Augmenting Robot Teleoperation with Shared Autonomy via Model Predictive Control

Rolif Lima*, Somdeb Saha, Vismay Vakharia, Vighnesh Vatsal and Kaushik Das

***Abstract*— Shared autonomy enabled teleoperation systems minimise the cognitive load on an operator by providing autonomous assistance during task execution. In contrast to prior approaches using policy blending methods that employ a predict-then-act principle where the robot takes over when confidence in a goal is high, our proposed approach involves continuous policy adaptation. This approach utilises the augmented state of the robot, incorporating both the operator's inputs as well as the robot's autonomous assistance, to provide final assistive control to the robot. To address the issue of the operator's trust in the robot, we formulate the approach as an optimal control problem with the objective of following the operator's input commands while simultaneously adapting the user's inputs to complete the task. We employ a Model Predictive Control (MPC) framework to solve this problem. We evaluated this framework through a user study on multiple goal picking tasks and compared it against pure teleoperation and proximity-based assistance methods. The results of the study show superior performance of our approach over the other methods in terms of trial completion times, collision avoidance, perceived ease of use, and responsive behaviour, indicating its effectiveness in improving teleoperation performance while maintaining user trust in the system.**

I. Introduction

Teleoperation systems have been leveraged for their ability to provide safe access to remote environments and have found use in applications such as surgical robots, hazardous waste handling, deep ocean, and space exploration [1], [2], [3], [4]. Such systems usually consist of a remotely located robot controlled by a human operator using a locally situated control system connected over a communication network. The local control system allows the user to provide inputs to the remote robot via different modalities such as a customized master system, joystick, virtual reality (VR) controllers etc., while also providing feedback from the remote environment to the user through modalities such as audio, video, and force feedback as shown in Fig. 1. Most of the research in the initial stages primarily focused on maintaining the stability of the complete system. Prominent methods for the same include wave variable techniques [5], passivity and scattering [6], adaptive controllers [7], [8], model mediated approach [9], [10], [11], and delay compensation observers [12], [13].

However, it has been noted that the techniques that focus solely on maintaining stability do so by adding more damping into the system. This affects both, the synchronization between the local and remote systems, as well as the feedback provided to the operator (transparency). While such systems are designed to provide the user with an experience of being present in a remote environment (tele-presence), they also introduce certain limitations due to the nature of the modalities involved. Although methods like Predictive Display [14] have been proposed to address these problems, the use of most teleoperation systems are limited to trained operators. This limitation arises due to factors such as lack of an appropriate user interface to access all the degrees of freedom (DoF) of the robot, and a constrained perception of the remote environment. These limitations motivated research along the direction of Shared Autonomy as proposed in [15], [16].

Fig. 1: Operator teleoperating a robot using a VR controller while simultaneously receiving visual feedback through a head-mounted display.

A. Related Work

Shared Autonomy (SA), synonymously known as Assistive Control, or Shared Control [17], generally employs a team consisting of a human operator and an autonomous robot, collaborating in the pursuit of a common task. Shared Autonomy can broadly be divided into two categories, namely policy blending approach [18], [17] and policy adaptation approach [19], [20]. The policy blending approach comprises two major steps: an inference phase that predicts confidence in the operator's intent, followed by an arbitration phase wherein the operator's input is blended in a weighted manner with the autonomous input.

Most of the policy blending works in the literature rely on Bayesian inference to predict the operator's intent. [21]

*Corresponding author. All authors are with TCS Research, Tata Consultancy Services, Bengaluru, Karnataka - 560066, India. e-mail: {rolif.lima}@tcs.com

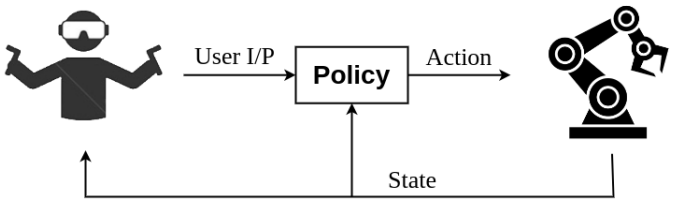


Fig. 2: Shared autonomy with policy adaptation

uses a recursive Bayesian formulation to predict user's goal based on a flexible number of inputs from the user. Javdani et al. [22] formulated the problem as a POMDP with uncertainty over user goals and solved the same using hindsight optimization to obtain approximate solutions. [23] combined natural gaze along with POMDP to predict the goals even when the user's motion is not goal-directed. Other works have aimed to predict a user's goal based on motion prediction models such as in [24]. Works like [18], [25] use Inverse Reinforcement Learning (IRL) to learn the cost function associated with a goal-directed motion which is later used as a reference cost to predict the user's goal. [26] uses Dynamics Movement Primitives (DMP) to learn task-specific human motions and further uses the same to predict the user's intended goal by comparing their inputs with that of the inputs from DMP.

