Poser: Active Manipulation of a Passive Humanoid

Lexi Foland
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA
lkfoland@mit.edu

Jocelyn Zhao
Department of EECS
Massachusetts Institute of Technology
Cambridge, MA
jjz300@mit.edu

Jieruei Chang
Department of EECS
Massachusetts Institute of Technology
Cambridge, MA
jieruei@mit.edu

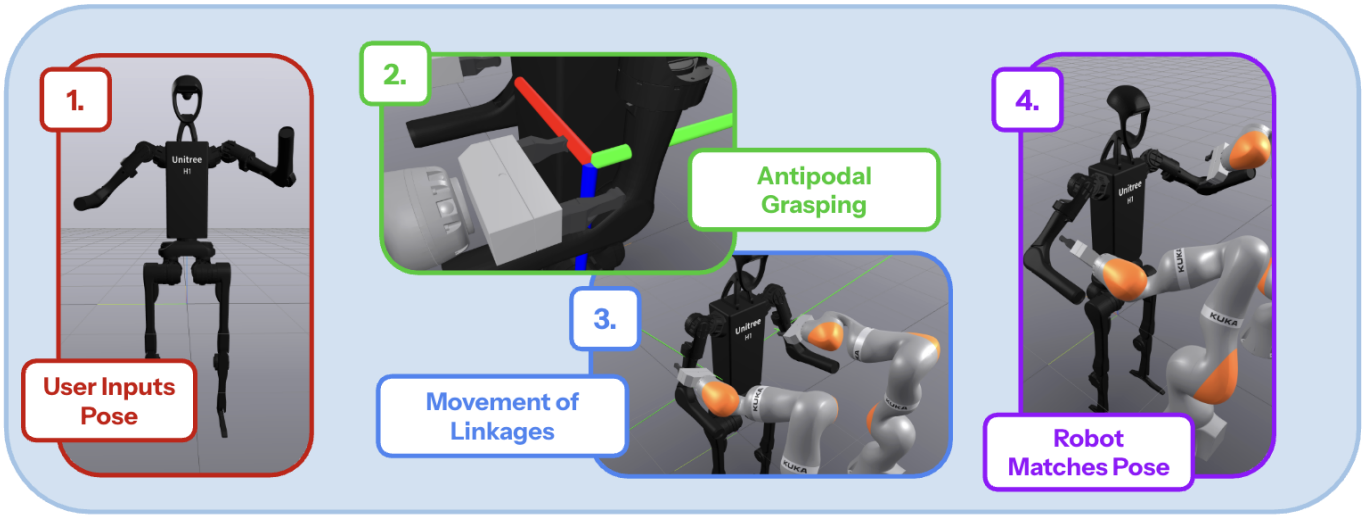

Fig. 1: *Overview of the manipulation pipeline.* (1) The user selects a pose for the humanoid via a GUI. (2) Antipodal points are sampled for grasping the target linkage. (3) IIWA grasps and moves the humanoid joints via contact forces. (4) IIWA manipulates the humanoid to match the user's input pose.

***Abstract*—Manipulation tasks that specifically require the robots to contact one another are underexplored; however, these tasks have significant potential to be applied in industry. We propose an instantiation of this problem that involves three robots: two robotic arms serve as "designer" arms that physically manipulate a third, passive "mannequin" humanoid. We use compliant control to enable the contact-rich interactions required to pose the humanoid in desired configurations. We leverage strategies such as antipodal grasp computation, updating trajectory planning, and control with gravity compensation and velocity damping to improve task success rates. Finally, we present a series of nontrivial target configurations that our system can mimic via posing of the passive humanoid with the robotic arm and an analysis of the system's ability to mimic arbitrary configurations. We conclude with contributions about effective implementation of a mannequin-posing system as well as strategies to improve robustness.**

***Index Terms*—motion planning, jointed systems, compliant control**

I. Introduction

Robotics problems explicitly encouraging robot-on-robot contact are often unexplored; in many cases, robots are programmed to avoid contact with one another. In the event of a high-speed collision, these machines can easily damage both themselves and the contacted objects. However, this leaves gaps in current research, specifically about robot-on-robot contact with varying types of control. This topic presents interesting challenges and potential in the domains of multi-robot cooperation and contact-rich manipulation, motivating our exploration of this subject. In this work, we seek to explore how effectively a robotic arm with 2-finger gripper can exert contact forces to manipulate a passive humanoid to a goal pose.

We instantiate this problem by considering a setup of three robots. First, we consider a Unitree H1 humanoid with a compliant controller, allowing it to react to external impulses in a manner representing a mannequin. Compliant control, which considers contact forces during computation of applied torques, is a commonly cited tactic in contact-rich tasks. Next, we add two additional KUKA LBR iiwa 14 robotic arms that act as "designers," exerting contact forces upon the mannequin humanoid to pose linkages in desired positions. Given an input pose for the mannequin, the arms complete a sequence of precise movements to manipulate the linkages of the humanoid to match the desired configuration.

