ATMAS: Assistive Teleoperation Method using Augmented Reality and Switching Control

Somdeb Saha[1,*], Sahil Gaonkar[1],Shubham Parab[1],Rolif Lima[1],Vighnesh Vatsal[1],Vismay Vakharia[1]& Kaushik Das[1]

***Abstract*— This paper presents an assistive teleoperation method using a combination of Augmented Reality (AR) markers and control switching between the operator and robot. This method is designed to enhance remote manipulation tasks in retail environments like convenience stores. The system integrates a novel intention recognition algorithm for predicting goals, augmented reality markers for visual guidance, and a variable autonomy framework. The system is able to adapt under goal switching, which otherwise fails for existing methods. A user study with 7 participants compared our method against two other teleoperation methods in terms of objective and subjective metrics. Results showed that our method significantly reduced collisions during task execution. The study provides insights into the strengths and limitations of augmented reality assistance and variable autonomy in teleoperation, laying groundwork for future research in enhancing human-robot collaboration for retail automation tasks.**

I. Introduction

Automation is rapidly transforming service industries, with convenience stores being ideal candidates for robotic integration as they involve repetitive tasks. Smart pick-and-place (PnP) robots can streamline operations, reduce labor costs, and improve customer experiences. Significant advancements in robot manipulation, grasping, navigation, and path planning have been documented in recent studies([1], [2]). However, teleoperation remains a challenge.

Existing research on robot teleoperation has explored various approaches to enhance user experience and effectiveness. One such approach is robot learning for improved control. Techniques like those proposed by Luo et al. [3] leverage robot learning to provide expert guidance to novice operators, improving efficiency for unskilled users. Another approach involves VR-based intuitive teleoperation. The integration of VR interfaces with robot control systems, as seen in the work of Nakanishi et al. [4] and Gallipoli et al. [5], allows for more natural and immersive teleoperation experiences.

Nicolis et al. [6] proposed a real-time optimization controller based on FSM for visual-servoed dual-arm teleoperation that ensures robust occlusion avoidance in cluttered environments. Garcia Ricardez et al. [7] proposed a robotic system based on compliant hardware design for retail automation tasks like restocking and straightening items on shelves. Naceri et al. [8] introduced the Vicarios Virtual Reality Interface for remote robotic teleoperation. Although it is proposed for disaster management, its interface suggests potential applications in convenience store settings.

While these advancements offer valuable solutions, a significant gap exists in leveraging Augmented Reality (AR) for shared control during teleoperation. AR overlays can provide real-time visual information about the environment and robot state directly onto the operator's view, bridging the gap between VR immersion and the need for real-world context. Additionally, a control framework that leverages both human input and robot autonomy can address complex manipulation tasks efficiently while ensuring safety in dynamic environments.

A. Contributions

Our contributions encompass a novel integration of systems and methods to enable assistive teleoperation for tasks. They can be summarised as follows:

- We formulate a method for online inference of goals based on offline demonstrations. We propose a cost function that is able to accurately capture the operator's intention while simultaneously adapting to a goal switching condition as well.
- We make use of augmented reality to place markers in a scene. This enables the operator to interact with objects via the robot while minimising environmental collisions.
- We provide the user with the freedom to switch autonomy along with real time feedback and predictive trajectory. This allows for faster execution of tasks while also providing novices with a certain degree of confidence in operating the hardware.

II. Problem Statement

In this study, we address the challenge of automated item retrieval from retail shelving units, focusing on a convenience store setting. The setup comprises a teleoperated mobile robotic manipulator and an array of common, distinct consumer products of different shapes and sizes, placed on a shelf as illustrated in Figure 1. The robot is controlled by an operator via a distributed teleoperation system, receiving only the stereoscopic visual feedback of the robot's environment via a Head-Mounted Display (HMD). The primary task in the problem involves efficiently retrieving items from the shelf in a randomized fashion while avoiding environmental collisions or product damage via teleoperation. Additionally, to mitigate the operator's cognitive burden, we propose a

