Shared Autonomy in Motion Mapping Teleportation

Trevor Rizzo, Nagarjun Vinukonda

***Abstract*—Teleoperation is a well known approach used for controlling the robot manipulator for different applications, but this causes fatigue in human while operating on difficult and monotonous tasks. Our goal is to implement a Semi-Autonomous function for controlling robot to decrease the fatigue. While Fully-Autonomous function reduces most of the fatigue it may also cause problems which will be discussed later in this paper. The novel aspect of our work is implementing Semi-Autonomous function for the application of our robot i.e. water pouring task. This projects results a statistical comparative analysis of Fully-Autonomous, Semi-Autonomous and Teleoperation using user study to determine better operation of operating nursing robot based on fatigue, time, efficiency and other parameters.**

I. Introduction

Teleoperation is the method of control of a device or machine remotely. Teleportation of robot is used in many applications in health, defense, nursing, etc. The main advantage of Teleoperation is that overall task control can rely on human perception, judgement, decision, dexterity, and training. The main disadvantage is that the human operator must cope with a sense of remoteness, be alert of and integrate many information, and coordinate the control of one or two mechanical arms, each having typically six degrees of freedom [1].

In order to reduce the fatigue while operating with a Teleoperated robot we can either implement either a Fully Automated or Semi-Autonomous function. Our purpose is to build a semi-autonomous teleoperated control of nursing robot at HIRO lab WPI as shown in Fig.1. The Fully operated robot has its perks such as ability of robots to perform very odd and repetitive tasks, to administer any medical treatment to patients, decreasing cost of human labour per year, etc. and especially at current times to take of patients (COVID-19). Even though they has many advantages they cant be trusted fully. For patients safety measures we require human presence. A robot doesn't have a Human feelings it simple executes program provided. Nursing a patient require not only requires a diligent and precise work but also to understand patients problem and execute properly. Hence, this paper focuses on Semi-Autonomous robot, where there exits human presence and also reduction of fatigue through making autonomous at the moment of intricate tasks while teleoperating the robot.

This paper focuses on building semi-autonomous function on water pouring application as it represents a basic task nurses must perform on a daily basis, requires both fine and gross motor skills, it is a motion that require precision, not a trivial action for humans and it is a easily learned skill.

We aim to conduct this project based on user study, evaluating Semi-autonomous function against teleportation and Fully Autonomous. This project results using Baxter simulation, where its simulation is provided in reference along with its package implementation in further sections.

Fig. 1: Nursing Robot

II. Literature Review

The related work considered in this project is focused on two topics 1) Pouring water with robots, 2) Shared autonomy systems.

Many of the existing works detailing robots pouring water focus on pouring specific volumes of liquids. Additionally, many of the papers do not focus on the grasping manipulation of the object being poured from.There are no existing works focused on using shared autonomy to pour water. Do and Burgard, as well as Dong et al. focus on using robots to preform precise volumes of liquids. These works focus specifically on the act of pouring and not the actions of choosing and moving to containers [2], [3]. These articles are useful in detailing the parameters that need to be considered when attempting to pour liquids accurately. We hope to improve upon the design of Do and Burgard by not requiring the use of transparent containers for our liquids [2]. Guevara et al. focus on reducing the amount of liquid a robot spills while pouring a liquid. This work was only tested in simulation and has not been applied to a physical robots [4]. Minimizing spills may be beyond the scope of this project, but it would be a great addition to our body of work if time allows.

The shared Autonomy Teleoperation has been used in many applications. Shared Autonomy came in development after discovery of Teleoperation back in 1950's for space probes exploration. The first shared Autonomy research was conducted for handling radio Active materials [5]. Later, the research development in its area started slowly from 1990's. Today, shared Autonomy is used in many applications: to assist user to control robotic arms [6], where they use semi-Autonomous hybrid BCI using EEG, eye tracking for visual feedback and computer vision to control a robotic upper limb prosthetic. [7] uses predict then blend methods for shared control of wheel chair. [8] uses shared Autonomy for grasping. [9] Uses Virtual fixtures to project user commands onto path constraints to decrease operator fatigue in surgical settings.

