Electromyography-Driven Hand Teleoperation

Cumulative Project Report

Abhi Gupta and Max Xu, Columbia University
COMS 6731 Humanoid Robotics, Professor Peter Allen

INTRODUCTION

Hand amputees struggle to adapt to a world designed for able-bodied individuals. A task as simple as grasping an object, such as picking up a glass of water, or as difficult as in-hand manipulation, such as adjusting grip to write with a pen, share the need for dexterity. Although the development of under-actuated, compliant hands is promising for their level of dexterity, we find ourselves at a dead-end when looking to effectively operate these devices. The challenge of teleoperation becomes twofold: finding an optimal control strategy for various robotic hands (1) while quickly and precisely maneuvering a robotic arm (2).

A. Hand Control Strategies

Two common methods to perform hand pose recognition include wearable (1) and remote (2) devices (Ciotti et al., 2016). The former comprises data gloves and surface-electromyography (sEMG) setups. Data gloves like the Cyberglove introduce a new layer of hardware between the user and his physical interactions with the environment in spite of providing tremendous accuracy. Thalmic Lab's inexpensive Myo armband is among many sEMG devices that rely solely on myoelectric activity–the way one's muscles contract in response to neural signals sent from the brain–for gesture recognition. Although sEMG is promising for its ability to tap into our neural pathways, estimating hand pose requires significantly better precision than what is sufficient for gesture classification. On the other hand, video recording represents the most commonly used strategy when using remote devices. Video and/or photo (RGB-D) information is processed for hand pose and gesture recognition with a combination of techniques such as k-means (Ong and Bowden, 2004), penalized maximum likelihood estimation (Cheng et al., 2012), and hidden Markov models (Beh et al., 2014). Convolutional Neural Network (CNN) encodings can effectively provide the hand information that we would expect to receive from a data glove (Bicchi et al., 2016).

In particular, we seek to further investigate how sEMG can bridge the gap between man and machine in the area of shape and grasp manipulation. Methods in computer vision limit the user's degree of mobility to a camera's field of view, unlike sEMG devices that can be mounted directly onto the user and thus do not suffer from the line-of-sight problem. The need for control robustness can be best described across relevant literature as the need for commercial myoelectric systems to be perfectly accurate in any situation. In spite of this challenge, a novel neural-interface technology introduced by Ctrl-Labs, a neuroscience startup founded by Thomas Reardon, Patrick Kaifosh, and Tim Machado aiming to translate mental activity into digital action, provides exciting opportunity to optimize and develop electromyography applications for robotics. Their EMG armband, Beast, comes packed with twice as many EMG channels (16) as the Myo. In addition, their intention capture engine can reconstruct hand pose from EMG, ultimately exposing the positions and motions of each finger joint. Finally, their recently updated software development kit (SDK) also now provides a measure of pinch and grasp forces in addition to gesture classification for a set of eleven discrete poses. These features suggest that Beast may be able to overcome the control difficulties traditionally faced with EMG in robotics when accompanied by perspective-shifting methods in machine learning. Our experience working with the Ctrl-Labs' device suggests that teleoperation is well within its capacity.

B. Arm Control Strategies

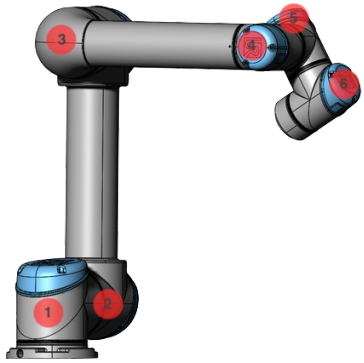
Mapping the user's wrist pose to the end-effector of an arbitrary robotic arm is necessary for completing pick and place tasks via teleoperation. Ascenion's Flock of Birds is a tracking system that can track the pose (i.e. position and orientation) of a small unit placed on the joint of interest relative to a magnetic base. Moving a robotic arm is then as straightforward as computing the inverse kinematics to position the end-effector to the given wrist pose in real-time.

Performing motion planning without introducing new hardware into the pipeline is ideal. The Flock of Birds reduces the flexibility and range of teleoperation by being tethered to a base. Therefore, the challenge becomes to effectively derive wrist pose from the information that Beast provides: joint states, EMG channels, and raw IMU values. Shen et al propose *AmTrak,* a system that fuses the IMU sensors and the anatomy of the human arm joints into a modified hidden Markov model (HMM) to continuously estimate position from orientation. Conventional methods like particle filters when applied to this state estimation problem take too long to converge to be practical. Shen et al. describe that for any given orientation, there exists a point cloud for the wrist and elbow joint, respectively, representing the combination of possible human arm positions as a function of five angles. Although the centroid of each point cloud proves to be a decent estimate of the wrist and elbow positions, employing Bayesian inference with an HMM can significantly improve this estimate. At the moment, the method is implemented to the extent that the estimates of the wrist position are the centroid of the point cloud at every timestep. The HMM framework will be further explored as this component becomes more relevant to teleoperation. Using the Flock of Birds should be sufficient to demonstrate a proof of concept.

C. Hardware Specifications

Designing an effective teleoperation method must take into account the variety of state-of-the-art hardware available in industry. Differing hand kinematics and physiological constraints between master and slave pairings can introduce challenges to teleoperation. Fully-actuated robotic hands–hands that possess as many degrees of freedom as the amount of joints–typically require control in high-dimensional spaces. Under-actuated hands simplify control complexity by controlling the same set of joints with fewer actuators. In spite of the trade-off in dexterity between the two kinds of hands, underactuation has gained traction over the years for the much-needed passive compliance it can provide for shape and grasp manipulation.

Robotic hands can also vary between those that perform top-down grasps exceptionally well like parallel jaw grippers and those that look very similar to our own hands. Anthropomorphic hands, robotic hands with human-like anatomical features, can be controlled in both a fully-actuated or under-actuated fashion. This project aims to teleoperate the Seed Robotics RH8D hand, an underactuated, anthropomorphic hand, and to generalize to other hardware specifications when possible. The Seed hand has 8 degrees of freedom: wrist rotation, flexion, and adduction (3 DOF), and index flexion, middle flexion, ring flexion, thumb flexion, and thumb adduction (5 DOF). The red dots in the first diagram below indicate the independently actuated joints on the Seed hand:



Any finger joint coupled immediately after an independently actuated joint is a child joint marked in green. By convention, the joint immediately succeeding the child joint is the grandchild joint marked in blue. While joints marked in red are fully actuated, joints marked in green or blue move a proportion of their parent joint.

The UR5 arm manufactured by Universal Robotics is a widely used robotic arm in both academia and industry. It will be teleoperated in conjunction with the aforementioned robotic hand. The second diagram above

numbers the six joints featured on the UR5 arm: shoulder pan joint (1), shoulder lift joint (2), elbow joint (3), and three wrist joints (4, 5, 6). The wrist rotation joint on the Seed hand was fixed as it provides the same function as the last wrist joint (6) on the UR5 arm. It is important to note that the UR5 can reach up to 850mm and work well within a payload of 5kg.

RELATED WORK

A. Investigating existing hand teleoperation methods

In parallel with developing with the Ctrl-Labs' hardware, the performance of existing hand teleoperation methods was investigated. Attention was first directed towards approaches that can generalize across both non-anthropomorphic and anthropomorphic hands. In particular, Meeker et al. proposed a low-dimensional and continuous teleoperation subspace as an intermediary for mapping between different hand pose spaces. Knowledge of the joint limits along three dimensions, the size, spread, and curl of the fingers as described in her work, are assumed. This teleoperation method was implemented end-to-end, but out-of-the-box was designed for the Schunk Hand–hardware we do not possess. Writing an XML script defining the joint limits of the Yale OpenHand Model O, a hand with four degrees of freedom fabricated during the 2018 New England Manipulation Symposium, was required to employ her algorithm on such underactuated hands.

As expected of the teleoperation method, it captured the size, spread, and curl motions of the human hand as closely as possible with the given kinematics of the slave hand. Because this continuous teleoperation subspace essentially emulates underactuation in software, it would ignore altogether the more fine-grain joint mappings that exist between different hand pose spaces. With this insight, joint and fingertip mapping teleoperation methods were explored.

Scarcia et al suggest that exclusively mapping joints between a human hand and a fully actuated 20 DOF robotic hand is insufficient for grasping objects as the fingertip positions between the two hand pose spaces need not match. Using the Leap Motion, a set of infrared sensors that can construct a model of the human hand, they iteratively improve the linear map that optimally projects the human hand into a known robotic hand space. In other words, the Euclidean distance between cartesian fingertip positions of the master and slave in the robotic hand space is minimized. This method can be seen as a way to calibrate a user's hand to an anthropomorphic robotic hand, including information about the dimensions of the human hand into the teleoperation architecture. While an interesting idea nonetheless, Ctrl-Labs only measures joint angles and not the lengths of the various finger bones required to implement this method. Introducing the Leap Motion as part of a teleoperation calibration procedure may prove to be effective.

B. Investigating existing teleoperation methods to move an end-effector

Many robotic arm teleoperation methods were explored as well. In order to track the movement of the human arm, M. Bergamasco developed an arm exoskeleton measuring the external forces exerted by the human operator (Bergamasco M., 1994). The arm can be as heavy as 10 kg, proving to be impractical for tasks as arduous as human teleoperation. In contrast, Tadakuma designed a master-slave robotic arm where the master possesed 6 degrees of freedom (DOF) while the slave had 7 (Tadakuma R., 2004). His arm, Telesar II, proposes a dual motion transmission method to achieve smooth teleoperation, however the master is once again far beyond realistic weight limits. Under these circumstances, Kim D. proposed a new minimalistic teleoperation method requiring only three sensors attached to the human arm to track the master. Although the performance is adequate, it is hypothesized that a far superior architecture can be developed to teleoperate an arm using even fewer sensors.