*Corresponding Author: `somdeb.saha@tcs.com`

[1]All authors are with TCS Research, Tata Consultancy Services Ltd., Bangalore, KA, India-560066

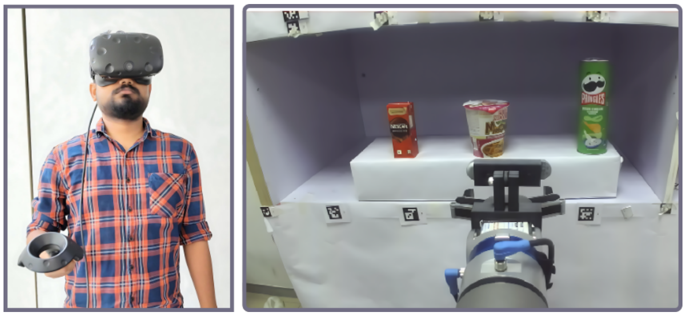

Fig. 1: (left) Operator wearing the Head Mounted Display along with handheld controllers. (right)A mock retail shelf unit along with the robot end-effector. It contains 3 distinct grocery items and the operator is tasked with picking them via teleoperation.

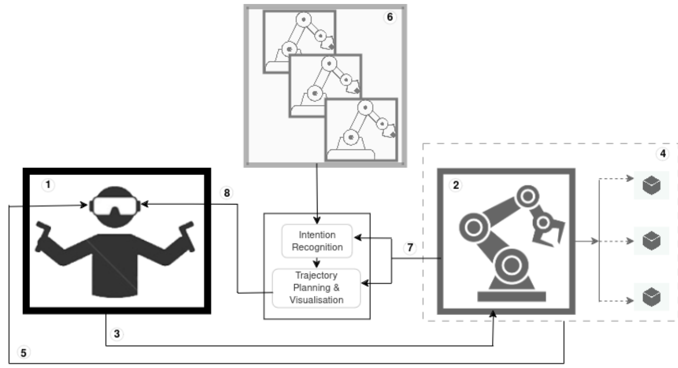

Fig. 2: The basic system architecture decomposed into smaller units. The interaction between them allows for efficient teleoperation of the robot while receiving real time goal prediction and visual assistance.

novel and efficient operator's intent prediction algorithm, which is further combined with the Augmented Reality (AR) feature to provide real-time visual feedback to the operator via HMD. The details of the methods used are provided in the subsequent sections.

III. Methods

Figure 2 illustrates the system architecture and interactions between various sub-units. The operator (1), wearing an HMD and using a joystick, controls the robot (2). A motion capture system translates the operator's hand movements into robot end-effector actions via inverse kinematics (3). The environment contains the robot and three shelf-mounted goals represented as (4), with a fixed stereo camera continuously relaying visual feedback (5) of the robot end-effector and goals to the operator's HMD. An offline motion library (6) stores preprocessed trajectories based on human demonstrations. The Intention Recognition Module combines this library with the current end-effector position (7) to predict the operator's intended goal. The Trajectory Planning & Visualization Module then plans a path to the predicted goal and generates AR markers. These markers are overlaid on the environment view in the operator's HMD (8), guiding them towards the goal. This integrated system facilitates efficient teleoperation by combining real-time feedback with predictive assistance. The processes and subsystems involved are described in the following subsections.

A. DMP

Dynamic Movement Primitive (DMP) is a popular framework for direct learning from demonstration [9]. It utilizes a system of second-order ordinary differential equations corresponding to a spring mass damper system as given in (1), where a forcing function $f(s)$ is learnt to encode the desired demonstrated trajectory. Further, the learnt DMP models is used predict future goals in real-time. We use the DMP model from [10] because of it's robustness. When dealing with d-dimensional trajectories, we get the following vector formulation :

$$\tau \dot{v} = K(g - x) - Dv + (g - x_0)s + Kf(s) \tag{1}$$

$$\tau \dot{x} = v \tag{2}$$

where $x, v, g, x_0, f(s) \in \mathbb{R}^d$ represent the position, velocity, terminal position, initial position and forcing function respectively. $K, D \in \mathbb{R}^{d\times d}$ are diagonal matrices analogous to spring and damping terms, to maintain each component decoupled from the others. $f(s)$ is a real valued non-linear forcing term which can be written as a linear combination of weighted basis functions.