Our mode of evaluation and main work is considered similar to an article that we referenced [10]. This paper introduces a probabilistic model called POMDP and contrasts with predict then blend method while conducting different experiments under user study. Although their method performs better in experiments, users prefer Blend method in few experiments. This might have a solution by decreasing speed while approaching an object or obstacle when using POMDP method which might have given users to have more feel of control. Similarly, [11] proposed a formalism for policy-blend method in an attempt to bridge the gap between less user input with more assistance and more user input with less assistance. Based on aggressiveness and other factors of users the robot is able to predict the users action.

[12] has implemented an Shared Autonomous control of camera viewpoint robot arm to control teleoperated robot arm picking objects through semantic, geometric and visual viewpoint explorations of environments using pretrained data-set and Aruco markers. Their evaluation is based on user study and shared-control camera method benefits the remote user's view without easing the manipulation.

III. Planned Methodology

Creating an improved method of controlling the nursing robot requires analyzing the existing control methods. To analyze the existing methods, the task of pouring water is discretized into individual motions. The identified motions for pouring a cup of water can be seen in figure 3. The fully teleoperation control and the fully autonomous control will be analyzed for each individual motion. For the teleoperation method, the operator's interaction with the robot controls will be analyzed. Interviewing the operator will provide insight into the mental and physical fatigue experienced. While controlling the robot, the operator's response time to simple mathematics problems can be used to quantify the mental strain. Additionally, an EEG can be used to measure the physical fatigue the operator experiences by measuring nerve activity. The time to complete each motion will also help indicate the efficiency of completing each motion for both the teleoperation and the autonomous method. The following are the discretized water pouring actions:

1) Identifying cups
2) Grasping Cups
3) Move cups towards each other
4) Pour
5) Stop pour
6) Place cups down

The shared autonomy control system will be derived from a fully autonomous system. The fully autonomous system will use computer vision to identify cups in the workspace. Each cup will be identified using an Aruco marker. Using the data collected from the image processing, the autonomous system will choose two cups and plan a path to pour water from one of the cups into the other. The autonomous system will also avoid any known obstacles, such as additional cups, in the workspace. The fully autonomous system will be analyzed based on the repeatability of each action and the computational load of each action.

The data from the analysis will be used to determine the structure of the shared autonomy system. Motions that require precision and motions that are mentally or physically fatiguing for the operator in teleoperation will be automated in the shared autonomy mode. Additionally, the motions and steps of pouring water that are computationally intensive, unreliable, or require operator choice will not be automated in the share-autonomy system.

IV. Adapted Methodology

The unforeseen circumstances of the COVID-19 pandemic altered the approach of creating a shared autonomy system with image processing capabilities. As, such the methodology followed over the course of this project was an adaptation of the proposed methodology. The adapted methodology focuses on creating a simulated environment for developing and testing shared autonomy methods. Using this simulated environment, autonomous, teleoperation, and shared autonomy systems are developed and tested. The proposed methods for developing a shared autonomy system are still adhered to in the adapted methodology.

V. Implementation

A. Aruco Tag Detection

Aurco tag detection provides a computationally simple way to detect object's location and orientation using image processing. Additionally, aruco tags allow or any object to be detect simply by apply a tag to it. Aruco detection was implemented in this project by using the openCV aruco library for python. The implementation of this library successfully identifies the position and orientation of aruco markers in real-time in both the real world and in the simulated environment as shown in Fig. 2.

