Human Robot Interaction
Final Project Report
Study of User preference for Teleoperating Sawyer among three modes.

Introduction

Teleoperation is the term for operation of a machine, system or robot from a distance – as defined by Technopedia [5]. The distance involved could vary from millions of kilometers as in space applications, to centimeters like in microsurgery. Teleoperation is used in situations and areas where manual control is not feasible, is too dangerous or when it would be very expensive if humans were to operate the machine/robot manually.

Teleoperation has a wide range of applications. Some examples of the situations/areas where teleoperation becomes either necessary or very useful are-

1. Industrial Machinery – in hazardous environments. Like the construction of Object Shelter after the Chernobyl accident and removal of debris after hazardous rockslides.
2. Underwater - Exploration, search and recovery.
3. Space – Exploration, Maintenance etc.
4. Resource Industry – Power Line maintenance, Mining etc.
5. Nuclear / Chemical control plants.

Teleoperation most commonly involves human operator who is controlling the remote robot. This project involves designing and developing user interface for a robot for a specific task and coming up with testing strategies for the chosen task. In this project we aim to study user's preference of different modes of teleoperation. The robot that is chosen is Sawyer.
The goal of the project is to design a Graphical User Interface for the Sawyer Robot so that it can be teleoperated. Another goal is to explore which mode of teleoperation the operators prefer- Using a keyboard or the designed Graphical User Interface for the task of moving the end-effector from start to goal position to grasp an item. For this project two Graphical User Interfaces were developed one to control the joints of the robot and another to move the end effector's position.

Task for the project:

The task for experiment is to move the end effector's position from one position to a goal position which would be the location of the item on the table to be picked up. Participants will teleoperate the Robot using keyboard and then the Graphical User Interfaces. Time taken to reach the goal position will be recorded and analyzed to determine which mode of teleoperation takes longest and least amount of time to reach the goal position from the start position. Then they will be given a questionnaire about their experience of the two modes of teleoperation.

Robot chosen for this project:

The Robot chosen for this project is Sawyer. Figure 1 shows the basic overview of the hardware that makes up the Sawyer robot.

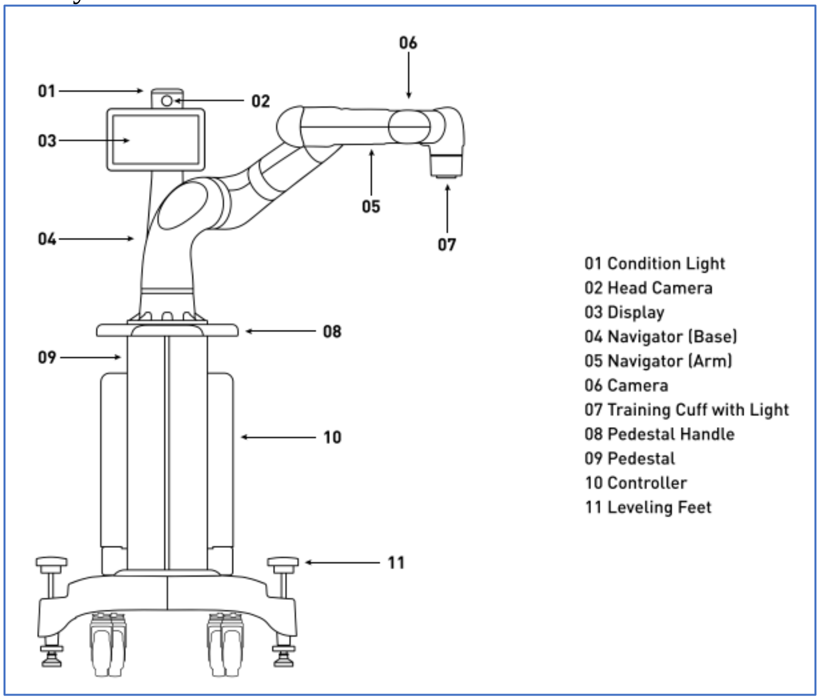

Figure 1: Basic overview of Sawyer hardware.

The joints are names J0 – J6.
The links are named L0 – L6.

