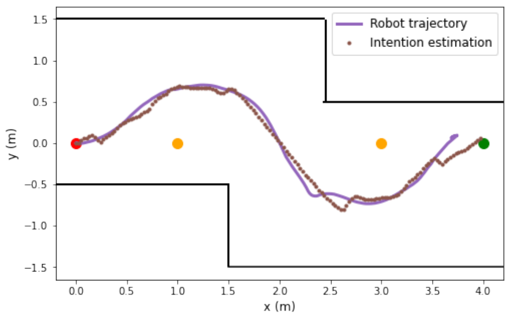

Fig. 3: Mobile robot trajectory and the intention estimation. In here the human intention refers to estimated desired robot position at the next time steps. Red, orange and green dots are starting point, obstacles, and target point respectively.

Using this framework a human operator collaborated with the agent to accomplish the designed task in the simulated environment. The intention estimation method's performance is shown in Fig.3. Also the obtained blending parameter, human and autonomous commands, and task state for the whole duration of the task are shown in Fig.4. As we can see in Fig. 4, the blending parameter that denotes the human share in control increases where the mobile robot is in some common points of the task (such as the beginning and in between the obstacles), since the agent cannot estimate the human intention with high confidence in these points. This is due to the fact that there are different trajectories demonstrated to the agent during training, that match these common points.

E. Feeding back shared trajectories to agent

When using this framework to accomplish a task, new trajectories are formedvia the shared control established by the combined human and autonomous control commands. These trajectories could be beneficial in the sense of making the agent more adapted to the human operator, and vice versa, which can eventually lead to a stronger collaboration. To assess this idea, two setups where designed. In the first setup, three shared demonstrations with evaluations were added to the demonstration data set, and in the second setup, three pure human demonstrations were added to the demonstration data with evaluations. These data sets and their probability distribution for selection (based on their evaluation scores) are shown in Fig.5. Having these two data sets, two new models were trained. To compare the performance of these two models, a human operator tried each model 10 times, to accomplish the designed task. The sum of absolute difference between the human commands and the autonomous commands was considered as a metric for the effectiveness of the collaboration. As the blending parameter depicts the human share in control, it was chosen as the metric for showing the human effort in the collaboration. The means of these two metrics in 10 trials for each setup were computed and compared (see Fig.6). The comparison indicates that the difference between the human and autonomous commands was lower in Setup 1. Furthermore, the human effort was slightly less in Setup 1 compared to Setup 2. Overall,

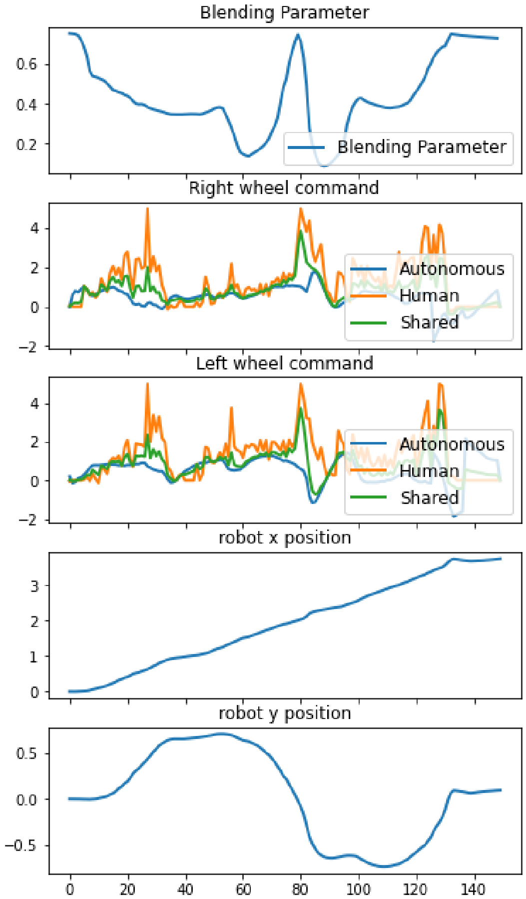

Fig. 4: Blending parameter, human and autonomous commands for right and left wheel of the mobile robot, and task state (robot's position). The blending parameter that denotes the human share in control increases where the mobile robot is in some common condition points of the task (such as the beginning and in between the obstacles)

these data suggests that adding the shared trajectories to the knowledge of the agent induces an abler collaboration and reduces the human effort.