Despite being demonstrated in various applications, policy blending approaches suffer from various limitations as they rely heavily on goal prediction methods. Due to the predict-then-act nature of these methods, there is an almost complete transfer of control from the operator to the robot once the confidence in the predicted goal surpasses a certain threshold. Such techniques have led to reduced human trust in robots [27]. Additionally, the final policy obtained by blending may perform differently from either individual policies, leading to possible instability. The blended policy can also lead to a catastrophic failure if the intent prediction algorithm fails to predict the correct intent and transfers control to the robot.

These disadvantages are solved elegantly by the always-active nature of policy adaptation techniques (Fig. 2), making them a favourable choice for teleoperation systems. Instead of identifying the operator's intent first, these approaches modify the operator's inputs directly by incorporating the user and environment states.

Prior works in policy adaptation have most commonly used a reinforcement learning (RL)-based approach with a composite reward function comprising of one component to stabilize the system and another to achieve the task-specific goal based on the user's feedback [19], or a measure of empowerment [20]. [28] proposed an RL approach using DDQN to learn a policy with a reward function that minimises the robot's intervention in the collaborative task while maximising the operator's performance. [29] proposed a residual-policy learning approach where assistance to the user is provided as a corrective action to the user's input. [30] proposed to train an RL policy in a hierarchical fashion utilizing the offline phase to pre-train an autonomous policy followed by an online phase incorporating the user's feedback.

Despite the success of these techniques in assisting a user to complete a task, their applications remain limited due to their reliance on a precise dynamic model of the environment for pre-training the policy offline. Furthermore, the results are predominantly showcased in simulated environments that typically involve single-goal tasks with discrete action spaces, without considering scalability to tasks with multiple goals and continuous action spaces. RL algorithms are also not data-efficient and face challenges of trust when transferred from simulations to real-world scenarios. This paper attempts to address these gaps by proposing a Model Predictive Control-based Shared Autonomy framework. We propose that the optimal control-based policy adaptation will be effective in an environment with continuous action spaces, enabling an operator to reach multiple goals while maintaining a healthy balance of human-robot trust.

B. Contributions and Overview

There are two main contributions of our work– firstly, we formulate Shared Autonomy as an optimal control problem by introducing a novel cost function along with appropriate constraints that will provide a continuous policy that can cater to multiple goals, while also providing a more satisfying user experience. Simultaneously, the cost function also ensures a smooth switching between user-assisted motion and autonomous robot motion.

Secondly, we conduct a user study where we compare our proposed method against two baselines, namely Proximity-based Shared Autonomy [21] and Pure Teleoperation. Through objective and subjective metrics obtained from the study, we found that our MPC-based method performed better than baselines in terms of trial times and collision avoidance, and was rated higher by users in terms of usability and responsiveness.

The rest of the paper is structured as follows: Section II lays out the problem statement in detail. Section III describes the optimal control formulation and solution methodology for the shared autonomy problem. Section IV describes the experimental study including all the relevant details and the procedures that were followed. The results from the study in terms of objective and subjective metrics are presented in Section V. Finally, in Section VI, we summarise our key findings and their implications, and describe the limitations and future research directions.

II. Problem Description

In this work, we consider the problem of robot teleoperation via policy adaptation-based shared autonomy. The objective of the current work is to perform picking tasks at various locations using a 6-DoF manipulator arm. The operator is tasked with picking three cubes in succession, each located at distinct positions on a table. This particular task was selected due to its simplicity, and the relative ease a novice might feel while using the teleoperation system.

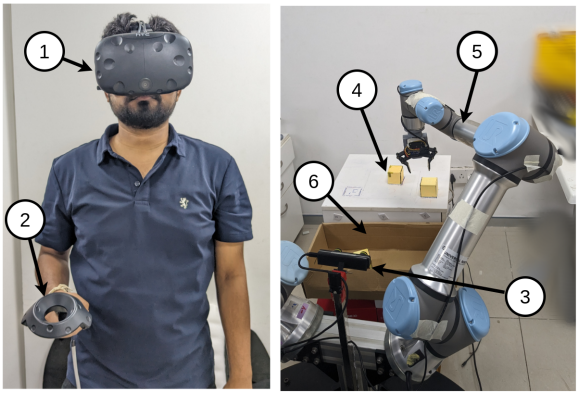


Fig. 3: Hardware Setup: Left shows an operator wearing HMD (1) along with hand held controller(2); Right shows stereo camera (3), goals (4), robot arm (5) and drop box (6).