Although control plays a large role in the success of this system, we consider the application of other foundational concepts in robotics to enhance the robustness of our setup. We utilize relevant functionalities of the software Drake to maintain state information of all three machines throughout

task execution. Specifically, we heavily rely on ground-truth poses provided by the kinematic trees of the humanoid and robotic arms; these quantities are used to track both desired and measured homogeneous transforms for linkages of the humanoid. Additionally, we seek to move the humanoid via contact forces from the robotic arms; we suggest that an efficient method for doing so is by grasping a mannequin linkage with the end effector of a robotic arm, then moving said end effector to a target position. We justify this method in III-A and in IV. We utilize antipodal grasp computation to tightly grip linkages during transportation and the Jacobian pseudoinverse to map end-effector velocities to joint velocities.

Through this project, we present a full posing system that receives a humanoid configuration as input and produces a full visualization of the linkage-movement process. We hope that our research provides valuable contributions to inquiries regarding robot-on-robot contact and complaint control. We believe the strongest applications of this work are in industrial settings, in which delicate machine parts are often handled by robotic arms during assembly. Although our specific problem instantiation differs from traditional industrial tasks, this work may give insight into considerations necessary for manipulation of actively controlled objects, such as the humanoid's linkages.

We hypothesize that a system consisting of antipodal grasp computation, trajectory planning, and compliant control will create a manipulation system that accurately poses the humanoid to resemble some goal pose within a reasonable amount of time. Through this project, we present a full posing system that receives a humanoid configuration as input and produces a full visualization of the linkage-movement process. The results from experimentation show that our system can successfully pose humanoid linkages within a few centimeters and slight rotation offsets for both benchmark and randomized poses.

II. Related Work

In terms of mannequin-like tasks, [1] offers an implementation of a soft robotics system that models human bodies. The primary contributions of this work revolve around deformable bodies and 3D modeling, providing insight into how to make humanoids gain some of the qualities of a model or mannequin. On the other hand, [2] proposes a humanoid robot that resembles a mannequin in terms of movement and expression. The authors propose a control system that allows for smooth movement of the humanoid, making the robot less stiff and more successful in interactions with real humans. These works concentrate on creating a realistic humanoid mannequin, but do not explore interactions with another robot's contact forces and manipulation into some target pose. While these works present thoughtful contributions at the intersection of mannequins and robots, their goals differ from ours. The central research question of our work focuses more so on the precise contact, planning, and controls required to manipulate a humanoid robot as if it were a mannequin. We use the conclusions in these works as bases for our research and then propose a further extension of the humanoid-mannequin problem.

Additionally, we consider prior works in which contact between robots is central to task completion; the most common applications of this research are the automotive and aerospace industries. [3] presents a general overview of common work provided by robotic arms in automotive assembly. Robotic arms must carefully and precisely execute planned trajectories to perform effective task execution while not damaging any of the handled parts. This behavior presents similarities to our project; in the mannequin problem, our robotic arms must also adhere exactly to trajectories to avoid damage to the humanoid and to successfully match the desired mannequin pose. Furthermore, the aerospace industry offers similar opportunities for robotic manipulation; [4] describes such work by robotic systems in quality control and tedious manufacturing jobs.

Like the problem in this paper, robots must execute precise, well-planned motions in order to effectively exert contact forces on delicate robot parts. Through our instantiation of the mannequin problem, we hope to further contribute to knowledge in automotive and aerospace assembly, especially considering the differences between our work and traditional usage of robotic arms in necessary tasks. While assembled parts are often portions of motors or frames in these fields, we seek to manipulate a fully built humanoid with a control system receptive to large external forces. We believe this work may provide insight into new techniques in robot and vehicle assembly.

III. System Architecture

The finalized posing system utilizes the following workflow: first, a user is presented with a browser interface for posing the humanoid in a desired configuration. Next, this configuration is recorded and used to calculate the sequence of motions required by the robotic arms to move the humanoid into these poses. For effective movement, the system calculates antipodal grasps to show the arms where to grip the humanoid's linkages. The initial location of each linkage can be gathered either through proprioceptive data or through a perception system that performs pose estimation. The arms execute a trajectory of end-effector commands based on the desired humanoid pose; this is accomplished via translation of end-effector velocities to joint velocities with the Jacobian pseudoinverse. Lastly, the arms finish execution of the trajectory, and the humanoid matches the desired input pose, thus functioning as a "mannequin." In this section, we dive into further detail about the key components of our system.