$$f(s) = \frac{\sum_{i=0}^{N} \omega_i \psi_i(s)}{\sum_{i=0}^{N} \psi_i(s)} \tag{3}$$

ω_i are the weights learned via a regression method while $\psi_i(s)$ are conventionally Gaussian type Radial basis functions. s is a re-paramtrization of time governed by the canonical system $\dot{s} = -\alpha s$ where $\alpha \in \mathbb{R}^+$ and $s(0) = 1$.

However a DMP learned from a single demonstrated trajectory does not take into account the mismatch in starting and ending points which arise due to difference in personnel teleoperating the robot. So it becomes imperative that we induce variability in demonstrations and yet somehow extract a common behaviour. In order to achieve this, we collect a set of demonstrations from various users for each goal involved in the task. We then perform classical regression over the set of collected trajectories to learn the parameters of a common DMP which is non-probabilistic in nature [10]. This common DMP corresponding to each goal acts as the set of of offline DMP motions and are stored as a part of offline motion library which is used in the Intention Recognition algorithm as described below.

B. Intention Recognition

We formulate a novel intention recognition algorithm. It takes in the set of offline learnt DMP motions and predicts the goal that the user is trying to reach. The set of offline DMP motions is represented as $y_{dmp}(g)\ \forall g \in g'$, where g' represents the set of goals. Based on this, the real time online trajectory planner plans a path between the current position of the robot end effector and the predicted goal. This path is then visualised as a set of AR markers which then aids the user in completing the task. The entire end-to-end method is described in Algorithm 1 with its various sub-methods described below.

Algorithm 1 Goal Prediction Algorithm

Require: Set of Offline DMP Motion : $y_{dmp}(g)\ \forall g \in g'$,
  Initialise empty buffer : $\vec{y_i} = [\ \ ]$
while $tasknotcompleted$ **do**
  $\vec{y_i} \leftarrow$ Current state of the robot end effector
  Append $\vec{y_i}$ to buffer to get $y_i = [\vec{y}_i^1, \vec{y}_i^2, \cdots, \vec{y}_i^K]$
  if buffer is full **then**
    for $g \in g'$ **do**
      $p^{dmp}(y_i|g) \leftarrow 1.0$ Reset Probabilities
      $C_{user}(y_i, y_{dmp}(g)) \leftarrow$ Compute using (6)
      $p^{dmp}(y_i|g) \leftarrow$ Compute using (5)
      $p^{dmp}(g|y_i) \leftarrow$ Compute using (7)
    end for
    Predicted Goal is : $\arg\max_g p^{dmp}(g|y_i)$
    Reset buffer as : $\vec{y_i} = [\ \ ]$
  else
    continue
  end if
end while

1) Trajectory Ratio: The DMP-encoded offline trajectory is represented as $y_{dmp} \in \mathbb{R}^{3\times N}$; where N is the number of samples which serves as trajectory length determined by the sampling time. Goal prediction uses two inputs: offline DMP-learned actions and real-time user motion. Since comparison of these two temporally misaligned motions is challenging, we introduce 'trajectory ratio' $(t_r) \in \mathbb{R}^+$, which is a factor of how much the current motion has elapsed compared to the offline learned motion.

$$t_r = \frac{||\vec{y_i} - y_0||}{||y_g - y_0||} \tag{4}$$

where $\vec{y_i} \in \mathbb{R}^3$ is the current user input (the robot's end effector position) whereas $y_0, y_g \in \mathbb{R}^3$ are the initial and final(goal) position of the offline learned trajectory as encoded by the DMP.