Fig. 3 shows the configuration of the hardware setup employed in our work. We used an HTC Vive virtual reality (VR) headset, along with its handheld controllers, to translate operator motions into commands for controlling the robot. The robot is a 6-DoF Universal Robots UR-5e series manipulator arm. The popular open-source framework Robot Operating System (ROS) is used as middleware on top of Linux to execute, integrate, and monitor the systems.

The operator's inputs are captured using Virtual Reality (VR) Controllers, and stereoscopic visual feedback is provided through the Head Mounted Display (HMD). With a limited perception of the environment available, it is required to develop an optimal control methodology to assist the operator in completing the teleoperation task smoothly while avoiding collisions with the environment. Our method optimizes the control policies that govern the interaction between the user and the system, resulting in enhanced performance and effectiveness in teleoperation tasks.

III. Optimal Control Formulation

The optimal control problem for achieving operator tracking and goal reaching is formulated as follows.

A. Cost function

In order to ensure that the motion of the robot is synchronized with that of the operator, we utilize a reference tracking cost. Additionally, we also use a goal-tracking cost when the robot approaches a certain goal location. The transition from operator motion tracking to goal tracking is regulated using a scalar weight computed from the user's predicted trajectory. The complete cost function is given as

$$\begin{aligned}\min_{\mathbf{x},\mathbf{u}} J(\mathbf{x},\mathbf{u};\mathbf{g}) = &\sum_i (w_{min}||(\mathbf{x(i)}-\mathbf{x_d(i)})||_{\mathbf{Q}} \\ &+||(\mathbf{u(i)}-\mathbf{u_d(i)})||_{\mathbf{R}} \\ &+\sum_j (1-w(j))||{}^{\mathbf{g}}\mathbf{R_b}(\mathbf{x(i)}-\mathbf{g})||_{\mathbf{P}})\end{aligned} \tag{1}$$

where, i is the i^{th} state and control element in the trajectory. $\mathbf{x}$ is the robot's position and $\mathbf{u}$ is its control effort. The first part of the cost function is the operator or the reference tracking part which ensures that the motion (states as well as control) of the robot is synchronized with that of the operator. $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{P}$ are the diagonal positive definite weighing matrices. The notation $||(.)||_{(*)}$ is used to represent the quadratic form, i.e., $||(\mathbf{x}-\mathbf{x_d})||_{\mathbf{Q}}$ corresponds to $(\mathbf{x}-\mathbf{x_d})^T\mathbf{Q}(\mathbf{x}-\mathbf{x_d})$. The reference trajectory $\mathbf{x_d}$ is generated by forward propagation of a constant velocity model wherein the filtered states for the model are obtained using a Kalman filter.

The final part of the cost function is goal-tracking, which aims to assist the user in taking the robot accurately to the goal location. This gains prominence when the robot is approaching a goal. In order to ensure that the robot approaches the goal only in the desired direction, we first transform the relative position between the robot's end-effector and the goal from the robot base frame to the goal frame using the rotation matrix ${}^{\mathbf{g}}\mathbf{R_b}$ followed by scaling it with the diagonal matrix $\mathbf{P}$ to minimize the error in a certain direction more rapidly than the other.

The two costs are linked using a scalar weight w which is designed to smoothly switch from tracking user commands to assisting the user in reaching a goal of interest based on user states. w_{min} is the minimum of all the computed weights corresponding to different goals. This scalar weight for the j^{th} goal is obtained as:

$$w(j) = \frac{1}{1+\exp(\beta-\alpha||(\mathbf{x}-\mathbf{g})||)} \tag{2}$$

where, α and β are tuning parameters, and g is the goal location.