A. User Input

Before any action occurs, our system allows users to specify an input pose for the humanoid. First, the script starts a Meshcat browser page. Sliders in the page's side window allow for users to toggle the joint values of the humanoid. Due to workspace constraints of the robotic arms, functionality is only supported for joints in the humanoid's left and right arms. To finish the process, the user presses a "Submit Pose"

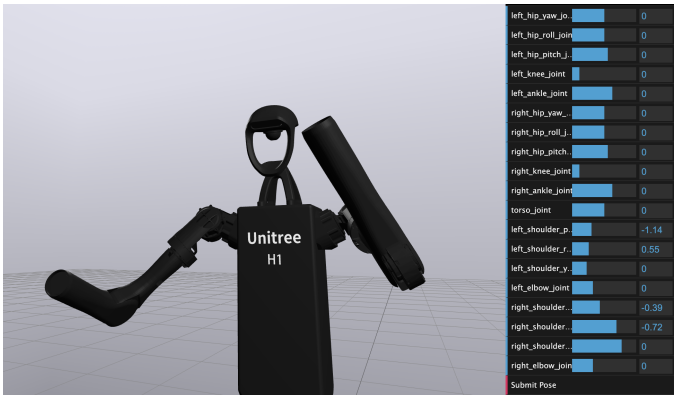

Fig. 2: The GUI for setting a desired pose for the humanoid's arm linkages. The user interface is shown in a Meshcat browser.

button; from there, the system records the target locations of the humanoid's shoulder and elbow linkages, stored as homogeneous transforms in the world frame.

From this point, the two robotic arms will grasp the linkages - shoulder first, then elbow - and move them to the target poses. Notably, the humanoid's parts are incredibly light and thus can be easily moved to target locations. Consequently, if the robotic arms move their end-effectors to the desired pose of each linkage (with some offset to account for grasp location), the linkages of the humanoid actually follow quite well. An analysis of this ability is presented in detail in IV. Computation of target poses is completed via references to the kinematic tree of the humanoid and robotic arms and via transform composition. 2 provides a visualization of the user interface for pose input.

B. Perception

The perception system allows the humanoid to start in an unknown and arbitrary pose. A system of five cameras produce depth images (shown in 3) that are fused to generate a complete point cloud of the arm geometry. The point cloud is used to localize the joints of the humanoid; this information is then passed to the antipodal grasp computation and trajectory optimization systems. Because of the articulated, nonrigid nature of the humanoid arm, standard point cloud registration algorithms such as classical ICP cannot be directly applied. Instead, we solve this using a two-stage RANSAC algorithm that fits cylindrical primitives to the upper arm and forearm. The algorithm consists of (1) point cloud preprocessing, (2) a three-point cylinder hypothesis construction, (3) an inlier-based scoring function incorporating both geometric and normal constraints, and two RANSAC stages corresponding to the (4) upper arm and (5) forearm.

1) Preprocessing: Let the input point cloud be $\mathcal{P}_0 \subset \mathbb{R}^3$. We first crop the workspace with an axis-aligned bounding box to remove robot-base structure and background clutter, producing a reduced set $\mathcal{P}_1$. To ensure uniform sampling density, we apply voxel-grid downsampling, yielding $\mathcal{P}$. For each point $x_i \in \mathcal{P}$ we estimate a surface normal n_i using a KD-tree hybrid search.

2) Cylinder Hypothesis from Three Points: Given three non-collinear points $p_1, p_2, p_3 \in \mathcal{P}$, we compute the circumcenter c of the triangle defined by these points. Let

$$\mathbf{a} = (p_2 - p_1), \qquad \mathbf{b} = (p_3 - p_1), \qquad \mathbf{n} = \mathbf{a} \times \mathbf{b}. \tag{1}$$

The circumcenter is given by

$$c = p_1 + \frac{(\mathbf{n} \times \mathbf{a})\,\|\mathbf{b}\|^2 + (\mathbf{b} \times \mathbf{n})\,\|\mathbf{a}\|^2}{2\,\|\mathbf{n}\|^2}, \tag{2}$$

and the circumradius by

$$r = \|c - p_1\|. \tag{3}$$

We can extract the cylinder axis direction from the normalized triangle normal using

$$\hat{u} = \frac{\mathbf{n}}{\|\mathbf{n}\|}. \tag{4}$$

3) Inlier Scoring: For a point x_i we compute its projection onto the cylinder axis

$$t_i = \langle x_i - c, \hat{u} \rangle, \qquad \hat{x}_i = c + t_i \hat{u}. \tag{5}$$

The radial distance relative to the axis is then

$$\rho_i = \|x_i - \hat{x}_i\|, \tag{6}$$

so the geometric surface deviation is

$$d_i = |\rho_i - r|. \tag{7}$$

We also measure normal (angular) misalignment with

$$u_i = |\langle n_i, \hat{u} \rangle|. \tag{8}$$

A point is an inlier if $d_i < \tau_d$ and $u_i < \tau_\theta$ for thresholds τ_d (distance) and τ_θ (normal angle). When the parent joint location p_J is known (e.g., the shoulder), we additionally reject hypotheses whose cylindrical volume does not contain the joint. This is enforced by computing the joint's radial distance relative to the axis:

$$\rho_J = \|(p_J - c) - \langle p_J - c, \hat{u} \rangle \hat{u}\|\,, \tag{9}$$

and accepting only hypotheses with $\rho_J \leq r$.