The joints in the Sawyer are classified into two categories – roll or pitch by their movement.

1. Roll: J0, J2, J4, J6
2. Pitch: J1, J3, J5

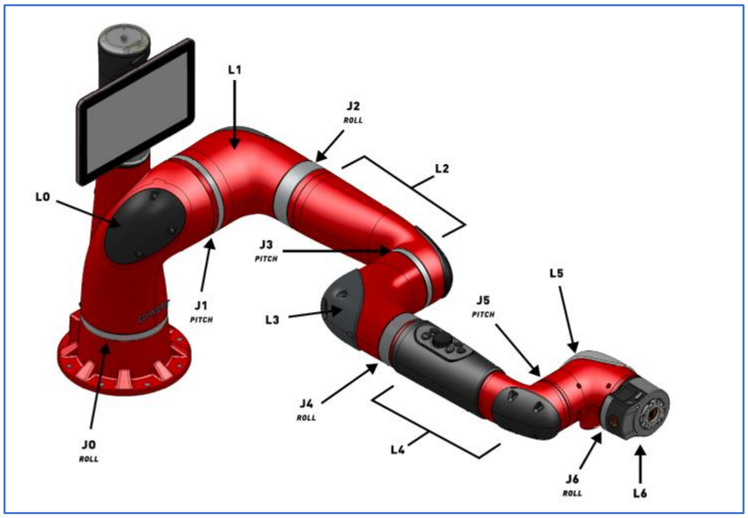

Figure 2: Joints and Links convention of Sawyer

Figure 2 shows the joints and link naming convention of the Sawyer robot. The joint and link naming for the sawyer arm starts at the base and increases incrementally up to the wrist.

Related work

Many Interface Designs have been developed to teleoperate robots in the recent years. However, I will be discussing mainly three papers which were designed and experimented on Sawyer Robot to maintain the relevance to this project.

1. Real-Time Joystick Teleoperation of the Sawyer Robot Using a Numerical Approach. [1]
   It is an open framework which allows a robot (Sawyer was used in their experiment) to be teleoperated in real time using joystick. The joystick acts as a velocity controller that drives the robot motion in cartesian space using a numeric approach to solve for inverse kinematics.

2. Virtual Reality Mediated Robot Teleoperation and Grasping. [2]
   Unity Game Engine based Virtual Environment is designed where user wears HTC Vive Headset and Manus Data Gloves. The objects are created in Virtual Reality. For each object in VR, an object and corresponding AR tag is presented in the real workspace. As user moves in VR, Sawyer follows the motions and when user picks up an object in VR, the Sawyer despite being kinematically dissimilar, follows the user's motion and picks up the corresponding real object.

3. Hands-Free: a robot augmented reality teleoperation system. [3]
   It is a vision based augmented reality system where users can teleoperate the robot's end-effector with their hands in real-time. This system user OpenPose [4] neural network to detect the human/operator's hand. The user's index position is extracted from the image and converted to real-world coordinates to move the robot's end-effector in a different workspace. The user hand skeleton is visualized in real-time moving in the actual robot workspace. This allows the user to teleoperate the robot intuitively, in real time.

   Hands-Free v2 to teleoperate robotic manipulators: include three axis precise positioning study
   This is an extension of the Hands-Free project by implementing z-axis control in the system. This version of the application integrates z-axis control and a trajectory planner.

Motivation:

Despite many teleoperation projects using Sawyer robot, I could not find a research which studied and developed a Graphical User Interface for the Sawyer robot. The sawyer robot is similar in design to many commercially available robotic arms like the Fetch or Baxter. These reasons motivated me to develop a Graphical User Interface for the Sawyer Robot and study how users prefer this mode of teleoperation.

Interface Design:

Since I could not find a baseline Graphical User Interface for teleoperating Sawyer, I designed two Graphical User Interfaces for the experiments and compared user preferences among them. The GUIs were developed with the goal to study which modes of teleoperation the user's preferred so interfacing the GUI with ROS node was given priority and not many features were kept in the GUI.
Some added features that I would prefer to have in the next iteration of the GUI:

- Camera view of the robot's arm, so that user can see what the robot is seeing.
- A status bar showing the status of the robot.
- An emergency stop button to stop the robot.
- Additional control to rotate the end-effector of the robot in GUI-2.