B. System model

In this work, we use a point mass model to represent the motion model of the robot. This is because we use a commercially available 6-DoF manipulator which comes with a high gain inner loop controller for controlling the robot joints [31]. This renders the system to be an identity-like system to the outer loop optimal controller. The model used for this work is:

$$\dot{\mathbf{x}} = \mathbf{u} \tag{3}$$

C. Constraints

We also enforce additional constraints on the above optimal control problem in terms of control bounds to restrict the applied control within safe limits.

$$\mathbf{u_{min}} \leq \mathbf{u} \leq \mathbf{u_{max}} \tag{4}$$

The collision avoidance constraint used to prevent collision with the planar surface is as an affine constraint of the type:

$$\mathbf{Ax} \geq \mathbf{b} \tag{5}$$

where rows of matrix $\mathbf{A}$ comprise the unit normal to the plane and the elements of $\mathbf{b}$ correspond to the calibrated limits. For example, to prevent the end-effector from colliding with the table, one of the rows of the $\mathbf{A}$ will be chosen as $[0,0,1]$ and the corresponding value of $\mathbf{b}$ will be chosen as the height of the table.

Similarly, to avoid collision with smaller objects, an ellipsoidal constraint is used to restrict the motion of the robot from entering the enclosing ellipsoid around objects in the workspace.

$$(\mathbf{x}-\mathbf{x}_{ref})^T\mathbf{R}\mathbf{M}\mathbf{R}^T(\mathbf{x}-\mathbf{x}_{ref}) > \alpha \tag{6}$$

where $\mathbf{M}$ is a diagonal matrix and $\mathbf{R}$ is the rotation matrix defining the orientation of the ellipsoid. α is chosen empirically to enclose the object in the ellipsoid

The above optimal control problem can be solved using a direct method, employing classical nonlinear programming (NLP) methods, or by using indirect methods, where the system dynamics are first simulated in the forward direction and smaller optimization problems are formulated at each simulation step. These techniques have a computational complexity of $O(Nn^2)$ as compared to $O((Nn)^2)$ of the NLP techniques, where N is the horizon and n is the state dimension. In this work, we use the Augmented Lagrangian Iterative Linear Quadratic Regulator (AL-iLQR) for optimization of the above optimal control problem.

D. Augmented Lagrangian Iterative Linear Quadratic Regulator (AL-iLQR)

An AL-iLQR is an indirect method of solving an optimal control problem with equality and inequality constraints. In this method the constrained optimization problem of the type:

$$\min_x \; l = f(x) \tag{7}$$

$$\text{subjected to: } c(x)\{\leq or =\}0 \tag{8}$$

is solved by first converting it to an unconstrained one as

$$\min_x \; AL = f(x) + \lambda c(x) + \frac{1}{2}c(x)^T I_\mu c(x) \tag{9}$$

where $c(x)$ can represent an equality or inequality constraint in general. λ is the Lagrangian multiplier and I_μ is a diagonal matrix with its diagonal entries restricted to zero for equality constraints and a positive value for inequality constraints.

The modified unconstrained optimization problem is now solved iteratively to update the control variables, Lagrange, and penalty parameters in an alternating fashion. The control parameters are optimized using an iLQR update with the Lagrange and penalty parameters kept constant, followed by their update as follows:

- Solve $min_x\ AL(x,\lambda,\mu)$, holding λ and μ constant.
- Update the Lagrange multipliers as
  - $\lambda_i := \lambda_i + \mu_i c_i(x^*)$ if $i \in \mathscr{E}$
  - $\lambda_i := \max(0, \lambda_i + \mu_i c_i(x^*))$ if $i \in \mathscr{I}$
- Update penalty term: $\mu := \phi\mu$, $\phi > 1$
- Check constraint convergence and if tolerance is not met then repeat form step1

where, $\mathscr{E}$ and $\mathscr{I}$ are equality and inequality constraint sets respectively, and ϕ is the penalty scaling parameter.

The first step of the above procedure is performed using the iLQR method, where the unconstrained optimal control problem is solved using the linear approximation of the system dynamics and quadratic approximation of the cost function. The incremental control law at k^{th} iteration is computed as:

$$\delta\mathbf{u}_k = \mathbf{K}_k\delta\mathbf{x}_k + \mathbf{d}_k \tag{10}$$

where, $\mathbf{K}_k$ and $\mathbf{d}_k$ are the feed-back and the feed-forward gains respectively, which are computed as

$$\mathbf{K}_k = \mathbf{Q}_{uu}^{-1}\mathbf{Q}_{ux} \tag{11}$$

$$\mathbf{d}_k = \mathbf{Q}_{uu}^{-1}\mathbf{Q}_u \tag{12}$$

where $\mathbf{Q}_{uu}$ is the partial double derivative of the action value function $\mathbf{Q}(x,u)$ with respect to $\mathbf{u}$ and similarly $\mathbf{Q}_{ux}$ and $\mathbf{Q}_u$ are the partial derivatives with respect to state $\mathbf{x}$ and control $\mathbf{u}$ accordingly. See [32] for a more detailed explanation of iLQR. The optimal solution obtained from the solver is used as an input to the robot and the corresponding joint angles are computed using an inverse kinematics solution.