4) RANSAC for Upper Arm Estimation: We repeatedly draw random triples $\{p_1, p_2, p_3\}$, construct the corresponding cylinder hypothesis, and count inliers according to the criteria above. The hypothesis with maximal inlier count is selected as the upper arm model. Because the cylinder axis is oriented but not directed, we resolve sign ambiguity by selecting the orientation $\hat{u}_{\text{ua}}$ that points away from the known shoulder joint and toward the majority of inlier points. Using the known upper-arm length L_{ua}, the estimated elbow position is

$$\hat{e} = p_{\text{shoulder}} + L_{\text{ua}}\,\hat{u}_{\text{ua}}. \tag{10}$$

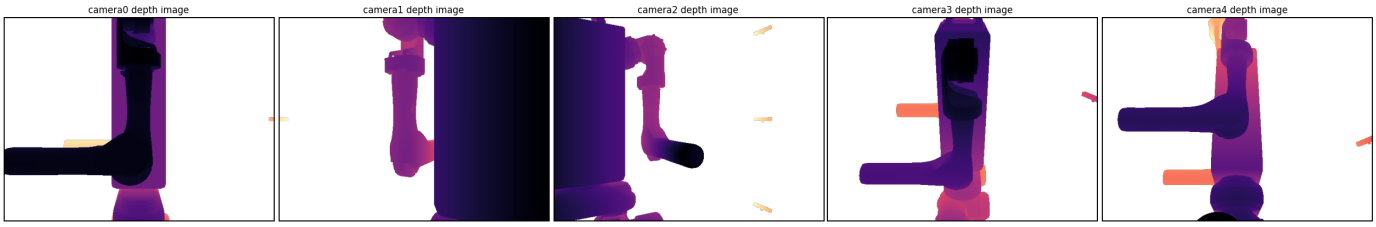

Fig. 3: Depth images from perception setup.

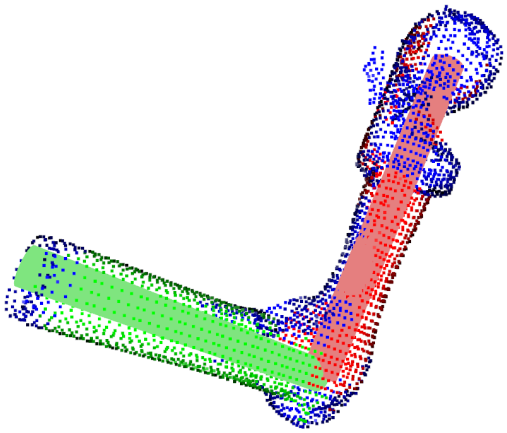

Fig. 4: Sample pose estimation result. The pose estimation algorithm fits two cylinders to the two humanoid arm linkages. Upper arm inliers detected by RANSAC are shown in red; forearm inliers are shown in green.

	Elbow Joint	Wrist Joint
Position (cm)	2.294	3.432

Fig. 5: Evaluation of error in joint positions reported by the pose estimation system. Lower numbers are better. The position error is Euclidean distance.

5) Forearm Estimation: All inliers belonging to the upper arm are removed from $\mathcal{P}$, and RANSAC is repeated on the remaining points. This produces a second cylindrical model $(c_{\text{fa}}, \hat{u}_{\text{fa}}, r_{\text{fa}})$. The sign of the axis direction is again chosen so that the axis points away from the elbow. With the known forearm length L_{fa}, the wrist position estimate is

$$\hat{w} = \hat{e} - L_{\text{fa}}\,\hat{u}_{\text{fa}}. \tag{11}$$

A visualization of the pose estimation result is shown in 4.

6) Perception Evaluation: We evaluate the accuracy of our perception system in 5. The positional error in the right shoulder and hand joints was determined using 10 randomized poses sampled from a Gaussian distribution of mean 0 and scale 0.25. The error of the shoulder joint is not calculated, since the perception system assumes it stays constant.

C. Grasp Computation

In order to generate physically plausible and effective grasps for the IIWA to manipulate the humanoid, we use antipodal grasp sampling. We choose antipodal grasps because they place the fingers of the end effector on opposite sides of the object with roughly opposing surface normals. While our implementation does not account for force closure, this method has empirically been shown to produce stable two-finger grasps.