The Graphical User Interface 1- (Sawyer Joint Control)

The first Graphical User Interface Design was motivated by rethinkrobotics'[6] keyboard manipulation for controlling the Sawyer joints.

This Graphical User Interface is designed for controlling the joints of the Sawyer robot.

The 7 joints of Sawyer can be controlled by the J0 – J6 +/- buttons on the GUI.
The 'Close Gripper' and 'Open Gripper' buttons are for grasping and releasing an object.

This GUI captures the naming convention of the actual robot as mentioned in the 'Robot chosen for this project' section.

The joints in Sawyer are named from the base and so the buttons in this GUI are placed accordingly, so that it is more intuitive for the operators, who are already familiar with the manes of the joints.

This GUI was developed in PyQt 5.

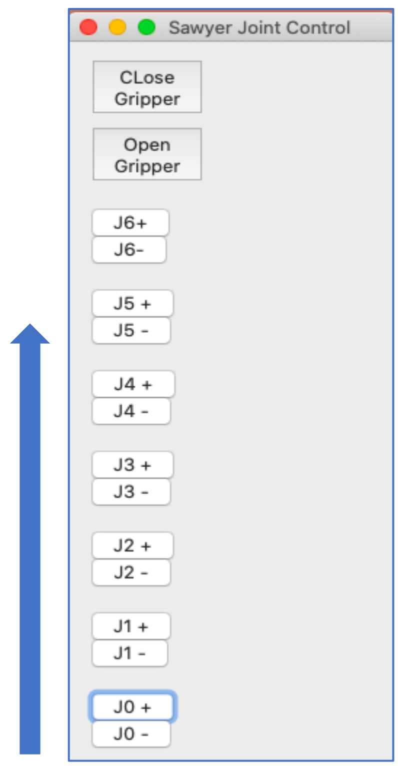

Figure 3: GUI-1 (Sawyer Joint Control)

The Graphical User Interface 2- (Sawyer Pose Control)

The second Graphical User Interface was designed to change the position of the end-effector of the robot.

It has ‘Close Gripper’ and ‘Open Gripper’ buttons like GUI-1, and buttons for the ‘x’, ‘y’ and ‘z’ coordinates for the end-effector.

The current pose of the end-effector is defined by:

- Position: (x, y, z)
- Orientation: (x, y, z, w)

One click is a small unit of movement in that coordinate (x, y, or z).

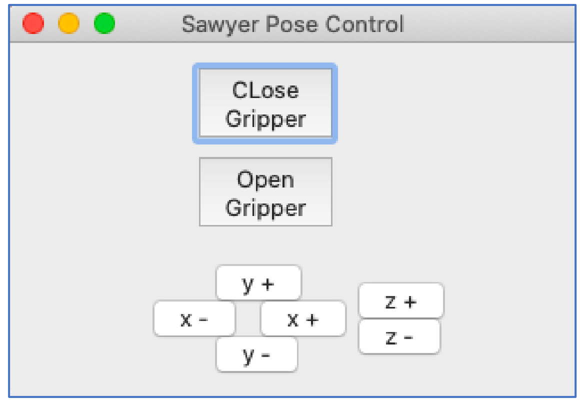

Figure 4: GUI-2 (Sawyer Pose Control)

The end-effector orientation was kept fixed for the experiments; however, one more button can be added in the GUI for rotating the end-effector. That way the two GUIs would have the same capabilities for teleoperation, and the comparisons done in the experiment would be even more relevant.
This GUI is also implemented in PyQt 5.

Experiments and Results:

The experimental set-up consisted of launching a Sawyer simulation from ROS on Gazebo. The Sawyer was interfaced to the PyQt GUI. All inputs to the GUI would be transferred to ROS nodes which would then be reflected on the Gazebo simulation.
The Gazebo world consisted of the Robot, a café table and a coca-cola can on the table.
The task given to the participants was to move the arm from a starting position to the coca-cola can in a way that the end-effector is able to grasp the can.