IV. Experimental Study

We conduct an experimental study involving human participants who perform a teleoperation task using three different methods. We assess the effectiveness of the methods used in these experiments as well as user satisfaction through objective and subjective metrics of human-robot collaboration. The three methods used were :

1) Pure Teleoperation (T): Obtained operator's pose is directly mapped to robot joint angles via inverse kinematics (IK).
2) Proximity-based Shared Autonomy (P): The robot is guided to the goal using a proportional controller when the operator reaches close to the goal.
3) Shared Autonomy using MPC (S): Our approach as described in section III

We consider a goal-picking task where a human operator must pick three cubes in succession, each located at a distinct position on a table. Fig. 3 displays the configuration of the hardware setup employed in our experiments and the corresponding details are mentioned in Section II.

A. Procedure

We conducted a within-subjects study with N=7 participants, drawn from our research facility staff (7 males, mean age < 25). To minimize the impact of unfamiliarity with the experimental hardware, the subjects were allowed to spend a few minutes doing trial runs. They were also verbally briefed about the questionnaire that they had to answer after each trial. Each trial commenced with the operator starting the hardware shown in Fig. 3 with a button press followed by an attempt to pick the cubes in a prescribed order. The trial ended when the subject reached the final cube. Appropriate data like the robot and operator trajectories, as well as trial completion times, were collected during the trials. On completion of each trial, subjects were prompted with survey statements for subjective assessment before moving on to the next trial. Fifteen trials per participant were conducted in a random order with five trials per method (T, P, S). However, the participants were unaware of which method was being used to control the robot in a particular trial to

avoid any biases. The experimental procedure was compliant with ethical principles set forth in the Belmont Report [33].

B. Performance Metrics

Three objective metrics were utilized for assessing the performance of the system: (1) trial completion time, (2) number of collisions while reaching the goals, and (3) trajectory lengths. Lower values for each of these metrics signify better performance. We expect the MPC-based shared autonomy controller (S) to complete reaching tasks faster, with minimal collisions, and with shorter path lengths than the other two methods.

Subjective measures were extracted from surveys conducted with the participants. The surveys consisted of 7-point Likert items [34], where 1 represented the lowest rating (strongly disagree) and 7 represented the highest rating (strongly agree). Each participant was asked to rate the following set of statements adapted from [35], [36], after each trial:

1) I thought the system was easy to use (Q1).
2) I felt in control of the robot while using the system (Q2).
3) The system was responsive to my commands (Q3).

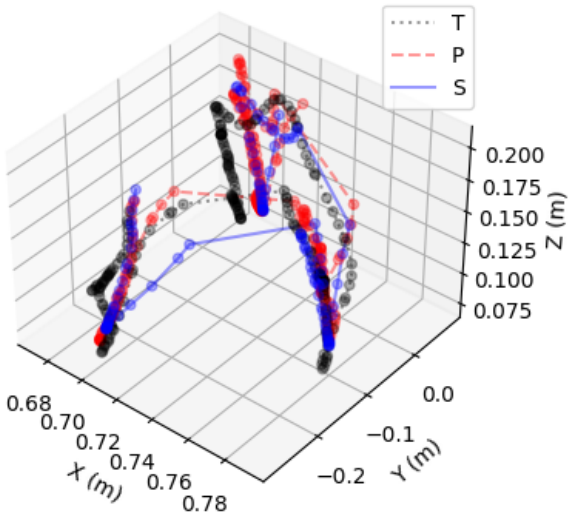
We again expect the MPC-based shared autonomy controller to be rated more highly than the other methods in terms of ease of use, feeling of control, and responsiveness due to the adaptive nature of its assistance.

V. Results

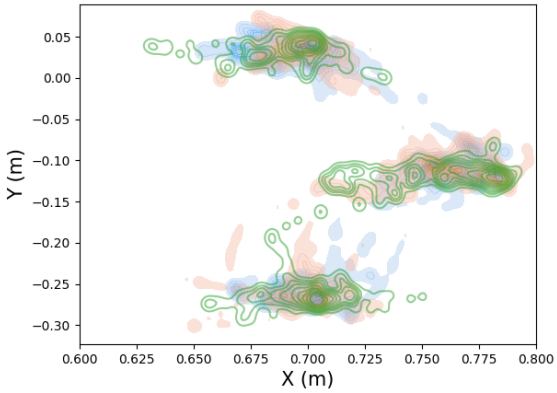
This section presents the evaluation results from the experimental study involving N=7 participants, with 15 trials per participant (5 each for three different control methods). The results include analyses of the quantitative and qualitative metrics as described earlier.