The method takes as input a 3D mesh of the target link and the link's pose with respect to the world frame. This data is either given from the perception module or taken from ground-truth poses and the humanoid's STL file. We first sample the 3D mesh for 500 surface points. We chose 500 points as a tradeoff between computational cost and grasp selection: we could sample fewer points to reduce runtime but this may fail to produce feasible grasps for the curved humanoid linkages. Sampling more points can improve the collection of grasps to sample from but increase the scale of later computations in a linear manner. For reproducibility, we set a constant seed for the point sampling process.

For each sampled surface point, we cast a ray that originates at the point and travels in the direction opposite to the surface normal. We compute the first intersection of this ray with the mesh with trimesh's `ray.intersects_location()` function. If no intersection is found, the sample is discarded. Otherwise, the ray gives a second surface point on the opposite side of the object, along with its associated surface normal.

To enforce approximate antipodal contact, we apply a strict criteria based on the dot product of the two surface normals. Specifically, we require the dot product between surface normals on opposite sides to be < -0.95 (the dot product of two unit vectors is -1 when they point in entirely opposite directions). This threshold ensures that the normals are oppositely aligned, while still permitting moderate deviations. Candidate pairs that do not satisfy this condition are rejected. We further constrain candidate grasps based on the Euclidean distance between the two contact points. This distance is required to lie within a range determined by the mechanical limits of the gripper (<0.04 meters). We also require grasps with distance >0.005 meters because any smaller would indicate an error with ray casting computation. Of the qualifying grasps, we arbitrarily select one. A visualization of a chosen antipodal grasp can be seen in 6.

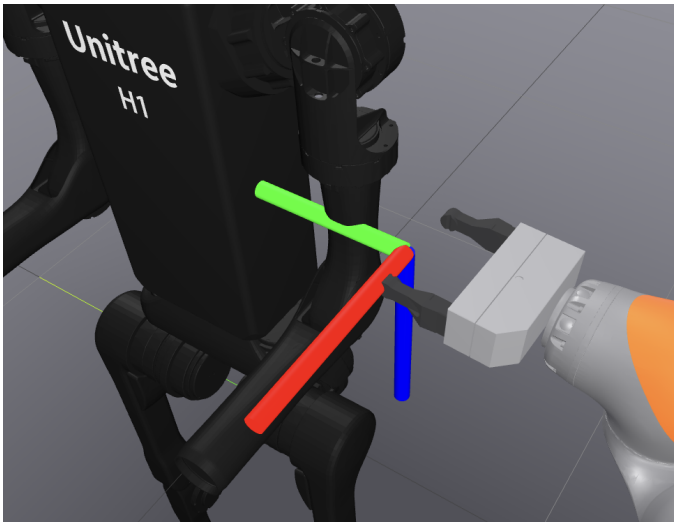

Fig. 6: An image of antipodal grasp, computed for the left shoulder linkage. Given a target joint, our system computes an optimal antipodal grasp for the associated humanoid linkage.

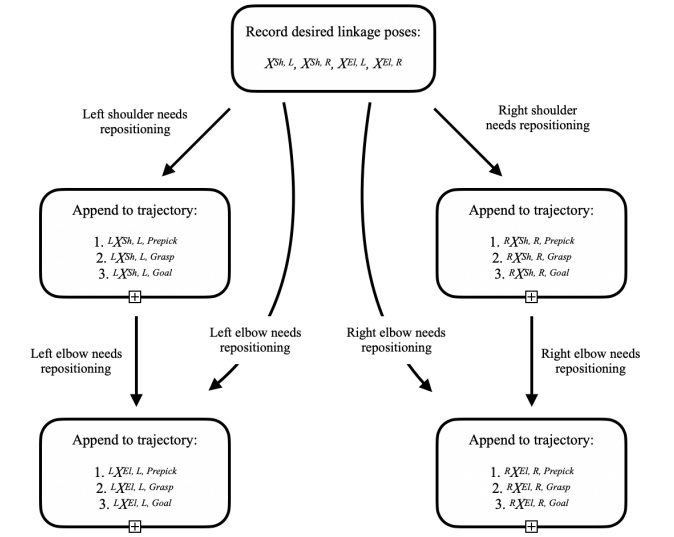

Fig. 7: A diagram showing the progressive creation of a trajectory for each arm to execute. The left column shows the trajectory created for the left arm, and the right column shows the trajectory created for the right arm. The abbreviation Sh means shoulder, El means elbow, L means left, and R means right. For clarification, ${}^{L}X$ means that pose is expressed in the base frame of the left robotic arm, and same for right.

D. Trajectory Creation

Our implementation records a sequence of keyframes to compose trajectories for each robotic arm. These keyframes are defined as 4 by 4 homogeneous transforms representing the desired poses of each end effector, expressed in the corresponding arm's base frame. Additionally, each keyframe is accompanied by a gripper command, denoting whether the fingers should be open or closed.