Electromyography-Driven Hand Teleoperation

Cumulative Project Report

Abhi Gupta and Max Xu, Columbia University
COMS 6731 Humanoid Robotics, Professor Peter Allen

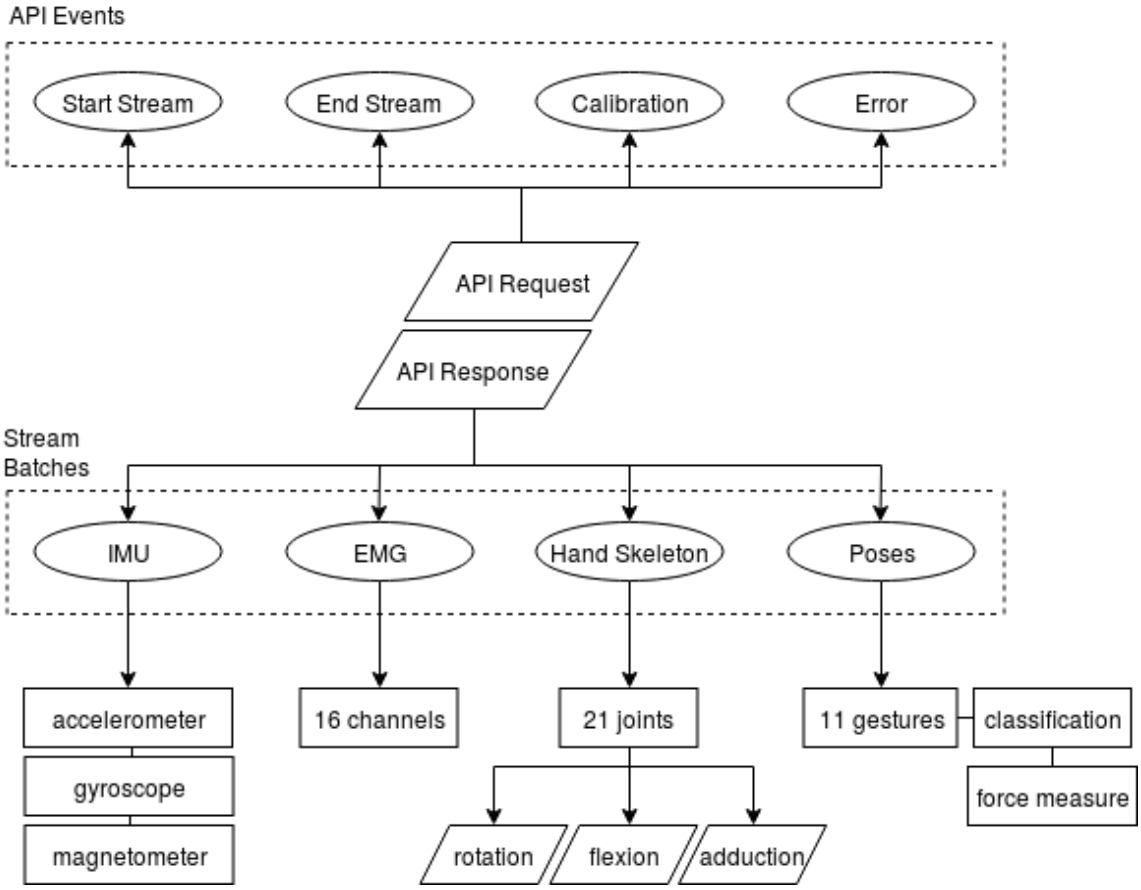
TELEOPERATION ARCHITECTURE

The teleoperation architecture is divided into two critical components: the teleoperation of the hand (1) and the teleoperation of the arm (2). These components were originally developed in isolation before coming together to provide the necessary functionality to pick and place objects on a table with the robotic hand-arm configuration. Hand teleoperation requires the Ctrl-Labs' device while arm teleoperation at the moment needs the FOB tracking system. Approximating the wrist pose as provided by the FOB with Beast remains a work in progress.

The user can expect to control the hand accurately and with minimal latency without worrying about performing gestures that the slave hand cannot match. In addition, collision between the hand and the environment is taken entirely into account by the hand, further reducing the burden of teleoperation on the user. While the end user can also neglect collision between the arm and its environment, the change in wrist position must remain minimal to prevent latency from robbing the teleoperation architecture of any performance. Latency in teleoperating the arm render grasping a variety of tabletop objects quite challenging because small changes in master wrist pose are not instantly reflected on the slave arm. What follows is a breakdown of electromyography-driven teleoperation of the Seed-UR5 robot.

A. Interfacing with the Ctrl-Labs' EMG Armband

The ROS wrapper for ctrl-release, the Ctrl-Labs SDK, previously developed in the Columbia Robotics Lab was used to extract joint angles of the teleoperator's hand. All of the information available from the armband is outlined below:



B. Debugging with the Cyberglove

The complexity of the armband would result in significant downtime when attempting to debug various aspects of the teleoperation pipeline. For example, the armband would often run out of battery or perform inadequately leading us to believe that there was a bug in our architecture when in fact it was a byproduct of using Beast. It is important to note that the Ctrl-Labs' device attempts to predict the joint states of the user's hand while the Cyberglove provides ground-truth, therefore it cannot be completely accurate. The Cyberglove acted as a substitute for the armband consequently, allowing us to efficiently scrutinize bugs within teleoperation without worrying about troubleshooting any hardware malfunctions. For these reasons, any teleoperation-related work can be interfaced with both the armband and the data glove with the -a and -d flags, respectively.

Unlike the Ctrl-Labs' device, the Cyberglove needs to be calibrated to the user's hand before use. Running the Cyberglove with the calibration flags as described in the documentation provided with this project will render the following calibration procedure:

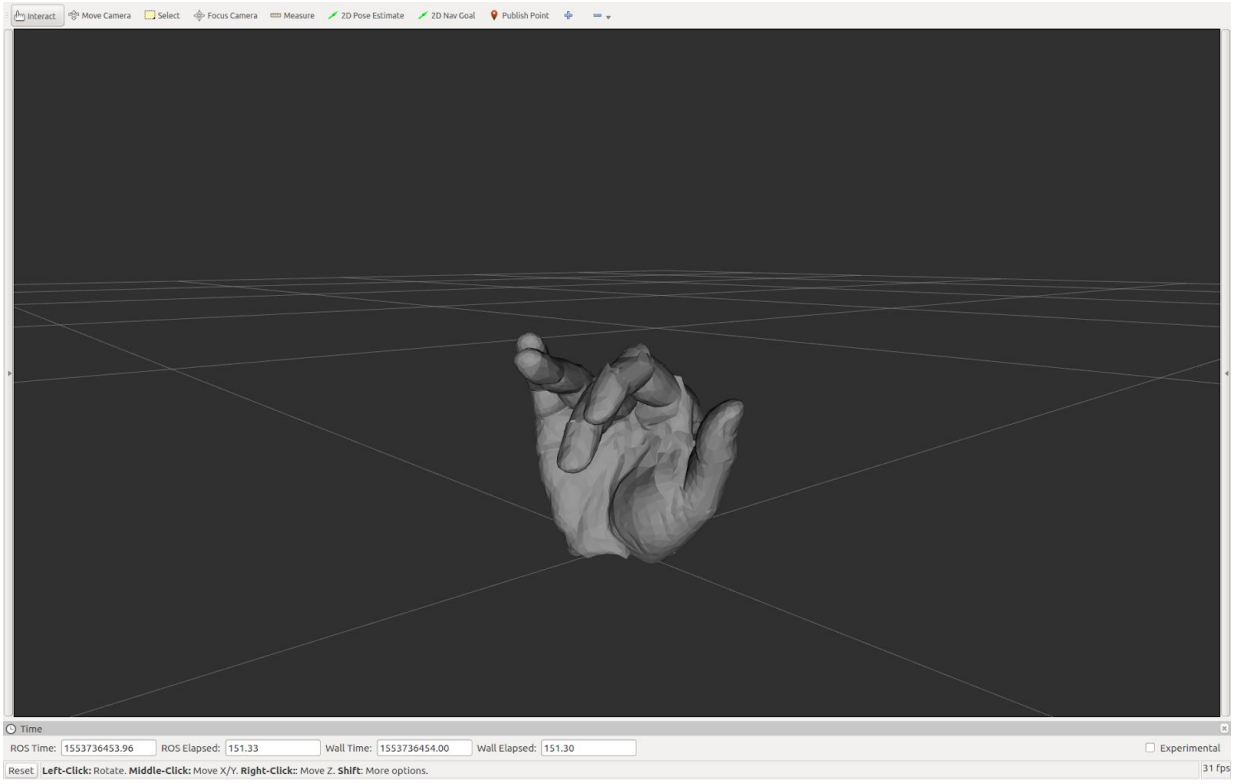
At each calibration step, raw data from the Cyberglove is collected for a unique hand pose to find the sensor readings that map to the kinematic model of the human hand. The lowest level Cyberglove library uses the calibration file, denoted by the extension *.cal*, to calibrate the had as follows:

```
<?xml version="1.0" ?>
<Cyberglove_calibration>
<Joint name="G_PinkieRingAb">
<calib raw_value="0.248031" calibrated_value="50.000000"/>
<calib raw_value="0.311024" calibrated_value="0.000000"/>
```