B. Simulated Environment

The Baxter environment in the HiRo lab consists of the Baxter robot, a table used a work surface, and Solo cups that are used as the target of the robot's manipulations. The Baxter robot in the HiRo lab also has an intel RealSense RGB-D camera located on its head. In order to accurately test shared autonomy functions, all these components of the HiRo lab environment needed to be simulated.To mimic the Baxter robot

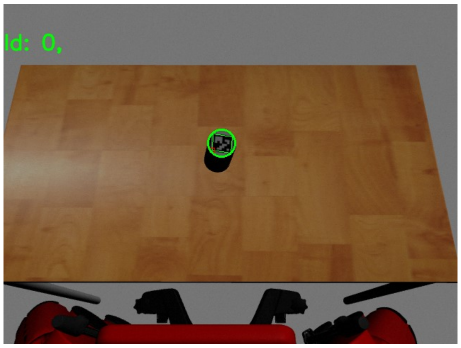

Fig. 2: Baxter simulation in gazebo

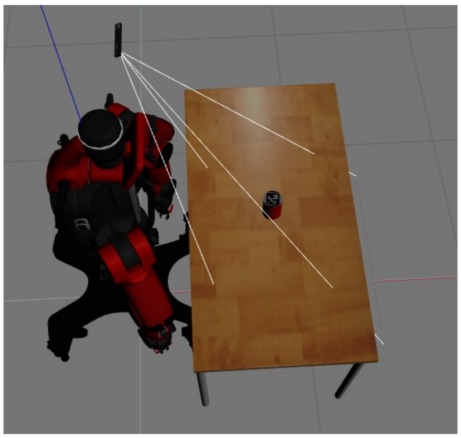

Fig. 3: Baxter simulation in gazebo

environment in the HiRo lab, a simulated environment was created using Gazebo simulation software as shown in Fig.3.

The starting point for this environment was the ”baxter_world” launch file included with the Baxter ROS package. This launch file launches an empty Gazebo environment containing only a simulated Baxter robot. The launch file also launches the ROS nodes necessary to send commands to the simulated Baxter robot. The new environment detailed in this section is launched using a new launch file ”baxter_depth.launch”, located in the same ROS package as the original launch file.

The Baxter robot in the HiRo lab has a intel RealSense RGB-D camera located on its head. This camera provides a top-down view of the robot’s workspace. This view is used for teleoperation and tasks that require images processing. To provide this capability, an open-source simulated real-sense was implemented in the Gazebo simulation. This simulated camera continuously publishes RGB, infrared, and depth images to separate rostopics. Implementing this camera required making minor changes to the source code to make it compatible with recent version of Gazebo. Once the simulated camera was implemented properly, it was added to the launch file. The chosen spawn position and orientation of the simulated RealSense camera mimics the viewpoint of the RealSense camera in the HiRo lab. This viewpoint can be changed by altering the spawn pose in the launch file.

In order to accurately simulated the robot’s interaction with the environment a cup was modeled for the robot to interact with. To provide support for image processing techniques, an aruco market was added to the top face of the simulated cup. To create an object to be simulated in Gazebo, an sdf file most be create. This file type allows the create to manually define the shape, mass and inertia of the simulated object. Additionally, sdf files can be used as wrappers to import objects defined by dae files into Gazebo. The later application of sdf files was chosen to create a simulate cup or use in Gazebo. To create the dae file, a cylinder with similar dimension to a Solo cup was modeled in blender. The cylinder was given a red coloring on all faces except the top face, which was textured with an image of an aruco tag. A custom sdf files was then written as a wrapper for the cup’s dae file. This sdf file also defined the mass and inertia of the cup.

To give the simulation a workspace for the Baxter robot to interact with the simulated cup, a table was added to the simulation. The table was taken from the standard Gazebo library of models. Using the launch file, the table and cup were spawned in locations that mimicked the workspace int he HiRo lab. The simulated environment created a workspace that allowed for the successful development of autonomous, teleoperational, and shared autonomy functions

C. Controlling the Baxter Robot

Developing a functions to autonomously move to a cup, pick it up, and pour it required understanding how to control the Baxter robot using ROS. The code provided in the Baxter ROS package controls the Baxter robot jointwise. Because the focus of this project was to use the end effector for manipulation, we developed a method for controlling the position and orientation of the Baxter robot’s end effectors. End effector based movement were achieved by implemented a modified version of the Baxter inverse kinematics service client. This code was mortified to return the joint angles required to achieve a given pose of an end effector. Using this code, the joint angles for any valid end effector pose could be calculated. Additionally, the ”endpoint_pose()” function was used to determine the pose of each limb’s end effector. To send the calculated joint angle to the Baxter robot, the joint angles were packaged into a ROS joint command and published using the ”set_joint_positions()” function. These three basic function were the foundation of the autonomous, teleoperation, and shared autonomy functions that were developed.