In order to illustrate the three control conditions for one particular user, we plot the robot end-effector trajectories in Fig. 4(a), clipped to a distance of 0.15 m from the goal locations. The clipping is performed as the initial approach trajectory to the goals from the start position in free space is not indicative of performance differences between the methods, which are more prominent near the goal locations due to collision avoidance concerns. The performance differences in the robot arm trajectories are not directly apparent from these 3D plots.

To better capture the spread in the motions, we also plot kernel density estimates in the X-Y plane for all 15 trials for the particular user (Fig. 4(b)). This is because the robot is expected to have more jittery motion in the X-Y plane between and around the goal locations. From Fig. 4(b), we observe that the MPC-based method appears to have a lower spread than the other two methods. These plots, however, serve as illustrations of the procedure without offering concrete insights into performance. In order to better quantify the performance differences between the three control methods, we perform statistical tests on the objective and subjective metrics extracted from the experimental study and summarize the results as follows.



(a) Trajectories in 3D during one trial in each condition.



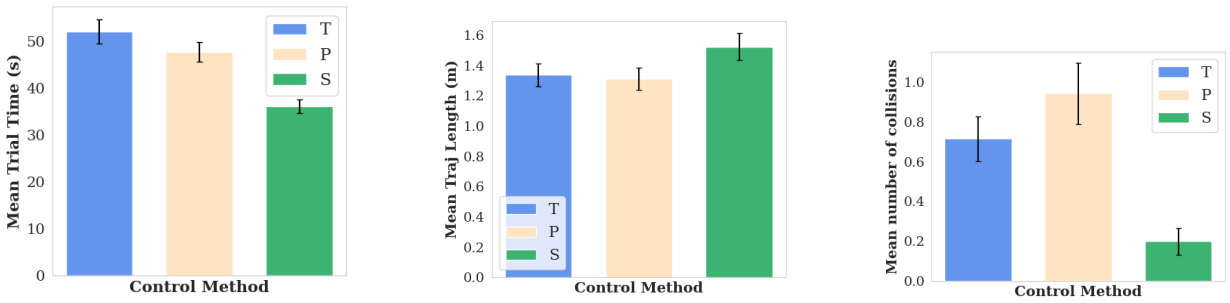
(b) Kernel density maps– Pure teleoperation (T, blue), proximity-based method(P, pink), and MPC-based method (S, green) across all trials for the participant.

Fig. 4: The robot's end-effector trajectory near the tabletop goal positions for one particular user in the three different control methods (T, P, S)– (a) 3D poses in one trial, (b) kernel density estimates of scatter plots in the X-Y plane for all 15 trials by the user.

A. Objective Metrics

Repeated measures ANOVA was performed for each objective metric described in Sec. IV-B for the three control methods. We find that there are significant differences between the three control methods in terms of trial times ($F = 24.01$, $p < 10^{-5}$) and number of collisions ($F = 7.91$, $p < 0.005$). While the mean trajectory length is slightly longer for the MPC-based method (Table I), the differences are not statistically significant ($F = 2.9$, $p = 0.072$).

Using paired t-tests for comparisons between control methods, we find that the MPC-based method performs significantly better than the Proximity-based method both in terms of trial time ($t(6) = 4.94$, $p < 0.0.005$) and number of collisions ($t(6) = 3.49$, $p = 0.013$). It is also better than pure



(a) Trial Completion Time (b) Trajectory Length (c) Number of Collisions

Fig. 5: Objective metrics from the experimental study comparing the three control methods - pure teleoperation (T), Proximity-based (P) and MPC-based (S) shared autonomy.

Metric	Condition	Mean	Std. Dev.
Trial time (s)	T	52.09	14.67
	P	47.72	12.29
	S	**36.13**	8.54
Trajectory length (m)	T	1.34	0.44
	P	1.31	0.43
	S	1.52	0.52
Number of collisions	T	0.71	0.65
	P	0.94	0.89
	S	0.20	0.40

TABLE I: Mean and Std. Dev. for Objective Metrics

teleoperation in terms of trial time ($t(6) = 8.16$, $p < 0.001$) and the number of collisions ($t(6) = 3.75$, $p = 0.009$). These metrics are plotted in Fig. 5.