First, the system gathers the desired world-frame poses of each of the humanoid's arm linkages, given by user input. Next, we calculate the desired antipodal grasps for the humanoid's shoulder linkages, based on either proprioceptive data or pose estimation via perception; this output homogeneous transform (along with a pre-pick pose) makes up the next portion of the keyframe trajectory. The grippers close around the object, and we then move the end-effector to the desired input pose of the shoulder, with the offset of the grasp considered. Because the contact forces exerted by the grippers on the linkages are much stronger than any frictional or gravitational forces, the linkages track the motion of the end-effector quite accurately as they are transported; for validation of this claim, see IV. The final keyframes consist of a retreat sequence back to the original, home pose of the robotic arm. If an elbow link of the humanoid requires movement as well, we repeat the same sequence with that robot part in focus, instead of the shoulder.

At first, we composed this trajectory all in one shot, before any simulation occurred: compose left shoulder trajectory along with elbow trajectory, and same for the right arm. However, this approach introduces a key issue: any slight error in positioning of the shoulder affects the positioning of the elbow as well, since the elbow depends upon the shoulder in the humanoid's kinematic tree. Consequently, if the shoulder location is off after trajectory execution, the robotic arm will grasp the elbow in a non-optimal way - or fail to grasp the elbow altogether.

Instead of completing all trajectory calculations beforehand, we changed our approach to moving each shoulder as required, reassessing the world frame poses of each elbow linkage, and then computing the second half of each trajectory with the updated elbow locations in mind. For further clarification, a detailed diagram of the trajectory calculation can be seen in 7.

E. Control

The IIWA arm is controlled with the pseudoinverse controller. Given the desired velocity of the end-effector, we solve for the desired velocities of the joints with the pseudoinverse of the Jacobian. The pseudoinverse is required here because the IIWA has 7 degrees of freedom (DOF) while the task space has only 6 DOF. The Jacobian cannot be inverted directly and we can optimize for minimal position change.

To simplify the scope of the project, control is only implemented for joints of the humanoid that should actually be moving during task completion; all other joints are locked via MultibodyPlant's LockJoint method, and the pelvis of the humanoid is rigidly welded to the world frame. When designing the shoulder and elbow controller, we considered the specific requirements of the arm-mannequin interactions: the humanoid must be receptive to contact forces from the arms, compensate for the force of gravity, and remain still if not experiencing external forces. With these considerations

in mind, a control law was designed and is described in 12. The first term drives all joint velocities to zero, encouraging stillness, the second term compensates for the force of gravity, and the third feedforward term encourages the joints to move in tandem with the external forces applied by the robotic arms.

$$u = K_d \cdot \dot{q} - \tau_g(q, \dot{q}) + \tau_c(s) \tag{12}$$

IV. Evaluation

Ideally, a system that robustly addresses the mannequin-posing problem should receive arbitrary input commands and, after execution, report measured poses of the humanoid's linkages that match the input command as closely as possible. Consequently, evaluation of the system's success revolves around this goal. First, we consider the ability of our system to move the humanoid to each of four benchmark poses. Secondly, we consider our setup's ability to execute randomized poses. For both tests, we quantify pose error on a per-linkage and aggregated basis. This error is considered with respect to both position and orientation by considering the desired versus measured homogeneous transforms from the world frame to each rigid body. We quantify the position error of a rigid body R as the Euclidean distance between the desired and measured position of R's body frame in the world frame. Similarly, we calculate the orientation error as a quaternion distance metric, gathered from [5]. The metric, described in 13, gathers the distance between two different quaternions and expresses it on a scale of $[0, \sqrt{(2)}]$.

$$d(q_1, q_2) = \min(||q_1 - q_2||, ||q_1 + q_2||) \tag{13}$$

A. Benchmark Poses

We consider four benchmark poses (see 8) and the ability of our system to accurately rearrange the mannequin into such configurations. As the simulation is deterministic, we run our system once with each input pose and examine the error after task completion. The results are reflected in 9. We see certain poses are more difficult for certain linkages; for example, "Salute" has the highest position error for the right elbow and shoulder but the lowest position error for the left elbow and shoulder. Overall, we tend to see the highest position error in the elbow linkages, but orientation error varies on a pose-by-pose basis, with "On Hips" consistently showing the highest values.

Next, we examine the system's robustness to randomized poses. In its base configuration, each joint of the humanoid starts at value 0 radians. Consequently, we construct an alternative testing setup to user input, in which each desired joint angle of the humanoid's unlocked joints is sampled from a normal Gaussian distribution centered at 0, with an output scale of 0.25. One might argue that a uniform distribution across the valid range of joint values would make more sense, to which we provide the following counterarguments:

1) When multiple joints get assigned values far from 0, which is more likely with the uniform distribution, the humanoid often ends up in poses of self collision. For example, if a shoulder's pitch and roll joints both take extreme values, the corresponding arm will collide with the torso of the humanoid.
2) Additionally, joint assignments far from 0 can result in desired end-effector movements outside of the workspace of the designer robotic arms. We recognize this as a limitation of our project and consider fixes for this in section V.