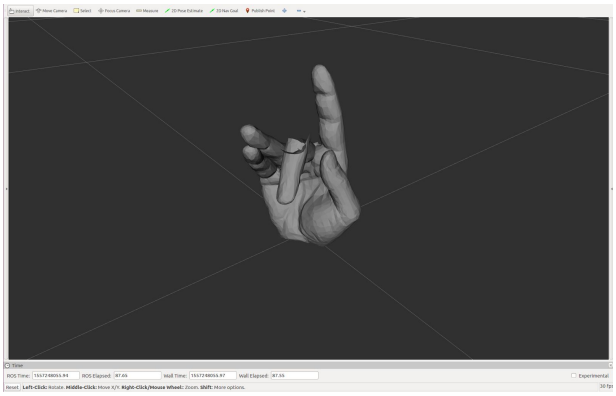
```
</Joint>
<Joint>...</Joint>
</Cyberglove_calibration>
```

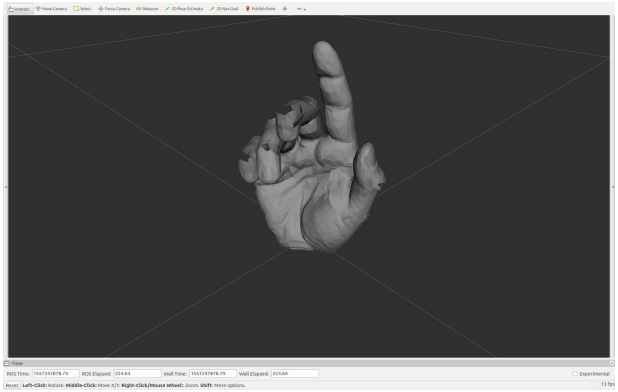
D. Visualizing the Master and Slave Hands

Both the Ctrl-Labs' armband and the Cyberglove can be visualized in RViz for the purposes of debugging teleoperation within simulation. A unified robot description format (URDF) of the human hand previously developed in the Columbia Robotics Lab was used to render the joint angles streamed in real-time by either piece of hardware. The master hand displayed within RViz is shown below:



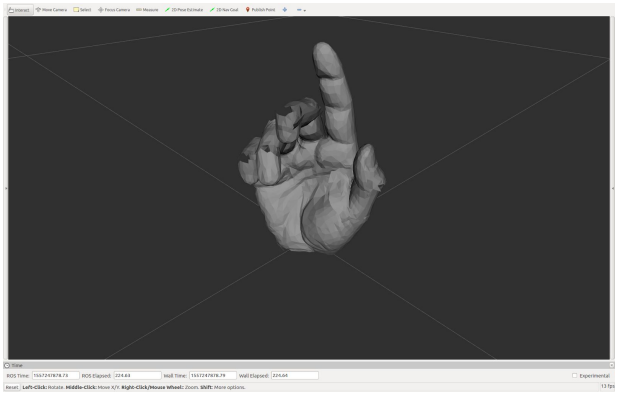
The diagram below drives home the importance of calibrating the Cyberglove. The figure to the left visualizes the hand without calibration while the figure on the right represents post-calibration:

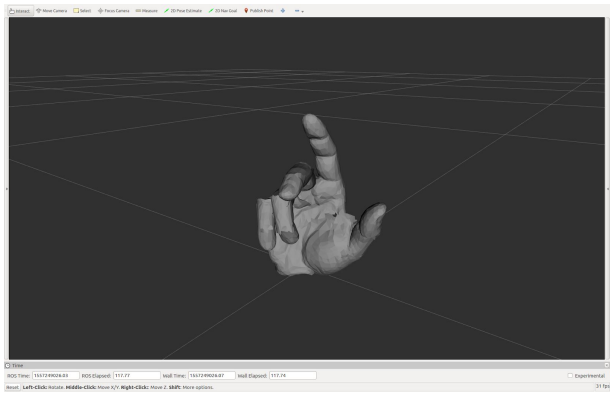




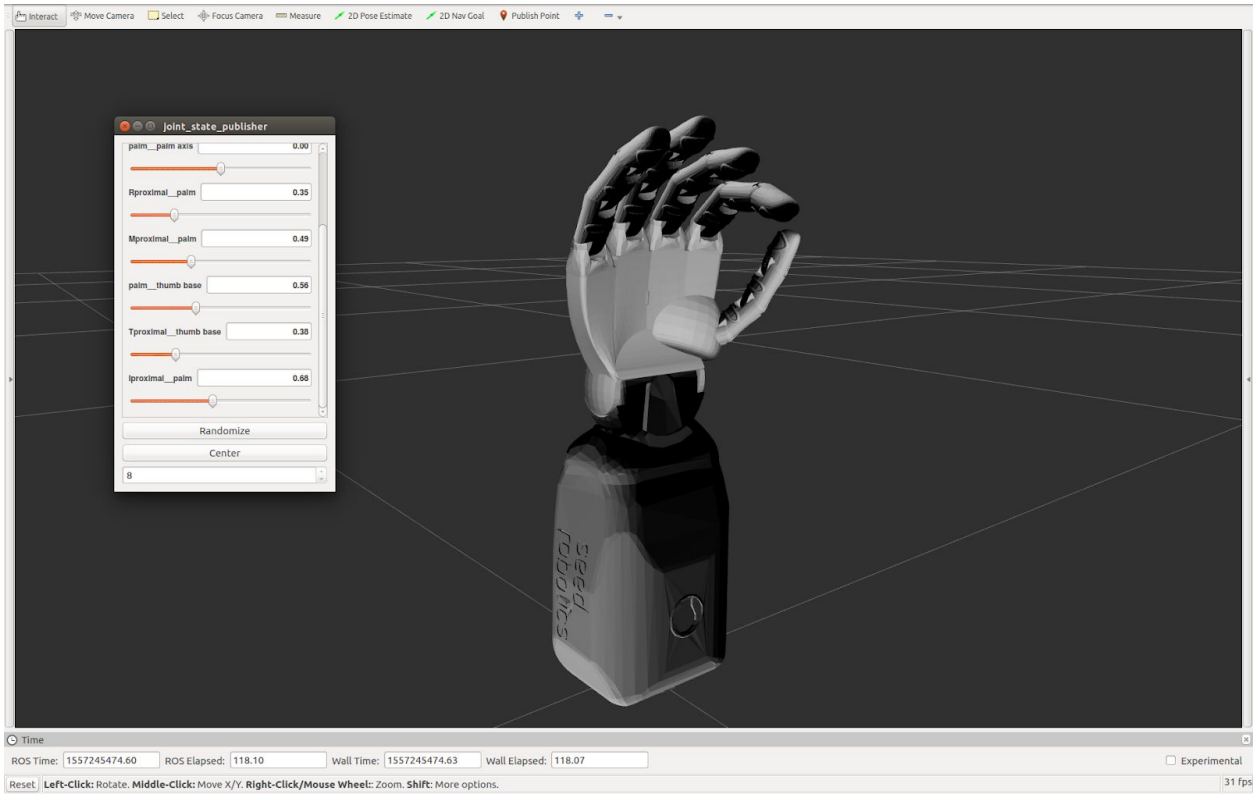
Abhi Gupta and Max Xu, Columbia University
COMS 6731 Humanoid Robotics, Professor Peter Allen

Juxtaposing the Cyberglove (left) and the Beast (right) demonstrate the performance of Ctrl-Labs' predictive hand state model against ground truth:



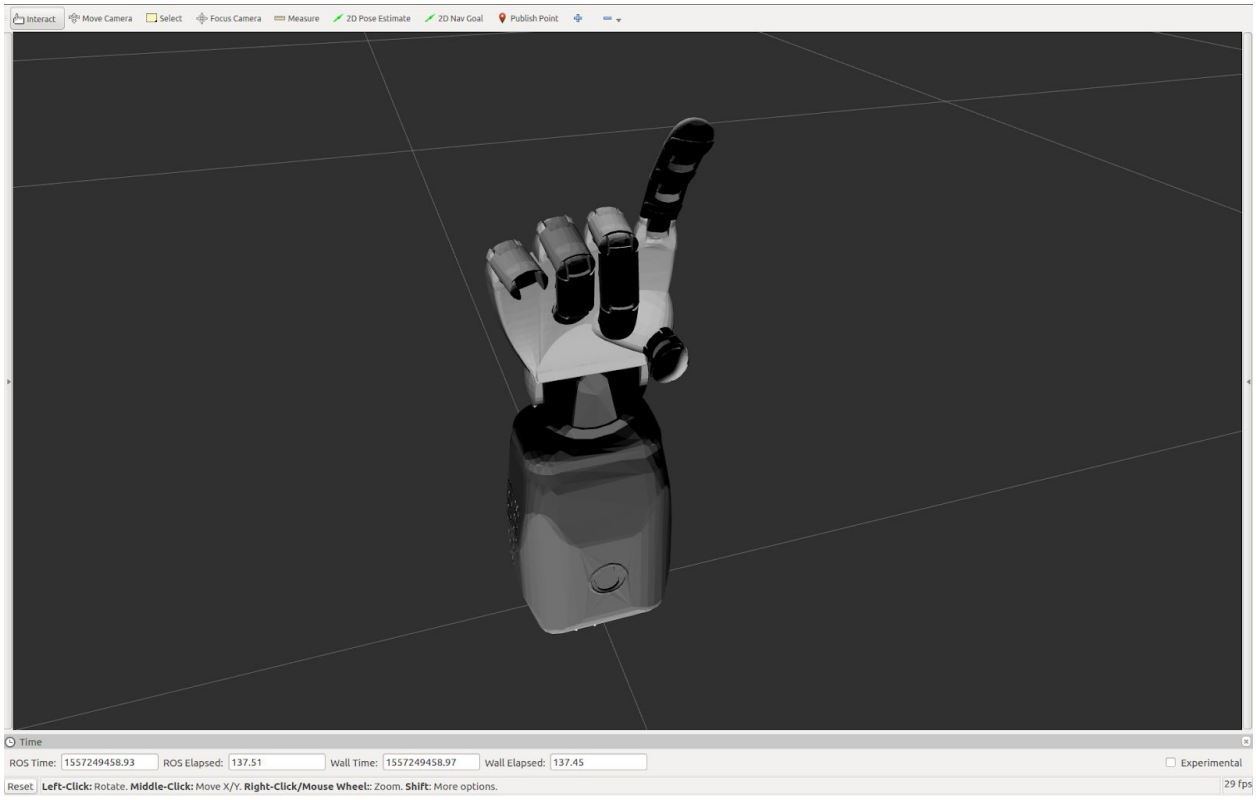


Visualizing the slave hand similarly requires the URDF of the Seed hand. Although not publically available, the manufacturer provided exclusive access to the computer-aided design (CAD) meshes needed to build a model of the hand within simulation. The Seed hand when loaded in RViz looks as follows:



The sliders depicted in the diagram above are provided by a vanilla ROS node called *robot_state_publisher,* which makes use of the URDF and the joint angles from the ROS topic */joint_states* to publish the forward kinematics of

the model. The Seed hand URDF was originally fully actuated, thereby permitting control of 18 degrees of freedom when in practice not every joint can be independently controlled. To account for this difference in experiment and simulation, several observations were made about how the Seed hand actuates in practice. As described in *Hardware Specifications,* child joints tend to move about ninety percent of their parent joint (i.e. the fully-actuated finger joint) while grandchild joints tend to move about thirty percent of their parent joint (i.e. the child joint). Mimicking these ratios within the URDF of the slave hand, the Seed hand can now be controlled in simulation just as it would be controlled in reality. The slave hand in simulation is depicted below after maximally flexing the index finger during teleoperation:

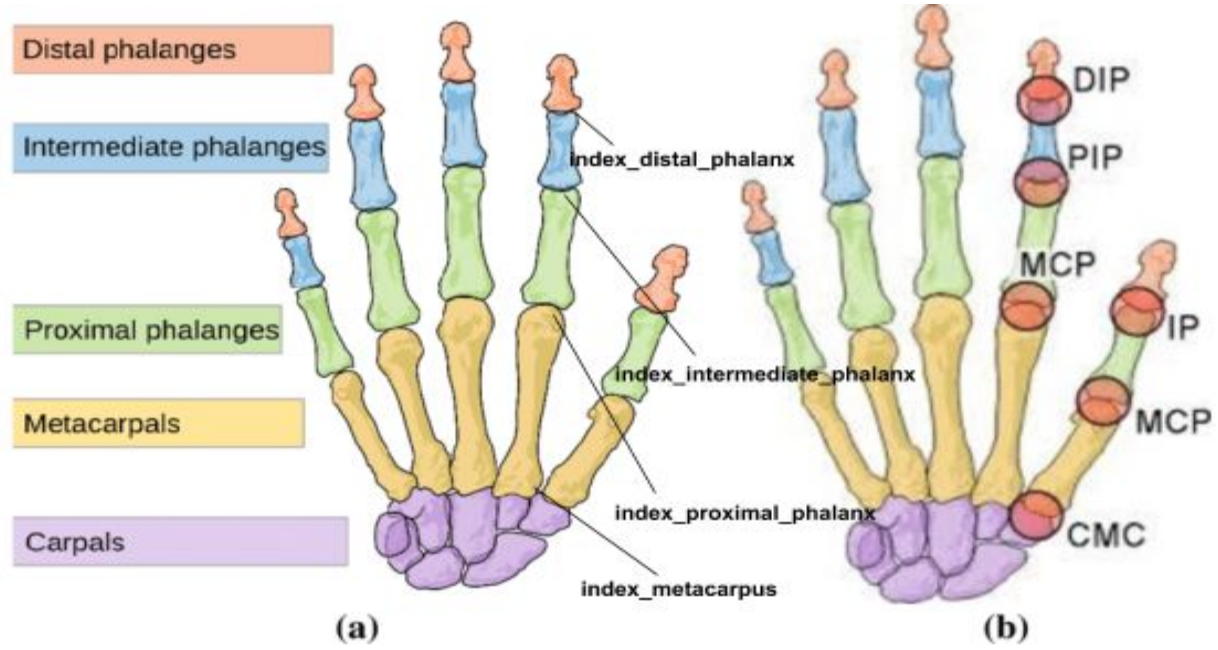


E. Mapping between Master and Slave Hand Pose Spaces

There are several ways in which the master hand can be mapped onto the slave hand. The most appropriate method must consider the hardware specifications of the slave hand as described earlier. In particular, finding a one-to-one mapping between every joint of the human hand and the Seed hand would only be possible if it was fully-actuated.

As explained in *Related Work*, Cassie et al's hand teleoperation method suffers from the inability to capture individual joint movements partly because it only considers three dimensions: the size, spread and curl of the fingers. However, it performs exceptionally well for robotic hands that do not share similar physical characteristics with the human hand. This suggests that the strategy to optimally map between the two hand pose spaces should capitalize on as opposed to neglect the anthropomorphism of the Seed hand. The following diagram illustrates the terminology of human finger bones for its relevance in interpreting the map between the master and slave hands:

Abhi Gupta and Max Xu, Columbia University
COMS 6731 Humanoid Robotics, Professor Peter Allen



When attempting to simulate underactuation of the Seed hand, the joint couplings discovered empirically in *Visualizing the Master and Slave Hands* proved to be a decent estimate of how the hand actuates in practice. For every fully-actuated joint, the child and grandchild joints were actuated by a proportion of the fully-actuated joint. In more well-defined terms:

$$\mathcal{J} = \{\text{fully actuated joints}\}, |\mathcal{J}| = 8$$
$$\mathcal{P} = \{\text{fully actuated proximal flexion joints}\} \subset \mathcal{J}, |\mathcal{P}| = 4$$
$$\mathcal{I} = \{\text{underactuated intermediate flexion joints}\}, |\mathcal{I}| = |\mathcal{P}|$$
$$\mathcal{D} = \{\text{underactuated distal flexion joints}\}, |\mathcal{D}| = |\mathcal{P}|$$
$$\mathcal{S} = \{\text{pinky},\ \text{ring},\ \text{middle},\ \text{index},\ \text{thumb}\}$$

The slave hand may only be controlled by a vector of joint angles from $\mathcal{P}$. Hence:

$$p^t = \{p_x,\ \forall x \in S\} = \{p_{pinky}, p_{ring}, p_{middle}, p_{index}, p_{thumb}\}$$
$$p_{pinky} \notin \mathcal{P},\ p_{pinky} = \alpha p_{ring},\ \alpha = 0.9$$
$$i^t \in \mathcal{I},\ i^t = \{i_x,\ \forall x \in S\},\ i_x = \beta p_x,\ \beta = 0.9$$
$$d^t \in \mathcal{D},\ d^t = \{d_x,\ \forall x \in S\},\ d_x = \gamma i_x = \beta\gamma p_x,\ \gamma = 0.3$$

Each under-actuated flexion joint position is a fraction of the fully actuated proximal flexion joint. The entire hand state of the Seed hand can then be expressed as a function of the fully actuated joints by:

$$f(p^t) = y^t,\ y^t \in \mathcal{P} \cup \mathcal{I} \cup \mathcal{D}, f : \mathbb{R}^{|\mathcal{P}|} \mapsto \mathbb{R}^{|\mathcal{P}\cup\mathcal{I}\cup\mathcal{D}|}$$
$$f(p^t) = p^t A,\ A \in \mathbb{R}^{|\mathcal{P}|\times|\mathcal{P}\cup\mathcal{I}\cup\mathcal{D}}$$

Abhi Gupta and Max Xu, Columbia University
COMS 6731 Humanoid Robotics, Professor Peter Allen

$$A = \begin{bmatrix} 1 & \beta & \beta\gamma & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \beta & \beta\gamma & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & \beta & \beta\gamma \end{bmatrix}$$

Given that the entire hand state can be approximated by exclusively the fully actuated joints of the slave, the inverse method will approximate the control parameters of the fully actuated joints of the slave from the hand state of the master as follows:

$$\text{Let } g \cong f^{-1}, f^{-1} : \mathbb{R}^{|\mathcal{P} \cup \mathcal{I} \cup \mathcal{D}|} \mapsto \mathbb{R}^{|\mathcal{P}|}$$
$$y^t A^\top = p^t A A^\top$$
$$y^t A^\intercal (A A^\intercal)^{-1} = p^t$$
$$f^{-1}(y^t) = y^t A^\intercal (A A^\intercal)^{-1}$$

The inverse function maps the joints such that a fully actuated joint on the slave hand is a weighted sum of all of the joints along the respective finger on the master:

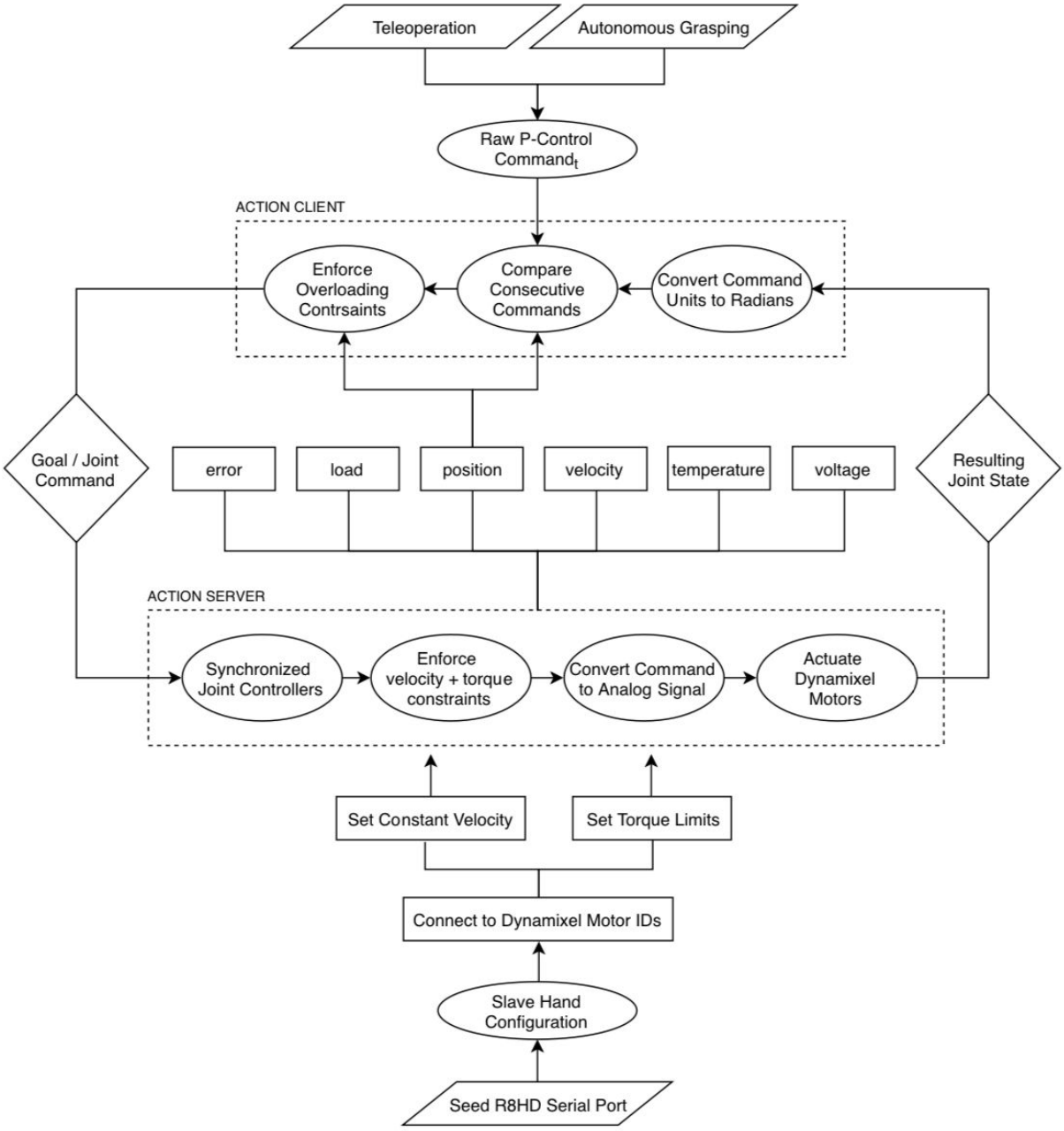
$$\text{Let } j \subset y^t,\ j \in \{\mathcal{P}, \mathcal{I}, \mathcal{D}\}$$
$$p_x^t = \sigma \sum_{j \in \{\mathcal{P}, \mathcal{I}, \mathcal{D}\}} j_x^t,\ \forall x \in \mathcal{S}$$
$$\forall\ k^t, l^t \in \mathcal{J} \wedge\ k^t, l^t \notin \mathcal{P} \cup \mathcal{I} \cup \mathcal{D},\ k^t \in p^t,\ l^t \in y^t, k^t = l^t$$
$$g(y^t) = p^t = \sigma y^t A^\top, \because A^\intercal (A A^\intercal)^{-1} = \sigma A^\intercal, \sigma = 0.53$$

σ is chosen as the eigenvalue of $(AA^\intercal)^{-1}$, representing a scaling of the inverse mapping to compensate for error in the coefficients beta and gamma found empirically earlier.

F. Developing with the Seed Hand

The Seed Robotics hand utilizes the open-source ROS dynamixel drivers available at http://wiki.ros.org/dynamixel_driver. Given the variety of dynamixel libraries available for ROS, it is important to exclusively work with the aforementioned drivers to avoid experiencing unexpected behavior at runtime.

It was necessary to develop a hand controller for the Seed hand to operate the hand both in experiment and in simulation with a physics engine like PyBullet or Gazebo. In particular, an action server was written to group the motors of the hand into a single, controllable unit by an action client. The speed of each actuator is set to 100.0 by default but may be reconfigured by invoking the ROS service /set_speed on the joint of interest. The diagram below describes the controller on a high-level:



The status light on the forearm of the Seed hand indicates three different states of the hand. The hand is functioning as expected when the status light steadily emits a green, blue, and red pattern. A bold, flashing red pulse instead suggests that the hand has either *overheated* or *overloaded.* Console messages from the action server will specify which of the two triggered the error.

Overheating occurs when the hand has been in use for prolonged periods of time or if the ventilation system built into the hand has been blocked. On the other hand, overloading is a safety measure put in place by the manufacturer, Seed Robotics, to prevent damage to any actuator caused from exerting torque beyond its capacity. In other words, the hand will inevitably shut down if any of the fingers attempt to move to a position that is unreachable due to a colliding body. Such a safety measure in a nutshell impedes the hand's ability to robustly grasp objects since it cannot continue to exert the force needed to ensure a firm and functional grasp.

Resolving the challenge of overloading depends on the purpose of the action client being used to control the hand. An action client with the intent of sending grasp commands from an autonomous grasp planning algorithm does not need to account for the added complexity of sending real-time position commands as in teleoperation. Therefore, the solution to prevent the Seed hand from overloading during teleoperation builds upon the algorithm that corrects overloading for direct grasp commands.

Each fully-actuated joint has a domain in which its position can lie--a minimum angle of zero radians and a maximum angle of 2pi radians. A simple grasp command will typically instruct the hand to move to a new joint state (i.e. the grasp) characterized by a set of eight joint angles. Knowing nothing about the environment, it can be concluded that to successfully execute such a joint command, the upper bound of each joint should be the joint angle command or the maximum valid angle (2pi radians), whichever is smaller. In essence, the position of any finger should never exceed and not necessarily even reach the provided joint command due to unanticipated collision with an object. Following this line of thinking, we instruct the hand to move to the upper limit of its joint domains as computed above. Object collision can then be inferred when the joint angle of any of the fingers marginally increases from the last timestep to the current timestep. In these cases, the maximum value in the domain of the specific joint becomes the angle that it was able to reach at the current timestep since anything greater will have overloaded the hand (i.e. [0,theta_sub_t]). With a set of updated joint domains, the hand is again instructed to move to the upper end of its domains, resetting the domains back to their initial values (i.e. [0,2pi]) whenever collision cannot be inferred.

Taking further precautionary steps to avoid damaging the hand, it was decided that the domains of the fingers should be changed to the best reachable value at the current timestep less a pullback constant when in collision (i.e. [0,theta_sub_t-K]). This ensures that the hand loosens its grip around an object just enough without dropping it before tightening its grip again, fundamentally reducing the load on the actuators.

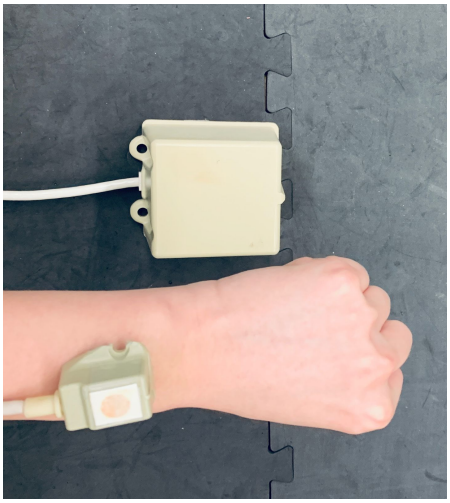
Within the context of teleoperation, the solution to overloading does not only depend on the state of the slave hand (i.e. Seed hand) but also on the state of the master hand (i.e. the teleoperator). Consider the case where the teleoperator remains stationary: the action client will command the Seed hand to move to the same position between consecutive timesteps. Earlier collision was inferred between the hand and object by recognizing a marginal change in joint position between consecutive timesteps. Sending identical or similar enough joint commands between consecutive timesteps is thus no different than detecting collision with an object, inevitably instructing the slave hand to cease motion even when it is not colliding with anything. Therefore, it is important for the action client to skip any consecutive joint commands that are not substantially different from each other.

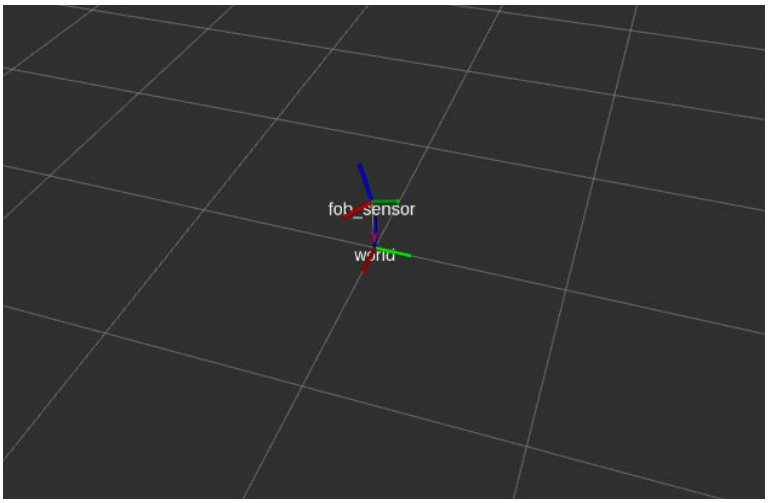
Overcoming overloading during teleoperation is a three-step procedure: observe the state of the master hand and preempt any consecutive actions that are too similar (1), observe the state of the slave hand to detect object collision (2), and downsample the joint commands from the teleoperator to a frequency appropriate for the Seed hand while minimizing latency. Synchronizing sampling frequencies between master and slave is necessary because the Seed hand can respond to actions at a rate no higher than 2 Hz whereas it is possible to stream joint commands

from the user at either 50 Hz with the Ctrl-Labs' armband or 120 Hz with the Cyberglove. However, observations of the Seed hand are also made at either 50 Hz or 120 Hz as opposed to 2 Hz since evaluating deltas in position twice per second would be far too slow to detect collision prior to triggering overloading.