2) Finding appropriate offline DMP motion: Once the trajectory ratio is known the offline learned trajectory y_{dmp} is scaled by it to get an appropriate comparison reference. The determined motion is represented as $\vec{y}_{dmp}(g) \in \mathbb{R}^3\ \forall g \in g'$ and is computed as per Algorithm 2.

3) Moving Window Approach: Contrary to the approach in [11], which uses the entire history of user inputs to predict a goal, we focus on a snapshot of recent history. The previous approach suffers from the issue that when goals are changed dynamically while performing the motion, it is unable to predict the new goal because the approach renders the probability for the old goal to be close to 1 while the probability for other goals becomes close to 0. The history of probabilities in the multiplicative factor which were close

Algorithm 2 Find DMP Motion for appropriate comparison

Require: Set of Offline DMP Motion : $y_{dmp}(g)\ \forall g \in g'$,
  Trajectory Ratio : t_r
for $g \in g'$ **do**
  if $t_r < 1$ **then**
    $uptoIndex \leftarrow \lceil t_r \times length(y_{dmp}(g)) \rceil$
    if $uptoIndex = 0$ **then**
      $\vec{y}_{dmp}(g) \leftarrow y_{dmp}(g)[0, :]$
    else
      $scaledTrajectory \leftarrow y_{dmp}(g)[0 : uptoIndex, :]$
      $\vec{y}_{dmp}(g) \leftarrow scaledTrajectory[end, :]$
    end if
  else
    $\vec{y}_{dmp}(g) \leftarrow y_{dmp}(g)[end, :]$
  end if
end for
return $\vec{y}_{dmp}(g)\ \forall g \in g'$

to 0 reduces the overall probability even when confidence is high for the new goal. To address this, we propose a 'Moving Window' that considers only the recent K samples stored in a buffer. User cost and probabilities are computed within this window, allowing the algorithm to make real-time predictions. Once the buffer of K samples is filled, it resets and continues. Since the entire system is running at a very high frequency the goal prediction works real time while also adapting to change in user intention.

4) User Cost and Goal Probabilities: The probability that a given recent history of user inputs given a goal g is given by:

$$p^{dmp}(y_i|g) = e^{-C^2_{user}(y_i, y_{dmp}(g))} \tag{5}$$

where C_{user} is the user cost defined in (6). Here y_i represents the collection of current user inputs in the current window which is represented as a buffer of K samples. We capture the history of recent user inputs in a single pass of a window.

$$C_{user}(y_i, y_{dmp}(g)) = \sum_{k=1}^{K} \Bigg[\arccos\left(\frac{\vec{y}_i^k \cdot \vec{y}^k_{dmp}(g)}{\|\vec{y}_i^k\| \|\vec{y}^k_{dmp}(g)\|} \right) + ||\vec{y}_i^k - \vec{y}^k_{dmp}(g)|| \Bigg] \tag{6}$$

The user cost C_{user} is computed over the entire length of the buffer. It is a combination of the cosine similarity between the input and the offline motion as well as the Euclidean distance between them. Here $\vec{y}_i^k$ and $\vec{y}^k_{dmp}(g)$ represents the current user input and the appropriate offline DMP motion for k^{th} instance of the buffer (moving window).

Lastly via Bayes' rule, the probability of an object being the goal g for a given set of goals g' given the recent history of user inputs captured in a buffer is given as :

$$p^{dmp}(g|y_i) = \frac{p^{dmp}(y_i|g)p(g)}{\sum_{g'} p^{dmp}(y_i|g')p(g')} \tag{7}$$

C. Trajectory Planning and Visualization

The trajectory planning process integrates two key inputs: the goal predicted by Algorithm 1, and the current Cartesian state of the robot's end effector. We employ a relatively simple Minimum Jerk trajectory model [12] to chart a path between the current position and the predicted goal. This trajectory, represented as a series of sequential Cartesian waypoints, is then transmitted to the Visualization Module. This module processes the scene captured by a stationary camera, which encompasses the operational environment (including the end effector and the goals on the shelf). The module then augments this scene by overlaying AR markers along the trajectory generated by the planning module. The resulting visualization presents the operator with a composite view: the static environment overlaid with AR markers that serve as visual guides for goal attainment.