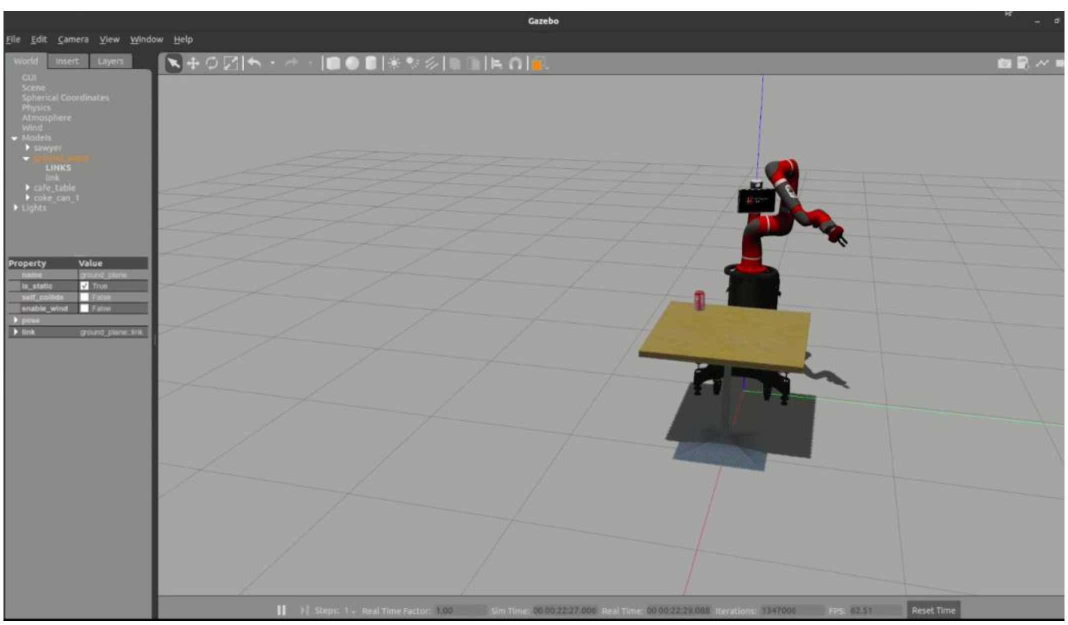

Figure 5: The environmental set-up in gazebo world.

There were 5 participants in the experiments of this project. All participants were either college students or had an undergraduate degree. The participant demographics were:

- 1 CS PhD student
- 1 CS undergrad student
- 1 Management PhD student
- 1 Environmental Science undergrad student
- 1 Corporate Finance

The participants were first asked how familiar they were with Robots and ROS in general. Then they were asked to try out each teleoperation mode (Keyboard, GUI-1 and GUI-2), without the table and item in the environment first, so that they can get an idea of the controls.

The participants were then asked to teleoperate Sawyer in different modes, in the following order-

1. Keyboard Teleoperation mode.
2. Graphical User Interface for Joint Control Mode.
3. Graphical User Interface for Pose Control Mode.

The time taken by participants to complete the task was recorded to analyze results. Users' preferences for the modes of teleoperation was recorded after they had finished operating all three modes.