G. Tracking Wrist Pose with Ascenion's Flock of Birds (FOB) Sensor

The Flock of Birds or FOB sensor tracks the pose of the subject's wrist within 1.2m of the magnetic base in each principle direction. The expected positional error is about 1.8mm. It was exceptionally difficult to interface with the device because the technology has now been deprecated for a while. After countless failed attempts, it was concluded that the FOB operates over serial port at a baud rate of 115,200. Any messages sent to or received from the FOB also must be encoded in or decoded from binary format, thereby requiring a parser to correctly interpret the pose information of interest. A transform between the world origin in RViz and the pose streamed from the FOB is published in order to visualize the raw tracking of the small unit placed on the wrist relative to the magnetic base. An example of real-time tracking with the FOB is shown below:





The accuracy and convenience of the FOB makes it an ideal candidate for recording the ground truth pose of 3D objects and calibrating a Kinect camera, for example.

H. Controlling the UR5 Arm in Gazebo

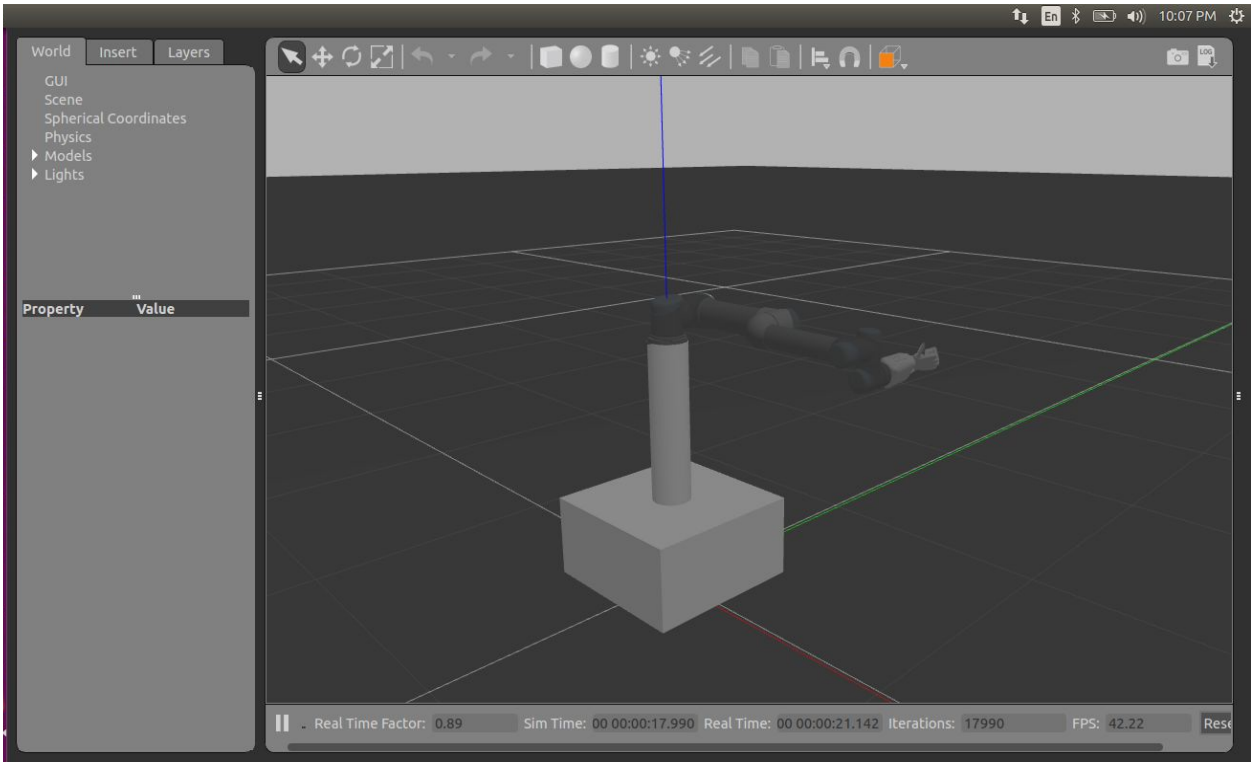
Gazebo is a commonly used simulation tool for robotics research. Simulation is always a safe way to do experiments before working with expensive and fragile hardware. A robot model of the UR5 arm can be spawned directly from the URDF. The ros_control package available at http://wiki.ros.org/ros_control is used to set up the simulation controllers for each joint of arm. Each controller is a joint trajectory controller, therefore allowing us to use the MoveIt! API to perform motion planning both in simulation and in experiment.

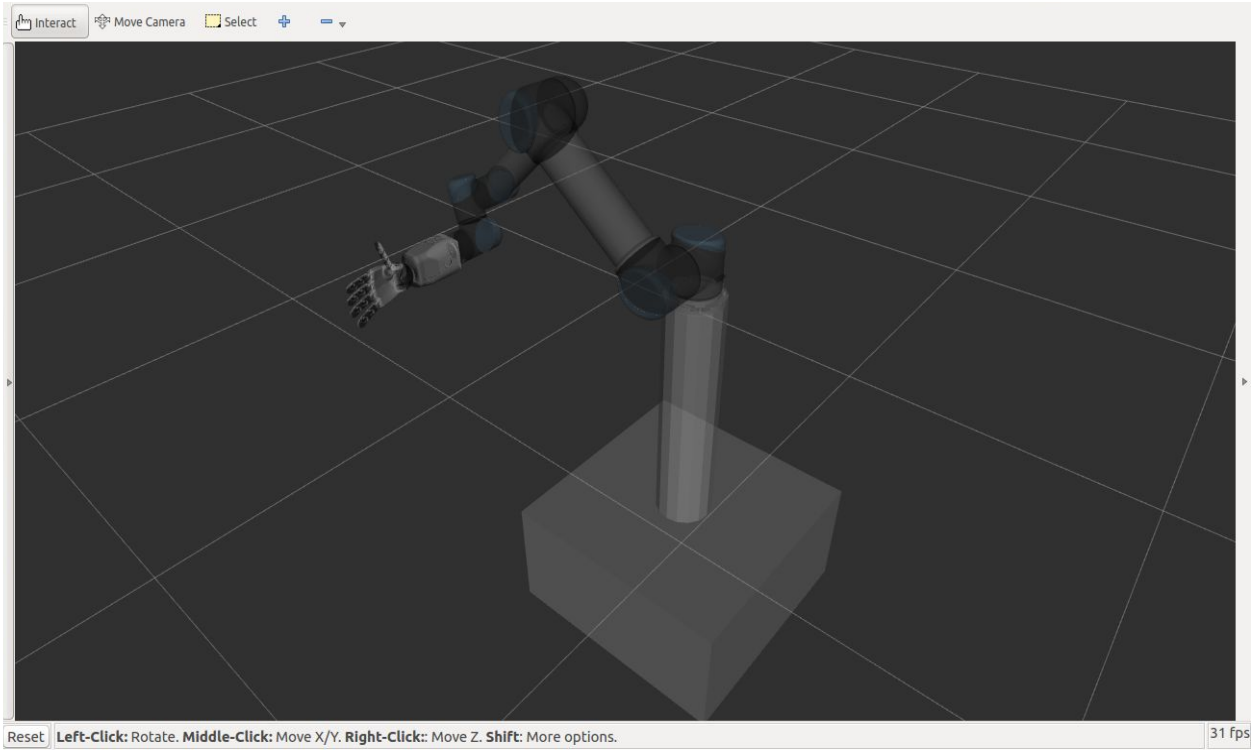
I. Simulating the Seed Hand and UR5 Arm Together

In order to simulate the Seed hand and UR5 arm together, the URDF files were unified. A fixed virtual joint called base_joint was connected to the parent link, the UR5 end-effector, and the child link, the base of the Seed hand. Another base model was introduced into the URDF to prevent the robot hand-arm configuration from

colliding with itself during motion planning. The robot when rendered in Gazebo and RViz, respectively, looks as follows:

Abhi Gupta and Max Xu, Columbia University
COMS 6731 Humanoid Robotics, Professor Peter Allen





J. Mapping between Wrist Pose and End-Effector Pose

The FOB sensor streams in real-time the pose of the small unit mounted on the wrist relative to a magnetic base at a frequency of 100 Hz: ${}^{su}T_{mb}$. However, teloperating the arm requires the wrist pose relative to the teleoperator's shoulder. The required transformation matrix from the human's wrist to the shoulder is denoted by ${}^{hw}T_{hs}$. Therefore, ${}^{hw}T_{hs} = {}^{su}T_{hs}$ because the small unit is attached to the subject's wrist. To find ${}^{su}T_{hs}$, a sensor calibration method is proposed.