IV. Conclusion

We proposed an adaptive shared control framework for human-agent collaboration. In this framework the agent learns the task with learning from demonstrations, where the given demonstrations are evaluated so that the training uses these demonstrations in weighted manner to obtain an overall action model for the robot. The robot also estimates the human intention with a confidence factor, and with the estimated intention, autonomous command is generated. Using the confidence factor, the shared blending parameter is obtained. Having the blending parameter, the shared control command is formed by the convex combination of human and autonomous command. The shared control command drives a manipulator to accomplish a task. Using this framework new trajectories are formed as the result of human and autonomous controller's commands. These trajectories are fed back to the agent and are used as new demonstrations

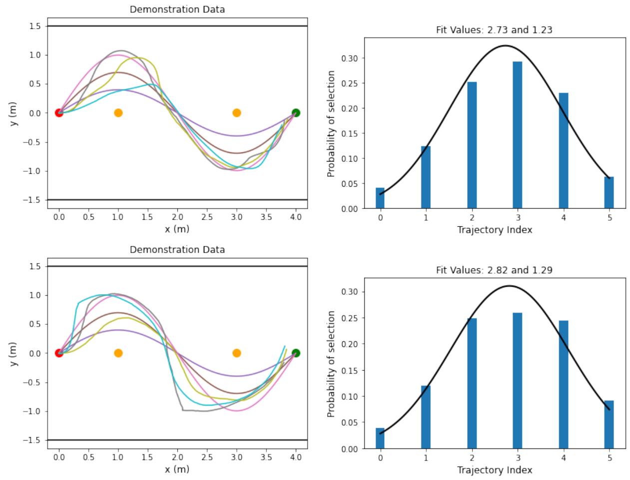

Fig. 5: Adding new demonstrations to the network. First row: previous and new shared trajectories. Second row: previous and new human trajectories.

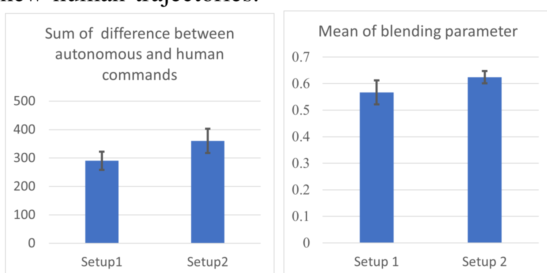

Fig. 6: Sum of absolute difference between human and autonomous commands, and mean of blending parameters in 10 trials for each setup. Error bars are standard deviations.

for learning. This framework was tested on a simulated environment for a designed task where a mobile robot was to reach a target while avoiding the obstacles. The results suggest that using this framework, we can (i) teach the agent to generate more desired trajectories based on their evaluated scores; (ii) assist and decrease the human effort by estimating human intention and obtaining a blending parameter adaptively; (iii) build a stronger collaboration by using the generated trajectories by this shared control framework as new demonstrations and adding them to the knowledge of the agent.

In the next phase of this study we will test this framework with dynamic tasks in uncertain environments. Further, we will employ different human operators to capture their individual strategies in the network so as to equip the network the ability to adapt to different operators seamlessly.

References

[1] R. Chipalkatty, G. Droge, and M. B. Egerstedt, "Less is more: Mixed-initiative model-predictive control with human inputs," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 695–703, 2013.