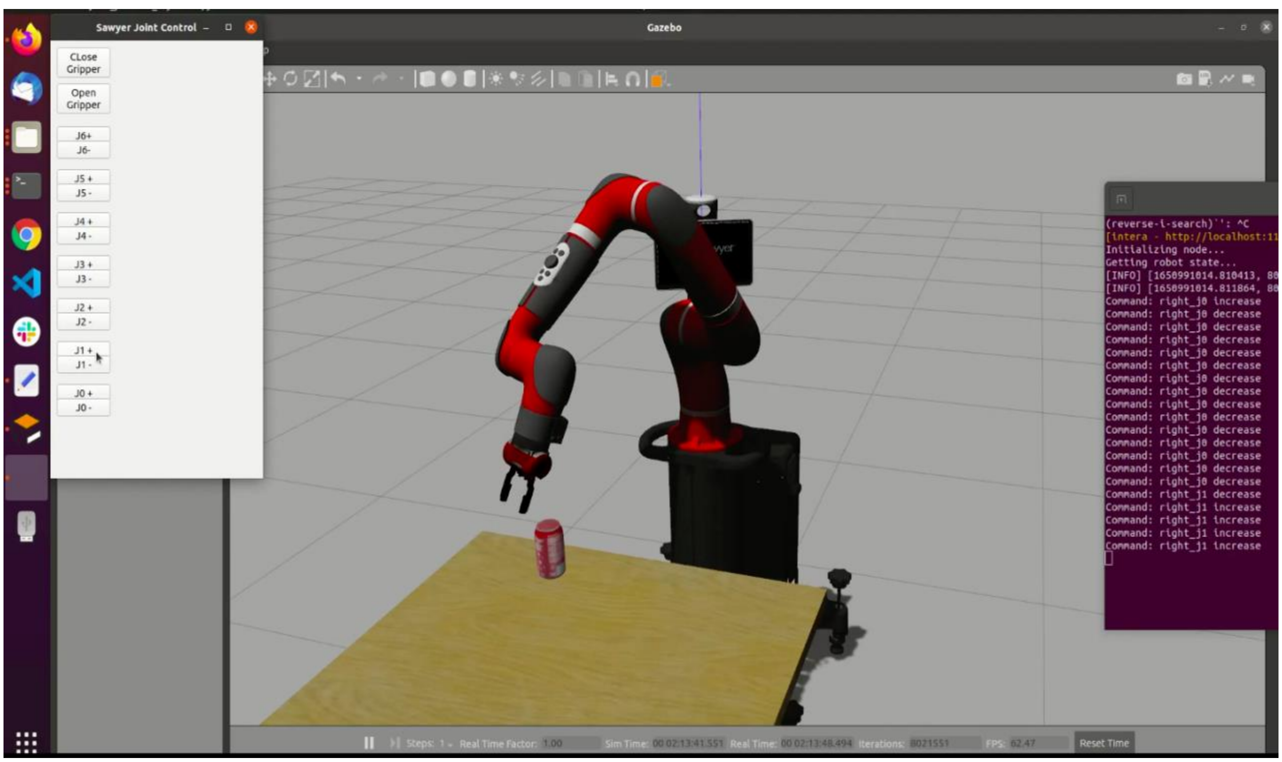

Figure 6: Teleoperating Sawyer using GUI-1 (Joint Control)

The following table shows the time taken by each participant for the modes of operation. Mean was calculated to get an idea of which mode was easier for the users in general.

	Keyboard joint	GUI-joint	GUI-Pose
P1	45	31	28
P2	65	50	30
P3	55	45	35
P4	105	95	40
P5	65	58	40
Mean	**67**	**55.8**	**34.6**

User's preference for the different modes of teleoperation-

	Keyboard joint	GUI-joint	GUI-Pose
P1	3	2	1
P2	3	2	1
P3	3	1	2
P4	3	2	1
P5	3	2	1

According to the user's preference for mode of teleoperation, we can draw the conclusion that Keyboard teleoperation mode was least preferred by the participants, and GUI-2 for Pose control was the one that users prefer most.

References

1. J. Cornejo, E. Denegri, K. Vasquez and O. E. Ramos, "Real-Time Joystick Teleoperation of the Sawyer Robot Using a Numerical Approach," 2018 IEEE ANDESCON, 2018, pp. 1-3, doi: 10.1109/ANDESCON.2018.8564651.
2. K. Chen, P. Dahal, M. Avagyan, K. Huang, " Virtual Reality Mediated Robot Teleoperation and Grasping" 2018.
3. C. Nuzzi, S. Ghidini, R. Pagani, S. Pasinetti, G. Coffetti and G. Sansoni, "Hands-Free: a robot augmented reality teleoperation system," 2020 17th International Conference on Ubiquitous Robots (UR), 2020, pp. 617-624, doi: 10.1109/UR49135.2020.9144841.
4. Cao, Z., Hidalgo, G., Simon, T., Wei, S., & Sheikh, Y. (2021). OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*, 172-186.
5. Teleoperation definition in technopedia. Technopedia.com
6. https://sdk.rethinkrobotics.com/intera/Joint_Position_Example#Joint_Position_Keyboard_Code_Walkthrough