The subject is asked to stretch and position his arm in the three directions: down, side and forward. Performing these poses in the order mentioned, it is possible to record three wrist position points within the magnetic base frame: p_1, p_2, p_3. Because the average length of the human arm is 0.635m, the three wrist positions should approximate a_1, a_2, a_3, where $a_1 = [0, 0, 0.635, 1]^\top$, $a_2 = [-0.635, 0, 0, 1]^\top$, $a_3 = [0, -0.635, 0, 1]^\top$. Finding ${}^{hs}T_{mb}$, the transformation matrix from the human shoulder frame to the magnetic base, then becomes an optimization problem with the loss function:

$$Loss = ||{}^{hs}T_{mb}p_1 - a_1|| + ||{}^{hs}T_{mb}p_2 - a_2|| + ||{}^{hs}T_{mb}p_3 - a_3||.$$

${}^{hs}T_{mb}$ is a function of (t_x, t_y, t_z, r, p, y), where t_x, t_y, t_z and r, p, y are the translation and euler angle from human shoulder to magnetic base, respectively. Using the machine learning python package Scipy and the Sequential Least Squares Programming (SLSQP) minimization method, the following was solved:

$$\underset{t_x,t_y,t_z,r,p,y \in R}{\operatorname{argmin}} ||{}^{hs}T_{mb}p_1 - a_1|| + ||{}^{hs}T_{mb}p_2 - a_2|| + ||{}^{hs}T_{mb}p_3 - a_3||$$

After finding ${}^{hs}T_{mb}$, with kinetics chain the wrist pose relative to the shoulder can also be derived: ${}^{su}T_{hs} = {}^{su}T_{mb} \cdot ({}^{hs}T_{mb})^{-1}$. The arm stretching forward during calibration of the last human pose provides the initial pose of the wrist. Let the initial pose transformation matrix from the forward frame to the magnetic base be ${}^{f}T_{mb}$ and from the forward frame to the shoulder be ${}^{f}T_{hs}$. The rotation in the transform from the forward frame to the shoulder is not meaningful, so we preserve the translation but eliminate the rotation from the initial pose, denoting it as ${}^{ff}T_{hs}$. The transformation between the post-processed initial pose and raw initial pose is then ${}^{ff}T_{f} = {}^{ff}T_{hs} \cdot ({}^{f}T_{hs})^{-1}$. Therefore, the x, y, z axis of the forward frame in the post-processed forward frame can be described as:

$$x = {}^{ff}T_f \cdot [1, 0, 0, 1]^\top, y = {}^{ff}T_f \cdot [0, 1, 0, 1]^\top, z = {}^{ff}T_f \cdot [0, 0, 1, 1]^\top$$

At every timestep, the FOB publishes the new or updated pose of the small unit placed on the subject's wrist relative to the magnetic base: ${}^{np}T_{mb}$. The transformation matrix from an initial pose to a new pose is then ${}^{f}T_{np} = {}^{f}T_{mb} \cdot ({}^{np}T_{mb})^{-1}$. The rotation in euler angles $r_i, i \in x, y, z$ can be found from ${}^{f}T_{np}$ by computing a rotation matrix R_i for each r_i rotating around the i axis. The translation t_f can similarly be found from ${}^{f}T_{np}$. The translation in the forward frame then becomes $t_f f = {}^{ff}T_f \cdot t_f$, which when converted to a transformation

matrix is denoted by T_t. Thus, ${}^{f}T_{np}$ in the frame of the initial pose is ${}^{ff}T_{np'} = T_t \cdot \prod_{i=x,y,z} R_i$. The resulting transform from the base of the UR5 to the end-effector given a new wrist pose from the FOB sensor, where T_{offset} is the offset translation matrix between the teleoperator and the UR5, becomes:

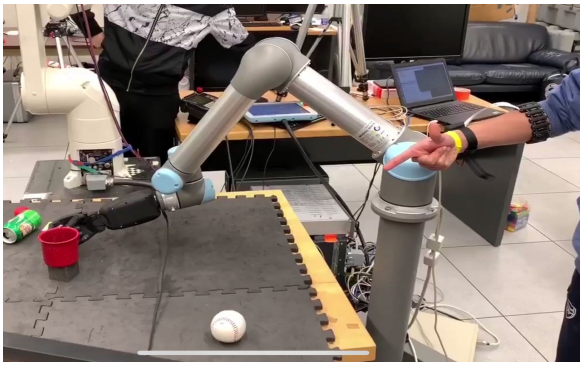
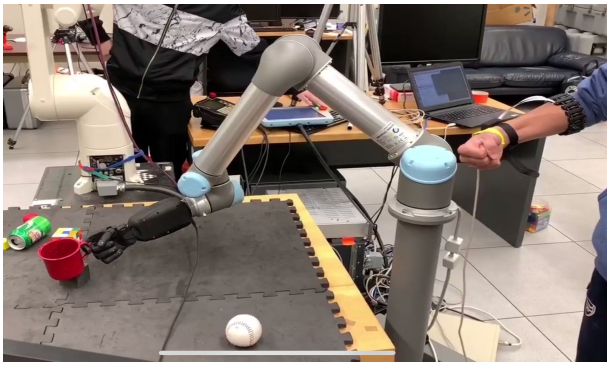
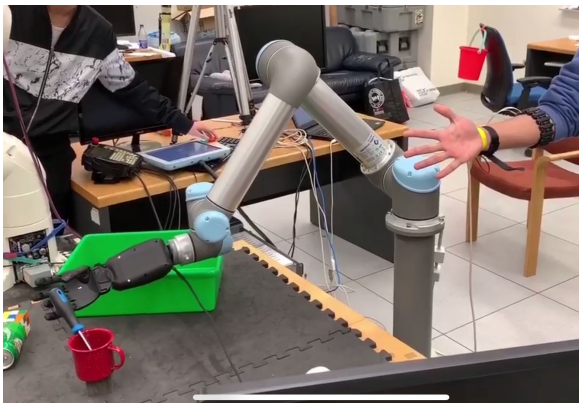
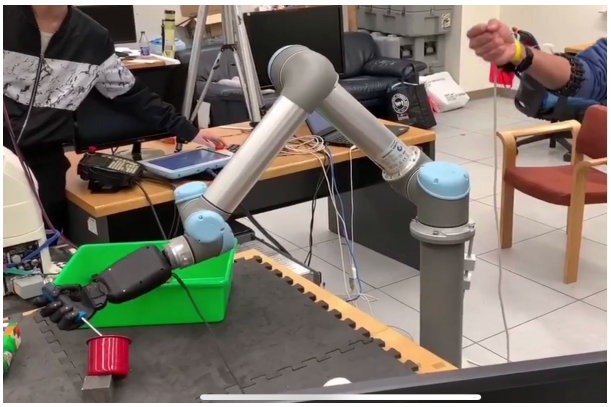
$$ {}^{b}T_{ee} = T_{offset} \cdot {}^{hs}T_{np'} = T_{offset} \cdot ({}^{ff}T_{hs})^{-1} \cdot {}^{ff}T_{np'}, $$

K. Overcoming Latency in Arm Teleoperation

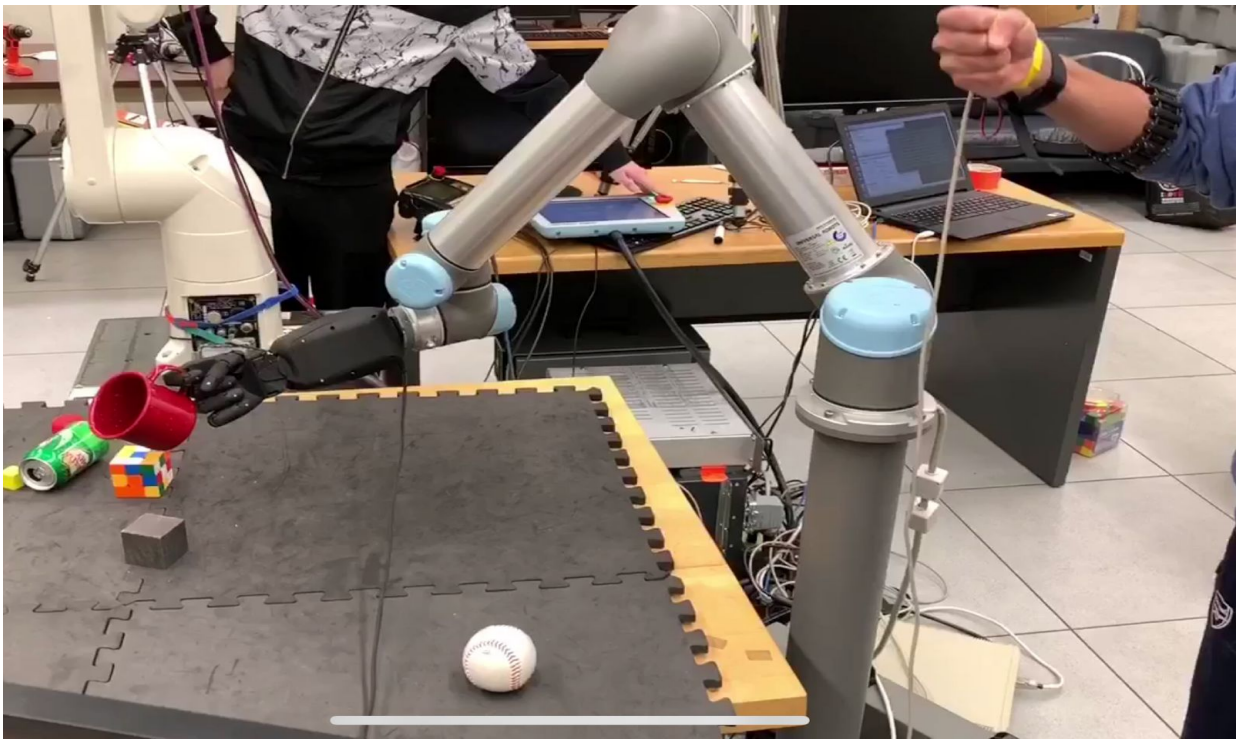
Teleoperation of the UR5 currently remains slow because we do not perform continuous motion planning, When a new target pose is received, it must wait for the current plan to finish. Moreover, moveit plans such that the start and end pose have zero speed for all joints. The average latency is around 2 seconds as a result. Descart is a cartesian path planning method for welding robots that currently supports dynamic re-planning. However, it only supports off-online planning and therefore cannot work in real-time as required by teleoperation. Motion planning with attention to velocity space is the next direction of overcoming latency in controlling the arm.