The randomization procedure was repeated over 50 trials. The results of these experiments are presented in 10 as the average position and rotation error for each joint over all trials. Similarly to the benchmark evaluations, we see the highest position and orientation error in the elbow linkages, and the smallest error values in the shoulder linkages.

V. Discussion

Based on the results of both the benchmark and randomized experiments, we see that our system is able to transport linkages within a few centimeters of the desired position and close to the desired orientation. Notably, our system performed the best at pitch rotations of the humanoid's arms; the "Clap" pose, which has this quality, showed low error across both position and orientation. Contrarily, our system exhibited poorer behavior with poses requiring yaw rotations of the shoulders. This phenomenon can be seen through the results of the "On Hips" benchmark pose, which mostly required yaw rotations; while position error stayed low, orientation error reached its highest values across all benchmark poses. We attribute this result to an undesired result during transportation of the shoulder links; as the end effectors rotated in the yaw direction, the shoulder linkages visually appeared stationary, suggesting that the robotic arm was twisting its grippers around the humanoid's shoulder without actually inducing movement. This occurrence may come from one of two explanations (or a combination of both):

1) The antipodal grasps we attempted may not have truly achieved force closure.
2) The large shoulder linkage of the humanoid is technically its "yaw" linkage, as per its kinematic design; the robot similarly contains "roll" and "pitch" shoulder linkages, which are much smaller and thus difficult to grasp. Consequently, manipulation of the yaw linkage in the yaw direction may be hindered, as the attempted movement is direct twisting of the yaw actuator.

We observe slightly higher error values in the randomized pose trials. We attribute this to the randomization of input configurations providing slightly more difficult or unrealistic poses for the system to execute. In both sets of experiments, elbow linkages tend to have higher error than shoulder linkages. This result can be directly explained by the kinematic tree of the humanoid; the pose of the elbow depends upon the pose of the shoulder, as the shoulder yaw link is the parent body of the elbow link for both of the humanoid's arms. Due to the kinematic dependency, any error in the elbow link's pose compounds with error in the shoulder link's pose.

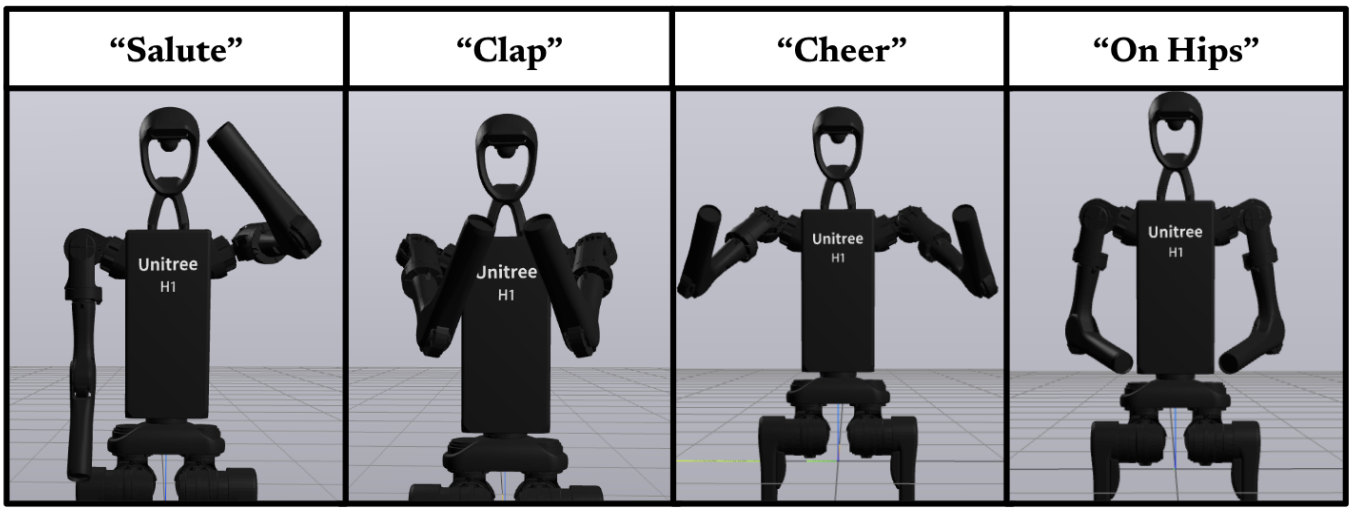

Fig. 8: Visualizations of the four selected primary evaluation poses.