IV. Experimental Study

We conducted an experimental study with human participants performing a teleoperation task using three distinct methods. The study aimed to evaluate the effectiveness of these methods and measure user satisfaction through both objective and subjective metrics of human-robot collaboration. The three methods were as follows:

1) Full Teleoperation without Markers (FTW): In this method, the operator's pose is directly mapped to the robot's joint angles using inverse kinematics (IK). No goal prediction or assistive AR markers are used to aid in goal reaching.
2) Full Teleoperation with Markers (FTM): This method guides the robot to the goal using goal prediction techniques and assistive AR markers.
3) Variable Autonomy (VA): This hybrid approach includes all the features of FTM, with the added ability for the user to switch between pure teleoperation and full autonomy. When full autonomy is granted, the robot can execute the planned path and retrieve the desired item from the shelf independently. So the operator has markers to guide them, as well as the ability to switch autonomy.

We conducted a within-subjects study (N=7, all male, mean age < 25) focusing on a goal-picking task where participants retrieved one of three distinct items from a shelf. The experimental setup is illustrated in Fig. 1, with details provided in previous sections. To mitigate hardware unfamiliarity, participants were allowed brief trial runs and were briefed on the post-trial questionnaire. Each trial began with a button press to activate the hardware, followed by the retrieval attempt. Trials concluded when the subject placed the target object in a designated bin. Post-trial, we prompted the participants with questions for subjective assessment. The study comprised 15 randomized trials per participant, with five trials per method (FTW, FTM, VA) and randomized item selection.

We employed both objective and subjective metrics to assess system performance. Objective measures included trial completion time and number of collisions while reaching goals. Lower values in these metrics indicate better performance. Subjective measures were obtained through participant surveys using 7-point Likert scales [13], where 1 represented "strongly disagree" and 7 represented "strongly agree". After each trial, participants rated the following statements adapted from [14] and [15]:

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)

V. Results

The effectiveness of our goal prediction algorithm is demonstrated in Figures 4, 5, and 6. These plots represent separate trial runs for each of the three goals using the FTM method, with markers indicating the path from the current end effector position to the predicted goal. As the probability for the predicted goal converges to 1, signifying increased confidence, the probabilities for other goals approach 0. Concurrently, the user cost for the predicted goal stabilizes with minimal variation, while costs for alternative goals increase significantly. Figure 3 illustrates snapshots from trials for each goal at various time intervals. Trajectories are computed continuously, with AR markers placed to guide the operator towards the goal. These markers dynamically adapt to changes in end effector position. The number of markers adjusts according to the remaining distance, enhancing user experience. Figure 7 demonstrates the advantages of our goal prediction algorithm over that presented in [11]. When the operator dynamically switches from goal 1 to goal 2, our method successfully adapts, predicting the new goal accurately. In contrast, while the comparative method initially predicts the original goal correctly, it fails to account for the switch, resulting in all probabilities dropping to 0.

The objective and subjective performance metrics are summarised in Table I. The means and standard deviations of trial times for the three methods are shown in Figure 8. The standard deviation for the three methods are $13.78, 18.29$ and 16.4 seconds respectively. Repeated measures ANOVA was used to compare the task completion times and number of collisions. The differences between the three methods for trial times were not statistically significant ($F = 0.46$, $p > 0.5$). The median number of collisions were $[1, 1, 0]$ for [FTM, FTW, VA] respectively. These differences were statistically significant ($F = 15.10$, $p < 0.001$). Tukey's HSD test for pairwise comparison showed that VA had significantly lower collision rates than FTM ($p = 0.001$) and FTW ($p = 0.001$). Subjective metrics for this user study were Likert items correspond to usability (q1), feeling of control (q2), and responsiveness (q3). These are shown in 9. These differences were not statistically significant ($p > 0.5$ for each item using repeated measures ANOVA).