D. Autonomous Function

The first step in developing the shared autonomy functions was to implement a fully autonomous water pouring function. The autonomous function used the end effectors pose as the start point and the given pose (the cup’s pose) as the goal pose. The autonomous function moves the end effector long the shortest path the the goal pose and increments the end effector in steps of a specified length. The incremental poses are calculating by divided the total path length and rotation by the number of steps need to traverse the path. The methods described in the previous section are used to calculated and send the joint angles for the poses along the path. This simple autonomous function was implemented to show how shared autonomy can enhance even the most basic autonomous functions. To complete the act of pouring water, functions were created to raise the cup a specified distance and to pour the cup a specified angle. Once the autonomous function reaches its goal these to functions are called.

E. Teleoperation

A method of controlling the end effector's pose via teleoperation was developed so that it could be integrated into the shared autonomy functions and so that it could be compared to the shared autonomy functions. The developed teleoperation method is designed to be used with a flight-stick style USB gamepad. The game-pad inputs were interpreted by launching a ROS "joy_node" which listens for USB gamepad messages and publishes them to a rostopic. Then, a "joystick" class was implemented to subscribe to this topic and save the button and axis status's as variables. The joystick class was implemented with a flag to show if a new message had been received, meaning the state of the gamepad had changed. Using the "joystick" class and the Baxter control methods, and end effector pose-based teleoperation function was implemented. In this function, any time the joystick receives an input, the end effector's pose is changed based on that input. The translation of the end effector is based on three of the axis from the gamepad, one for each axis. The rotation of the end effector is based on two axis and one set of buttons on the gamepad,one for each euler angle. Each cycle, the variable position of the each of the gamepad's axes is multiplied by a constant and added to the respective component of the end effector's pose. The new pose is then converted to joint angles and published so the robot can move to the new pose.

F. Shared Autonomy

Using the methods developed in the autonomous and teleoperation functions, three unique methods of shared autonomy were developed. These methods all blended the autonomous and teleoperation approach. All of the shared autonomy functions conclude by calling the lift and pouring functions.

The first method of shared autonomy implemented is referred to as the "Take Control" method. In this method, the robot starts autonomously moving towards the goal pose. At any time, the user can press a button on the gamepad which disable the autonomous function and enable teleoperation. Once the user is satisfied with the pose of the end effector, the gamepad button can be pressed again. This action will return the robot to autonomous mode, where a new path to the goal pose will be created, starting at the end effector's new pose. This method of shared autonomy allows the user to take control of the end effectors movements without any input from the autonomous functions. This method also allows the user to pause the robot by putting the robot into teleoperation mode and providing no gamepad input. This may be useful in provided the user time to think about the robots path and actions.

The next method of shared autonomy that was developed is referred to as the "Meshed" method. In this method the robot starts autonomously moving towards the goal pose. Any time the user applies an input to the gamepad, the end effector is moved in the same way as it would be in teleoperation. When the user stops applying input to the gamepad, the autonomy resumes. The end effector's new pose will be the start pose of the new autonomous path. An additional feature added to this method is that, with the press of a specified gamepad button, the user has the ability to slow the end effector's movement while in autonomous mode. This method of shared autonomy allows for a smooth transition from autonomy to teleoperation. This smooth transition allows the user to more easily make small adjustment to the end effector's path.