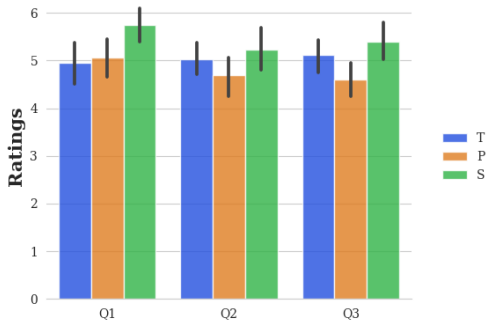


Fig. 6: Subjective ratings on 7-point Likert items to the three survey questions, split by control methods (T, P, S).

B. Subjective Metrics

In terms of subjective metrics, we find that the mean ratings for the MPC-based method are higher than those for the other two controllers for each item, as shown in Fig. 6. These items correspond to usability (Q1), feeling of control (Q2), and responsiveness (Q3). Using repeated measures ANOVA tests on these subjective metrics, we find that the differences between groups are statistically significant for Q1 ($F = 8$, $p = 0.006$), but not for Q2 ($F = 2.29$, $p = 0.143$) or Q3 ($F = 3.39$, $p = 0.068$). A paired t-test showed that participants rated the MPC-based method significantly better than pure teleoperation in terms of usability (Q1, $t(6) = 3.91$, $p = 0.008$).

VI. Conclusion

In this paper, we introduced a shared autonomy approach that utilizes an MPC-based policy adaptation algorithm. The assistive control for the robot is computed by minimizing an optimal control problem that is designed to follow the user's inputs, thereby increasing the operator's trust in the robot while assisting the user in achieving their goals and avoiding collisions with the environment. To evaluate our proposed algorithm, we conducted a user study involving 7 participants who performed a total of 15 trials under three different conditions: five trials using our proposed approach, five trials using pure teleoperation, and five trials using a proximity-based shared control method. The results of the user study revealed that our proposed method outperformed the other two conditions in terms of task completion time and the number of collisions, as indicated by the objective metrics. Additionally, the subjective metrics showed significantly higher ratings for our proposed method compared to the other two conditions in terms of ease of use.

The system in its current form is limited to a discrete set of finite goals. We propose to address this limitation in our future endeavours by using a learning-based approach to learn a generalised weighting function and test its applications to continuous goals such as door opening, liquid pouring etc. We also plan to incorporate our approach to a dual arm robotic setup for bimanual and goal-coordinated tasks.

Acknowledgements

We would like to thank all the members of the robotics team at TCS Research for their valuable assistance in contributing to the experiments.

References

[1] P. Desbats, F. Geffard, G. Piolain, and A. Coudray, "Force-feedback teleoperation of an industrial robot in a nuclear spent fuel reprocessing plant," *Industrial Robot: An International Journal*, 2006.

[2] T. Imaida, Y. Yokokohji, T. Doi, M. Oda, and T. Yoshikawa, "Ground-space bilateral teleoperation of ets-vii robot arm by direct bilateral coupling under 7-s time delay condition," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 499–511, 2004.

[3] C. Domingues, M. Essabbah, N. Cheaib, S. Otmane, and A. Dinis, "Human-robot-interfaces based on mixed reality for underwater robot teleoperation," *IFAC Proceedings Volumes*, vol. 45, no. 27, pp. 212–215, 2012.

[4] M. C. Çavuşoğlu, W. Williams, F. Tendick, and S. S. Sastry, "Robotics for telesurgery: second generation berkeley/ucsf laparoscopic telesurgical workstation and looking towards the future applications," *Industrial Robot: An International Journal*, vol. 30, no. 1, pp. 22–29, 2003.

[5] G. Niemeyer and J.-J. Slotine, "Using wave variables for system analysis and robot control," in *Proceedings of International Conference on Robotics and Automation*, vol. 2, pp. 1619–1625, IEEE, 1997.

[6] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," in *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1, pp. 131–138, IEEE, 1988.

[7] Y. Yang, C. Hua, and X. Guan, "Finite time control design for bilateral teleoperation system with position synchronization error constrained," *IEEE transactions on cybernetics*, vol. 46, no. 3, pp. 609–619, 2015.