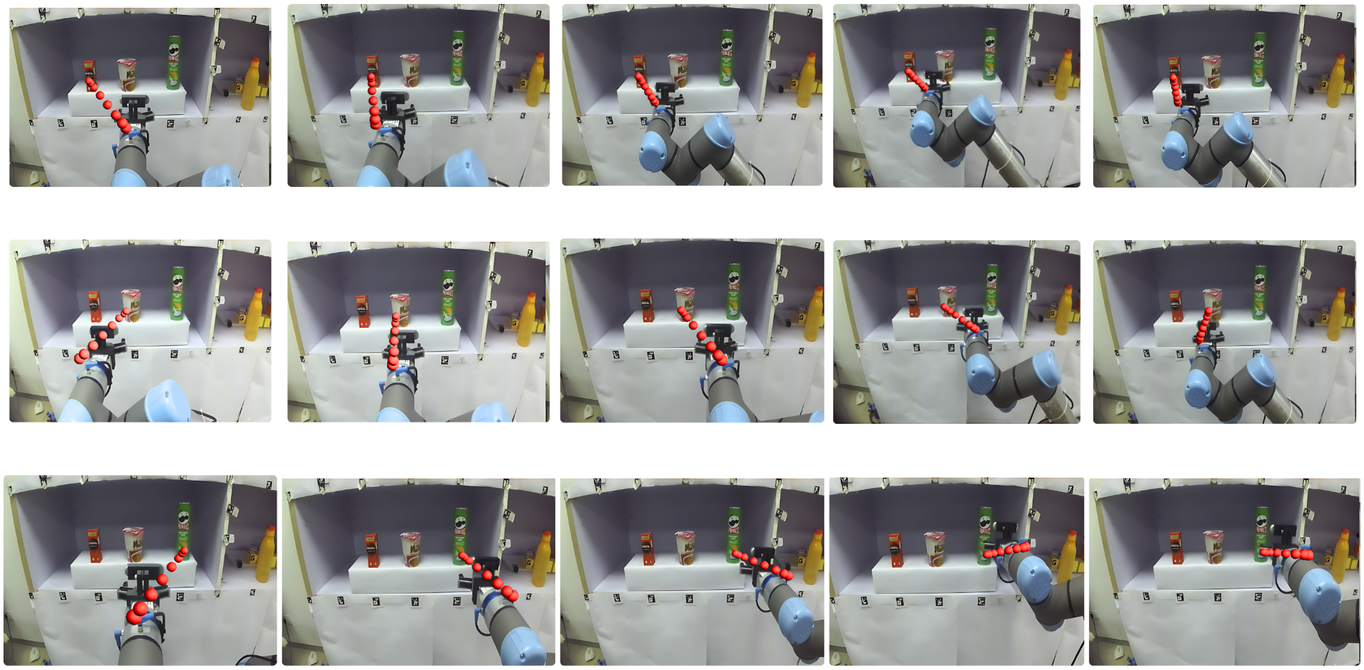

Fig. 3: Snapshots show the operator's perspective at different times using the FTM method. The shelf contains three goals (items left to right). Dynamic AR markers overlay the feed, adapting to the robot end effector's position relative to the predicted goal. Each row represents teleoperation towards one of the three goals, demonstrating AR guidance throughout task execution.

METHOD	Objective		Subjective		
	Trial Time(s)	Collisions	q_1	q_2	q_3
FTW	39.33	1	5	6	5
FTM	40.59	1	6	5	5
VA	44.64	0	6	6	5