The final method of shared autonomy implemented is referred to as the "Cone" method. This method is based on the "Meshed" method, but it limits the distance that he user can displace the end effector from the autonomous path. As the end effector moves closer to the goal, the distance the user can displace the end effector decreases proportionally. Allowing the user to have less control when the end effector is near the object may reduce the amount of unwanted collisions with the objects being manipulated.

VI. Proposed Experiment

To assess the effectiveness of the shared autonomy system, it will be compared against both the fully autonomous and fully teleoperation control systems. The experiment will involve pouring water into a stationary cup and pouring water into a cup being grasped by the nursing robot's end effector. To compare the shared autonomy and fully teleoperation, external participants will be used. The participants will attempt the experiment using both control schemes. The control scheme that each participant uses first will be alternated to avoid a bias. For each control scheme, the time to complete each movement will analyze. Additionally, to analyze mental strain, the participant will be asked to solve oral mathematical problems while completing each movement. Finally, an EMG will be used to measure muscle activity and approximate the physical fatigue felt by the participant. The participant will also be interviewed using a series of questions to gain insight into their experience with each control scheme. To compare the shared autonomy system to the fully autonomous system, the time and reliability of each motion will be compared.

If the shared autonomy system is successful, it should allow users to pour water quicker and more precisely than in fully teleoperation mode. This proves our main Hypothesis to be true:

Hypothesis: A shared-autonomy system will reduce the mental and physical fatigue on the operator while increasing the precision.

Our project is based on User study. We present few hypothesis for our experimentation:

H1: Using methods with autonomous assistance will lead to more successful task completions.

H2: Using methods with more autonomous assistance will result in faster task completion.

H3: Participants will agree more strongly on their preferences for the semi-Autonomous method compared teleoperation.

The experiment is proceeded in the following ways step by step: Identifying cups, Grasping Cups, Move cups towards each other, Pour, stop pour, Place cups down. It categorised into three parts:

1) Grasp 1st cup
2) Grasp 2nd cup
3) Pour the water

We consider about 20 users of equal gender. For each part of experiment the user is not provided knowledge of what method they are following. During the experiment, one of the cups is placed partially occluded in order for robot to teleoperate first and grasp the object. While controlling the robot, the operator's response time to simple mathematics problems can be used to quantify the mental strain. Finally, the task completion is evaluated based on few criteria.

A. Evaluation

We evaluate our experiment based on the following parameters:

1) **Success rate:** The chance of robot accomplishing the given task
2) **Total execution time:** The complete time taken for execution of task
3) **User Input:** The amount of control input given by user to move joystick
4) **User preference:** Evaluation based on users answers to questionnaire.

After completing tasks users were given a short questionnaire:

1) Did they felt control
2) The robot did want they wanted
3) Are they able to accomplish task quickly
4) Are their goals able to perceive accurately

VII. Initial Results

The initial results show that autonomous, teleoperation, and shared autonomy functions have been successfully implemented. Each of the shared autonomy functions has unique characteristics that will help provide insight into the types of shared autonomy that users prefer. The implemented shared autonomy functions are consistent and are ready to be tested on the real Baxter robot and in a user study. Additionally, the results show that the simulated environment adequately represents the HiRo lab workspace and can be used to develop many types of functions for the Baxter robot. Videos showing the results of the developed functions can be seen at the link provided in the submission folder.