Left Elbow

Error Type	Salute	Clap	Cheer	On Hips
Position (cm)	1.435	1.975	2.295	1.631
Rotation $[0, \sqrt{2}]$	0.118	0.028	0.033	0.18

Right Elbow

Error Type	Salute	Clap	Cheer	On Hips
Position (cm)	3.637	1.901	2.133	1.555
Rotation $[0, \sqrt{2}]$	0.086	0.027	0.029	0.159

Left Shoulder

Error Type	Salute	Clap	Cheer	On Hips
Position (cm)	0.695	0.854	0.979	0.843
Rotation $[0, \sqrt{2}]$	0.119	0.030	0.034	0.18

Right Shoulder

Error Type	Salute	Clap	Cheer	On Hips
Position (cm)	1.582	0.824	0.920	0.742
Rotation $[0, \sqrt{2}]$	0.055	0.028	0.031	0.159

Fig. 9: Evaluation of error in linkage positions and rotations after movement to benchmark poses. Lower numbers are better. The position error is Euclidean distance, and the rotation error is quaternion distance, described by 13, expressed on a scale of $[0, \sqrt{2}]$.

Elbows

Average Error	Left Elbow	Right Elbow
Position (cm)	2.878	4.194
Rotation $[0, \sqrt{2}]$	0.103	0.151

Shoulders

Average Error	Left Shoulder	Right Shoulder
Position (cm)	1.217	1.771
Rotation $[0, \sqrt{2}]$	0.073	0.113

Fig. 10: Evaluation of error in linkage positions and rotations after movement to poses sampled from a Gaussian distribution centered around the initial joint positions. Lower numbers are better. The position error is Euclidean distance, and the rotation error is quaternion distance, described by 13, expressed on a scale of $[0, \sqrt{2}]$.

We believe these results indicate that our problem addresses basic instantiations of the mannequin posing problem. We hope that our experimental results support our claim that our compliant control system, trajectory creation, and usage of antipodal grasping are adequate strategies for handling this problem. This being said, there are certain aspects of our system that could be improved for better robustness against adversarial pose inputs, such as the high yaw rotations mentioned in this section.

VI. Conclusion

In this work, we implement a robotic system that, given an input configuration, poses a "mannequin" humanoid using two "designer" robotic arms. We find combining seemingly simple compliant control, antipodal grasping, and trajectory planning

can effectively pose a humanoid. Through this research, we hope to make significant contributions to the explorations of deliberate physical contact between robots, a setting that is often avoided but critical for many industrial applications. Specifically, we propose that combining seemingly simple compliant control, antipodal grasping, and trajectory planning can effectively pose a humanoid and mimic input configurations with only slight error.

However, we acknowledge limitations of our current system. Firstly, we rely heavily on hand-designed trajectories, performing little collision-checking and trajectory optimization. This is largely due to the explicit encouragement of robot-on-robot contact and time constraints; we hope that future work may be able to incorporate both of these unimplemented strategies to create a more well-rounded approach to the posing problem. Additionally, our implementation of the passive humanoid's control is basic, and we suspect that a more rigorous approach would reduce some of the error observed in experimental evaluations. Furthermore, some of our strategies in this paper are not as tractable on real-world hardware, which is required for industrial applications. Overall we believe that the contributions of this paper provide a beneficial starting point for future work in explicit robot-on-robot contact, laying groundwork for ideas that could eventually be translated to hardware used in the automotive and aerospace industries.

ACKNOWLEDGMENT

We acknowledge the 6.4210 TAs and instructors for providing project advice, direction, and technical support. We also acknowledge 6.4210 communication instructors, who have helped us to strengthen the arguments and flow of this report.

CONTRIBUTIONS

Lexi F. built the trajectory calculation, GUI pose selector system, and the compliant humanoid controller. Jocelyn Z. built the antipodal grasp computation system. Jieruei C. built the perception system for humanoid pose estimation. All authors contributed to the writing of the manuscript.

REFERENCES

[1] Y. Tian, G. Fang, J. Petrulis, A. Weightman, and C. C. L. Wang, "Soft robotic mannequin: Design and algorithm for deformation control," *ArXiv*, May 2022. arXiv:2205.05166.

[2] M. Shidujaman, S. Zhang, R. Elder, and H. Mi, ""roboquin": A mannequin robot with natural humanoid movements," in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, (Nanjing), p. 1051–1056, IEEE, Aug. 2018.

[3] M. Bartoš, V. Bulej, M. Bohušík, J. Stanček, V. Ivanov, and P. Macek, "An overview of robot applications in automotive industry," *Transportation Research Procedia*, vol. 55, p. 837–844, 2021.

[4] S. Chowdhury, M. Siddique, R. K. Nayeem, M. Rahman, M. A. I. Talukder, and K. T. M. Hussain, "Study of robotics and automation in the aerospace industry," *ResearchGate*, 2022.

[5] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *J. Math. Imaging Vis.*, vol. 35, p. 155–164, Oct. 2009.