TABLE I: Comparison of Methods with Objective and Subjective Measures

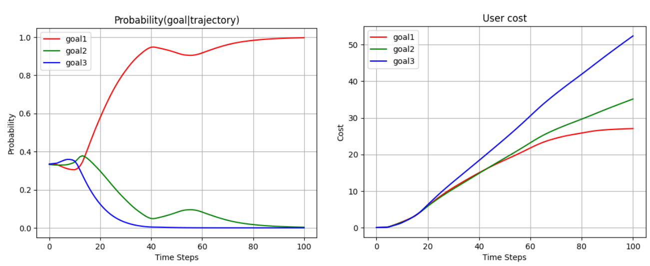

Fig. 4: Prob. & Cumulative User Cost for goal1

VI. Conclusion

This study aimed to enhance task retrieval from retail store shelves via teleoperation, addressing a significant challenge in remote manipulation. We developed a novel intention recognition algorithm capable of predicting the operator's desired goal well in advance. To reduce cognitive strain, we implemented adaptive AR markers that provide real-time guidance to operators in achieving their goals. Our goal prediction algorithm demonstrated high accuracy and real-time responsiveness to changes in operator intentions. We conducted a user study involving 7 participants, each

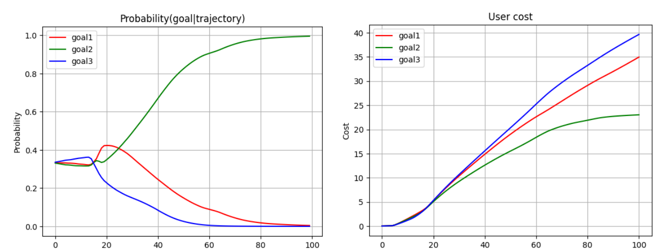

Fig. 5: Prob. & Cumulative User Cost for goal2

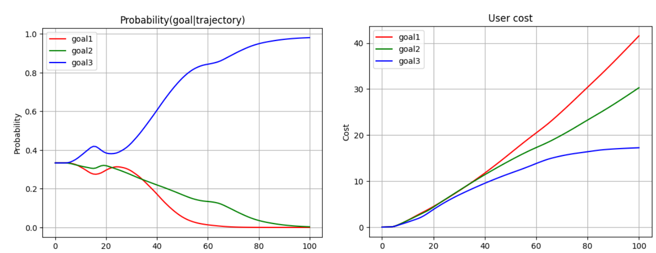

Fig. 6: Prob. & Cumulative User Cost for goal3

performing 15 trials under three distinct methods (FTW, FTM and VA), as detailed in section IV. These methods were evaluated using both objective and subjective performance metrics. While we initially expected that Variable Autonomy (VA) would outperform the alternatives, our findings revealed a more nuanced picture. Variable Autonomy excelled in reducing the number of collisions, making it particularly well-suited for enhancing operational safety and boosting

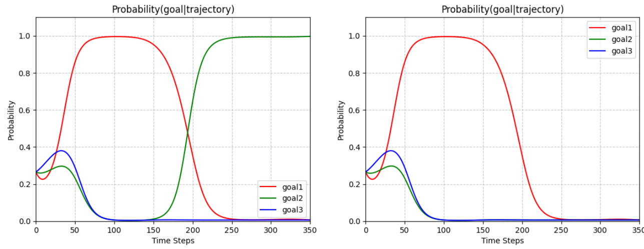

Fig. 7: The efficacy of our algorithm compared to [11] under goal switching condition. Left: Our algorithm updates the goal confidence when the user switches from goal 1 to goal 2. Right: The baseline algorithm predicts correctly for goal 1 but when switched to goal 2, the confidence is not reflected.

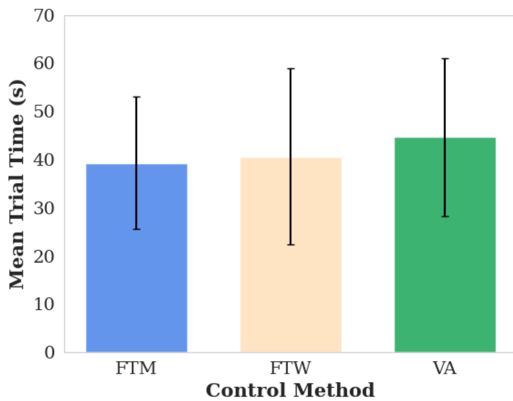

Fig. 8: Task completion times for the three control methods (FTM, FTW, VA).

novice users' confidence. For other performance metrics, the differences between methods were not statistically significant, suggesting that the effectiveness of our proposed approaches may be context-dependent. These results underscore the complexity of teleoperation systems and highlight areas for future investigation.