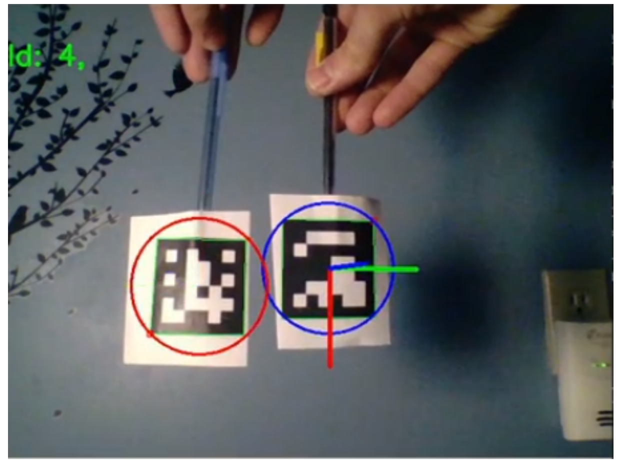

Fig. 4: Aruco marker tracking

VIII. Expected Results

In our project we are evaluating two different methods of shared Autonomy: ”Take control” and ”Mesh method” in contrast with Teleoperation and Fully Autonomous. For each trail we are going to evaluate based on evaluation criteria stated above and generate statistical graphs that are plotted for better comparative results. Using software called Anova we can generate graphs comparing teleoperation, Full Autonomy and Shared Autonomy against success rate, total execution time and user input. For user preference we plot graph of all the methods against rating for control, quickness, ability to reach goal and system they like. Overall we are expecting that users prefer shared Autonomy.

IX. Conclusion

The works detailed in this paper have shown that a simulated environment can be used to design and test shared autonomy functions. Along with the simulated environment, three shared autonomy functions, aruco marker detection, teleoperation, and autonomous functions were implemented and tested. A user study is needed to show what types of shared autonomy users prefer. The data from a user study could be used to further refine the work demonstrated by this project.

X. Future Work

Due to the current situations(COVID-19), we need to change our structure of project. For upcoming peers who want to work on simulation of Baxter can improve simulation by addition of water component and addition of Neural Networks for object detection without use of Aruco markers. They can utilise our Real sense plugins implemented in our simulation for visualisation and planning methods. the peer who wants to work on Hardware part of simulation can utilize our simulation for experimentation and complete the following tasks in future:

1) Testing on Baxter in HIRO Lab
2) Testing on different cups:
   Transparent
   Opaque

3) Developing better object detection method (Neural Networks)
4) Testing with different control speeds.
5) Analysing physical strain using EMG analysis measuring muscle activity.

REFERENCES

[1] R. Cortesão, R. Araujo, U. Nunes, and A. de Almeida, "Building a teleoperation environment," *IFAC Proceedings Volumes*, vol. 30, no. 3, pp. 173–178, 1997.

[2] W. B. Chau Do, "Accurate pouring with an autonomous robot using an rgb-d camera," 2018.

[3] S. K. T. S. Chenyu Dong, Masaru Takizawa, "Precision pouring into unknown containers by service robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[4] M. U. G. S. R. K. S. Tatiana Lopez Guevara, Nicholas K. Taylor, "Adaptable pouring: Teaching robots not to spill using fast but approximate fluid simulation," *1st Conference on Robot Learning (CoRL)*, 2017.

[5] R. C. Goertz, "Manipulators used for handling radioactive materials," *Human factors in technology*, pp. 425–443, 1963.

[6] K. D. Katyal, M. S. Johannes, S. Kellis, T. Aflalo, C. Klaes, T. G. McGee, M. P. Para, Y. Shi, B. Lee, K. Pejsa *et al.*, "A collaborative bci approach to autonomous control of a prosthetic limb system," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2014, pp. 1479–1482.

[7] K. Muelling, A. Venkatraman, J.-S. Valois, J. Downey, J. Weiss, S. Javdani, M. Hebert, A. B. Schwartz, J. L. Collinger, and J. A. Bagnell, "Autonomy infused teleoperation with application to bci manipulation," *arXiv preprint arXiv:1503.05451*, 2015.

[8] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, 2012, pp. 1–8.

[9] 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*. IEEE, 2005, pp. 1139–1144.

[10] J. A. B. Shervin Javdani, Siddhartha S. Srinivasa, "Shared autonomy via hindsight optimization," 2015.

[11] 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.

[12] D. Rakita, B. Mutlu, and M. Gleicher, "Remote telemanipulation with adapting viewpoints in visually complex environments," in *Proceedings of the Robotics: Science and Systems*, 2019.