[2] D. Kortenkamp, R. P. Bonasso, D. Ryan, and D. Schreckenghost, "Traded control with autonomous robots as mixed initiative interaction," in *AAAI Symposium on Mixed Initiative Interaction*, vol. 97, no. 4, 1997, pp. 89–94.

[4] J. H. Palep, "Robotic assisted minimally invasive surgery," *Journal of minimal access surgery*, vol. 5, no. 1, p. 1, 2009.

[3] N. Hogan, "Impedance control: An approach to manipulation: Part i—theory," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–7, Mar.

[5] M. Steele and R. B. Gillespie, "Shared control between human and machine: Using a haptic steering wheel to aid in land vehicle guidance," in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 45, no. 23, 2001, pp. 1671–1675.

[6] A. Herweg, J. Gutzeit, S. Kleih, and A. Kübler, "Wheelchair control by elderly participants in a virtual environment with a brain-computer interface (bci) and tactile stimulation," *Biological psychology*, vol. 121, pp. 117–124, 2016.

[7] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan, "Learning robot objectives from physical human interaction," in *Conference on Robot Learning*. PMLR, 2017, pp. 217–226.

[8] 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.

[9] G. Csibra, "Teleological and referential understanding of action in infancy," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 358, no. 1431, pp. 447–458, 2003.

[10] J. W. Crandall and M. A. Goodrich, "Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation," in *IEEE/RSJ international conference on intelligent robots and systems*, vol. 2, 2002, pp. 1290–1295.

[11] W. Yu, R. Alqasemi, R. Dubey, and N. Pernalete, "Telemanipulation assistance based on motion intention recognition," in *Proceedings of the 2005 IEEE international conference on robotics and automation*, 2005, pp. 1121–1126.

[12] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *Proceedings of the 2005 IEEE international conference on robotics and automation*, 2005, pp. 1139–1144.

[13] D. Nicolis, A. M. Zanchettin, and P. Rocco, "Human intention estimation based on neural networks for enhanced collaboration with robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1326–1333.

[14] N. Amirshirzad, O. Kaya, and E. Oztop, "Synergistic human-robot shared control via human goal estimation," in *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2016, pp. 691–695.

[15] N. Amirshirzad, A. Kumru, and E. Oztop, "Human adaptation to human–robot shared control," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 2, pp. 126–136, 2019.

[16] F. Vanderhaegen, "Heuristic-based method for conflict discovery of shared control between humans and autonomous systems-a driving automation case study," *Robotics and Autonomous Systems*, vol. 146, p. 103867, 2021.

[17] W. Li, Q. Li, S. E. Li, R. Li, Y. Ren, and W. Wang, "Indirect shared control through non-zero sum differential game for cooperative automated driving," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[18] M. Imre, E. Oztop, Y. Nagai, and E. Ugur, "Affordance-based altruistic robotic architecture for human–robot collaboration," *Adaptive Behavior*, vol. 27, no. 4, pp. 223–241, 2019.

[19] E. Ugur and H. Girgin, "Compliant parametric dynamic movement primitives," *Robotica*, vol. 38, no. 3, pp. 457–474, 2020.

[20] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6158–6166.

[21] W. Wang, R. Li, Y. Chen, and Y. Jia, "Human intention prediction in human-robot collaborative tasks," in *Companion of the 2018 ACM/IEEE international conference on human-robot interaction*, 2018, pp. 279–280.

[22] M. Y. Seker, M. Imre, J. H. Piater, and E. Ugur, "Conditional neural movement primitives." in *Robotics: Science and Systems*, vol. 10, 2019.

[23] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. Rezende, and S. A. Eslami, "Conditional neural processes," in *International Conference on Machine Learning*, 2018, pp. 1704–1713.

[24] L. Bottou, "Stochastic gradient descent tricks," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 421–436.

[25] E. Rohmer, S. P. N. Singh, and M. Freese, "Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework," 2013, www.coppeliarobotics.com.