[8] V. K. Pediredla, S. Annamraju, A. Thondiyath, *et al.*, "Enhancement of high-fidelity haptic feedback through multimodal adaptive robust control for teleoperated systems," *IEEE Systems Journal*, vol. 15, no. 4, pp. 5526–5536, 2020.

[9] P. Mitra and G. Niemeyer, "Model-mediated telemanipulation," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 253–262, 2008.

[10] B. Willaert, J. Bohg, H. Van Brussel, and G. Niemeyer, "Towards multi-dof model mediated teleoperation: Using vision to augment feedback," in *2012 IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE 2012) Proceedings*, pp. 25–31, IEEE, 2012.

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

[12] Y. Zheng, M. J. Brudnak, P. Jayakumar, J. L. Stein, and T. Ersal, "A delay compensation framework for predicting heading in teleoperated ground vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 5, pp. 2365–2376, 2019.

[13] S. A. M. Dehghan, H. R. Koofigar, H. Sadeghian, and M. Ekramian, "Observer-based adaptive force–position control for nonlinear bilateral teleoperation with time delay," *Control Engineering Practice*, vol. 107, p. 104679, 2021.

[14] N. Sridhar, R. Lima, U. Rai, V. Vakharia, K. Das, and P. Balamuralidhar, "Transparency enhancement in teleoperation: An improved model-free predictor for varying network delay in telerobotic application," in *2021 European Control Conference (ECC)*, pp. 230–235, IEEE, 2021.

[15] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction," *Applied Mechanics Reviews*, vol. 70, no. 1, 2018.

[16] W. Si, N. Wang, and C. Yang, "A review on manipulation skill acquisition through teleoperation-based learning from demonstration," *Cognitive Computation and Systems*, vol. 3, no. 1, pp. 1–16, 2021.

[17] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano, "Autonomy in physical human-robot interaction: A brief survey," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7989–7996, 2021.

[18] 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.

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

[20] Y. Du, S. Tiomkin, E. Kiciman, D. Polani, P. Abbeel, and A. Dragan, "Ave: Assistance via empowerment," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4560–4571, 2020.

[21] S. Jain and B. Argall, "Recursive bayesian human intent recognition in shared-control robotics," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3905–3912, IEEE, 2018.

[22] 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.

[23] R. M. Aronson and H. Admoni, "Gaze complements control input for goal prediction during assisted teleoperation," in *Robotics science and systems*, 2022.

[24] R. C. Luo and L. Mai, "Human intention inference and on-line human hand motion prediction for human-robot collaboration," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5958–5964, IEEE, 2019.

[25] M. Zurek, A. Bobu, D. S. Brown, and A. D. Dragan, "Situational confidence assistance for lifelong shared autonomy," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2783–2789, IEEE, 2021.

[26] C. Z. Qiao, M. Sakr, K. Muelling, and H. Admoni, "Learning from demonstration for real-time user goal prediction and shared assistive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3270–3275, IEEE, 2021.

[27] D.-J. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Portee, J. Bricout, Z. Wang, and A. Behal, "How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 1, pp. 2–14, 2011.

[28] W. Tan, D. Koleczek, S. Pradhan, N. Perello, V. Chettiar, V. Rohra, A. Rajaram, S. Srinivasan, H. S. Hossain, and Y. Chandak, "On optimizing interventions in shared autonomy," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 5341–5349, 2022.

[29] C. Schaff and M. R. Walter, "Residual policy learning for shared autonomy," *arXiv preprint arXiv:2004.05097*, 2020.

[30] S. Chen, J. Gao, S. Reddy, G. Berseth, A. D. Dragan, and S. Levine, "Asha: Assistive teleoperation via human-in-the-loop reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7505–7512, IEEE, 2022.

[31] P. M. Kebria, S. Al-wais, H. Abdi, and S. Nahavandi, "Kinematic and dynamic modelling of ur5 manipulator," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 004229–004234, 2016.

[32] T. A. Howell, B. E. Jackson, and Z. Manchester, "Altro: A fast solver for constrained trajectory optimization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7674–7679, IEEE, 2019.

[33] The National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research, "The Belmont Report: Ethical Principles and Guidelines for the Protection of Human Subjects of Research," 1979.

[34] R. Likert, "A technique for the measurement of attitudes.," *Archives of psychology*, 1932.

[35] Y.-S. Jiang, G. Warnell, and P. Stone, "Goal blending for responsive shared autonomy in a navigating vehicle," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 5939–5947, 2021.

[36] J. Brooke *et al.*, "SUS-A quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.