For future work, we aim to introduce more complex retail shelf environments with a greater number of objects and cluttered spaces to test the robustness of our methods. We also plan on enhancing the Augmented Reality interface to be more user-friendly and intuitive, with improved graphics for better operator guidance. Additionally we plan on conducting larger-scale studies and introducing additional metrics to uncover potential nuances in the effectiveness of our proposed methods. In conclusion, while Variable Autonomy did not universally outperform the alternative methods as initially hypothesized, our study provides valuable insights into teleoperation in retail environments. The superior performance in collision reduction demonstrates a clear advantage in certain contexts, laying a strong foundation for future research and development in this critical area of robotics and human-machine interaction.

Acknowledgements

We would like to thank the robotics team at TCS Research for their inputs and contributions to the user study.

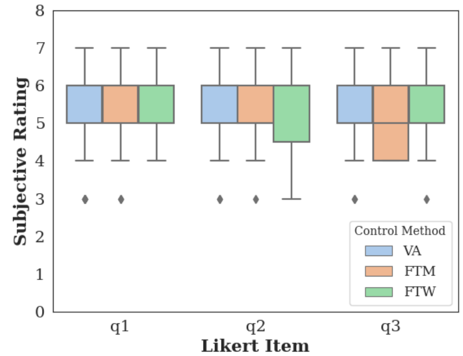

Fig. 9: Subjective ratings on 7-point Likert items to the three survey questions, split by control methods (VA, FTM, FTW).

References

[1] R. Sakai, S. Katsumata, T. Miki, T. Yano, W. Wei, Y. Okadome, N. Chihara, N. Kimura, Y. Nakai, I. Matsuo, and T. Shimizu, "A mobile dual-arm manipulation robot system for stocking and disposing of items in a convenience store by using universal vacuum grippers for grasping items," *Journal of Automation*, 2024.

[2] C.-Y. Su and H.-C. Wang, "Development of an autonomous robot replenishment system for convenience store," *Automation in Retail*, 2024.

[3] J. Luo and W. Liu, "A vision-based virtual fixture with robot learning for teleoperation," *International Journal of Robotics Research*, 2024.

[4] J. Nakanishi, S. Itadera, T. Aoyama, and Y. Hasegawa, "Towards the development of an intuitive teleoperation system for human support robot using a vr device," *Journal of Robotics*, 2024.

[5] M. Gallipoli, S. Buonocore, M. Selvaggio, G. A. Fontanelli, S. Grazioso, and G. Di Gironimo, "A virtual reality-based dual-mode robot teleoperation architecture," *Robotica*, pp. 1–24, 2024.

[6] D. Nicolis, M. Palumbo, A. M. Zanchettin, and P. Rocco, "Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 796–803, 2018.

[7] G. A. Garcia Ricardez, S. Okada, N. Koganti, A. Yasuda, P. M. Uriguen Eljuri, T. Sano, P.-C. Yang, L. El Hafi, M. Yamamoto, J. Takamatsu, and T. Ogasawara, "Restock and straightening system for retail automation using compliant and mobile manipulation," *Journal of Retail Automation*, 2024.

[8] A. Naceri, D. Mazzanti, J. Bimbo, Y. T. Tefera, D. Prattichizzo, D. G. Caldwell, L. S. Mattos, and N. Deshpande, "The vicarios virtual reality interface for remote robotic teleoperation: Teleporting for intuitive tele-manipulation," *Journal of Intelligent & Robotic Systems*, vol. 101, pp. 1–16, 2021.

[9] 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.

[10] M. Ginesi, N. Sansonetto, and P. Fiorini, "Overcoming some drawbacks of dynamic movement primitives," *Robotics and Autonomous Systems*, vol. 144, p. 103844, 2021.

[11] 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.

[12] R. Shadmehr and S. P. Wise, *The computational neurobiology of reaching and pointing: a foundation for motor learning*. MIT press, 2004.

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

[14] 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.

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