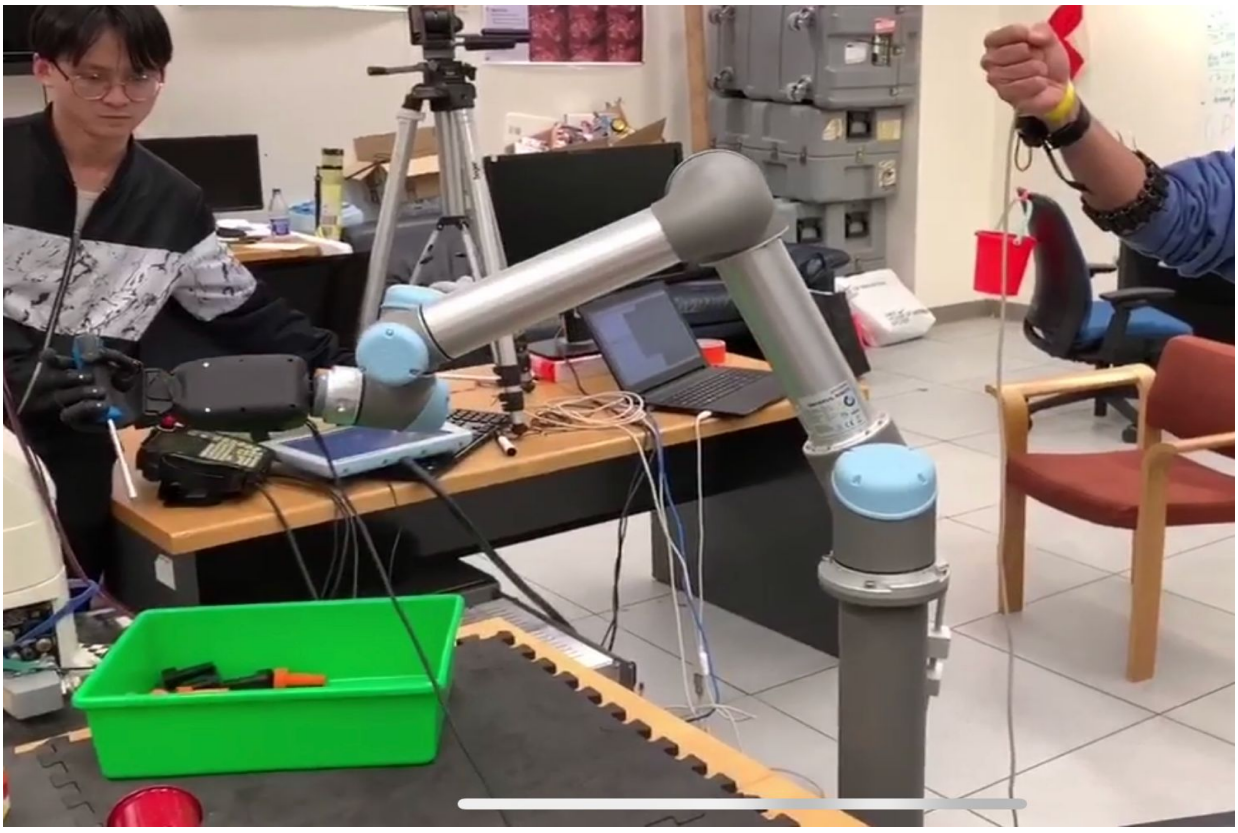
RESULTS (https://docs.google.com/document/d/1-6dEGYDNTfIVMsFmDtvFNfNLl4PSfF7fpGY88oBYViQ/edit)

Human teleoperation of the Seed R8HD hand and the UR5 arm was successfully completed. The method described in this project enables a user to pick and place a variety of objects including but not limited to a Rubik's cube, squash ball, soda can, water bottle, coffee mug, and screwdriver. Access to streams of data like master-slave poses, master-slave joint angles, EMG readings from the teleoperator's arm, and object pose open up the possibility for developing a grasping dataset from teleoperation alone. Examples of performing pick and place tasks via teleoperation are demonstrated below:

Abhi Gupta and Max Xu, Columbia University
COMS 6731 Humanoid Robotics, Professor Peter Allen





NEXT STEPS

A. Contributing to the Ninapro Dataset

The utility of benchmark databases has been demonstrated repeatedly in the fields of computer vision and image processing. Attempting to fill the gap between human performance and artificial results within robotics has also motivated the creation of manipulation and object datasets. In particular, the Ninapro (Non-Invasive Adaptive Hand Prosthetics) database provides a benchmark electromyography data source to test machine learning algorithms for hand prosthesis control. Ninapro currently includes six datasets collected from 67 intact and 11 amputated subjects, all of whom perform a set of exercises lasting 5 seconds, followed by 3 seconds of rest, as outlined by their protocol. The exercises capture fundamental movements of the fingers, isometric and isotonic hand and wrist configurations, and a variety of precision grasps.

In addition, Pizzolato et al. introduce six different acquisition setups, each acquisition method generating the aforementioned Ninapro dataset, all recording the information that is also exposed by the Ctrl-Labs armband. One acquisition setup of interest includes the use of two Thalmic Myo armband worn on the same arm coupled with an uncalibrated data glove to record hand state. Upon performing a series of classification algorithms on the Ninapro dataset, it is concluded that the dual Myo setup is comparable to devices affordable exclusively for the scientific research community.

A great way to introduce the Ctrl-Labs device as a viable research EMG instrument would be to contribute to the Ninapro dataset with the architecture developed in this project, employing the same evaluation metrics as those used in the original construction of the dataset to argue the performance of the hardware within the context of teleoperation. The results of this potential direction could be used as a benchmark to support the effectiveness of grasp planning and control strategies developed for teleoperating the Seed UR5 robot.

REFERENCES

Terlemez, O., Ulbrich, S., Mandery, C., Do, M., Vahrenkamp, N., and Asfour, T. (2014). "Master motor map framework and toolkit for capturing, representing, and reproducing human motion on humanoid robots," in *2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (IEEE) (Madrid), 894–901.

M. Ciocarlie, C. Goldfeder and P. Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping," *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, 2007, pp. 3270-3275.

Santello, M., Bianchi, M., Gabiccini, M., Ricciardi, E., Salvietti, G., Prattichizzo, D., et al. (2016). Hand synergies: integration of robotics and neuroscience for understanding the control of biological and artificial hands. *Phys. Life Rev.* 17, 1–23. doi: 10.1016/j.plrev.2016.02.001

Dipietro, L., Sabatini, A. M., and Dario, P. (2008). A survey of glove-based systems and their applications. *IEEE Trans. Syst. Man Cybern. C* 38, 461–482. doi: 10.1109/TSMCC.2008.923862

Fani, S., Ciotti, S., Catalano, M. G., Grioli, G., Tognetti, A., Valenza, G., et al. (2018). Simplifying telerobotics: wearability and teleimpedance improves human-robot interactions in teleoperation. *IEEE Robot. Automat. Magazine* 25, 77–88. doi: 10.1109/MRA.2017.2741579

Bianchi, M., Salaris, P., and Bicchi, A. (2013). Synergy-based hand pose sensing: Reconstruction enhancement. *Int. J. Robot. Res.* 32, 396–406. doi: 10.1177/0278364912474078

Ong, E.-J., and Bowden, R. (2004). "A boosted classifier tree for hand shape detection," in *Proceedings of Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004* (IEEE) (Seoul), 889–894.

Cheng, J., Xie, C., Bian, W., and Tao, D. (2012). Feature fusion for 3d hand gesture recognition by learning a shared hidden space. *Patt. Recogn. Lett.* 33, 476–484. doi: 10.1016/j.patrec.2010.12.009

IVRE - An Immersive Virtual Robotics Environment. https://cirl.lcsr.jhu.edu/research/human-machine-collaborative- systems/ivre/.

Beh, J., Han, D. K., Durasiwami, R., and Ko, H. (2014). Hidden markov model on a unit hypersphere space for gesture trajectory recognition. *Patt. Recogn. Lett.* 36, 144–153. doi: 10.1016/j.patrec.2013.10.007

Meeker, Cassie & Rasmussen, Thomas & Ciocarlie, Matei. (2018). Intuitive Hand Teleoperation by Novice Operators Using a Continuous Teleoperation Subspace. 1-7. 10.1109/ICRA.2018.8460506.

Pizzolato S, Tagliapietra L, Cognolato M, Reggiani M, Müller H, et al. (2017) Comparison of six electromyography acquisition setups on hand movement classification tasks. PLOS ONE 12(10): e0186132. https://doi.org/10.1371/journal.pone.0186132

U. Scarcia, R. Meattini and C. Melchiorri, "Mapping human hand fingertips motion to an anthropomorphic robotic hand," *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Macau, 2017, pp. 774-779.

Bergamasco, Massimo, et al. "An arm exoskeleton system for teleoperation and virtual environments applications." *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE, 1994.

Tadakuma, Riichiro, et al. "Development of multi-DOF master-slave arm with bilateral impedance control for telexistence." *Proceedings of 14th International Symposium on Measurement & Control in Robotics (ISMCR'04)*. 2004.

Kim, Dongmok, et al. "Excavator tele-operation system using a human arm." *Automation in construction* 18.2 (2009): 173-182.

Chitta, Sachin, Ioan Sucan, and Steve Cousins. "Moveit![ros topics]." *IEEE Robotics & Automation Magazine* 19.1 (2012): 18-19.

De Maeyer, Jeroen, Bart Moyaers, and Eric Demeester. "Cartesian path planning for arc welding robots: Evaluation of the descartes algorithm." *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2017.