Validating Humanoid Teleoperated Trajectories via Adaptive Joint-Space Control

Massachusetts Institute of Technology

Final Project of 2.165 - Robotics

Mauricio Pereira

December 11, 2025

Contents

1 Introduction **2**

2 Background: SoftMimic and Whole-Body Control from Examples **2**

3 Experimental Platform: Unitree G1 and VisionProTeleop **3**

4 Problem Statement and Method Overview **3**
4.1 Teleoperated Demonstration Acquisition and Trajectory Generation 3
4.2 Trajectory Validation via Adaptive Joint-Space Control 4

5 Experimental Evaluation **5**
5.1 Reference Trajectories 5
5.2 Evaluation Metrics and Plots 6

6 Results and Discussion **6**

7 Conclusion and Future Work **8**

8 Appendix **9**

1 Introduction

Modern humanoid robots are increasingly expected to operate in contact-rich environments, collaborate safely with humans, and exhibit compliant whole-body behavior rather than purely stiff tracking of reference trajectories. Achieving this type of behavior requires both powerful control methods and data that captures realistic, dynamically consistent motions. Teleoperation is a natural way to obtain such data: a human operator directly drives the robot, and the resulting motions can be used both for immediate control and for training or evaluating learned policies.

This work is carried out as part of an undergraduate research project in the Improbable AI Lab at MIT, within the SoftMimic effort on compliant whole-body control from examples. In that context, the goal is to build and validate an upper-body teleoperation and control pipeline that can later serve as a source of demonstrations and infrastructure for SoftMimic-style learned controllers.

This project investigates a teleoperation and control pipeline for the Unitree G1 humanoid robot. The upper body of the G1 is teleoperated using an Apple Vision Pro headset, and the resulting trajectories are then replayed and validated using a separate adaptive joint-space controller in simulation. The focus is not only on making the robot follow the teleoperated motion, but also on assessing whether the trajectories are kinematically and dynamically feasible in a way that makes them suitable as training data for future learning-based controllers.

2 Background: SoftMimic and Whole-Body Control from Examples

`SoftMimic` is a recent project on compliant whole-body control from examples: the robot is trained to behave "softly" and safely while still tracking tasks, using demonstrations as the primary source of supervision. Instead of designing a full whole-body controller by hand, the idea is to learn a policy that maps states or observations to joint torques, conditioned on task information, such that the robot exhibits compliant, coordinated motion across the entire body.

In its future work discussion, the SoftMimic project highlights several missing pieces that are essential for scaling to more complex tasks:

- **Richer and dynamically plausible data.** Current datasets rely heavily on kinematic augmentation or simplified interaction models. There is a need for demonstrations that respect the full dynamics of the robot, including joint limits, torque constraints, and contact forces.
- **Whole-body compliant control from teleoperation or large motion datasets.** A long-term goal is to train a whole-body controller using either live teleoperation or large datasets of human-driven motion, rather than purely offline optimization or scripted behaviors.
- **Multi-contact and distributed wrenches.** Future systems should handle multiple simultaneous contacts and forces applied on different links, with wrenches distributed over the body instead of being concentrated at the wrists or end-effectors only.

The present project is positioned as a step in this direction. The Unitree G1 humanoid is used as the main platform, with an emphasis on generating and validating upper-body teleoperation trajectories that are suitable for later use in compliant whole-body control frameworks. Conceptually, the overall pipeline can be viewed as generating trajectories and state-action pairs of the form

$$\{x_t, q_d(t), \dot{q}_d(t), u_t\}_{t=0}^{T}, \tag{1}$$

where x_t denotes the robot state at time t, $q_d(t)$ and $\dot{q}_d(t)$ are teleoperated references, and u_t are the control inputs produced by the follower controller in simulation.

These sequences can be used as supervision for a policy

$$\tau_t = \pi_\theta(o_t), \tag{2}$$

where o_t is an observation (for example, joint positions, velocities, and task-space targets) and π_θ is a parameterized controller, such as a neural network in a SoftMimic-style architecture. Under this view, the teleoperation and adaptive tracking stack serves two roles: it validates that the teleoperated trajectories are kinematically and dynamically plausible, and it produces structured data that can later be used to train or evaluate compliant whole-body controllers.

If upper-body teleoperation combined with adaptive tracking behaves well, a natural next step is to extend the teleoperation to the lower body and to multi-contact scenarios, including ground and environmental contacts. This would enable live whole-body teleoperation that directly addresses the future-work directions identified by SoftMimic: richer, dynamically consistent demonstrations with forces and wrenches distributed over the entire body.

3 Experimental Platform: Unitree G1 and VisionProTeleop

The experimental platform combines a Unitree G1 humanoid robot with an augmented-reality teleoperation interface based on the Apple Vision Pro. Improbable AI has developed a tool, referred to here as `VisionProTeleop` , which streams the human operator's hand (and optionally head) motion from the headset and maps it to robot commands in real time.

The Vision Pro provides 6D poses for the left and right hands. Each hand pose is represented as a homogeneous transform

$$T_{\text{hand}}(t) \in SE(3), \tag{3}$$

encoding the position and orientation of the operator's wrist at time t. These poses are transmitted to the control stack, where they are transformed into robot-frame targets for the G1 end-effectors.

For this project, VisionProTeleop is specialized to the Unitree G1 by designing a robot-specific teleoperation mapping for its upper body. This involves selecting which joints are driven by teleoperation, defining end-effector targets as functions of the headset hand poses, and implementing frame alignments, scaling, and joint-limit handling. The resulting interface is intended to be comfortable for the operator, respect the robot's kinematic constraints, and produce joint-space motion that can be recorded as a sequence of feasible reference trajectories.

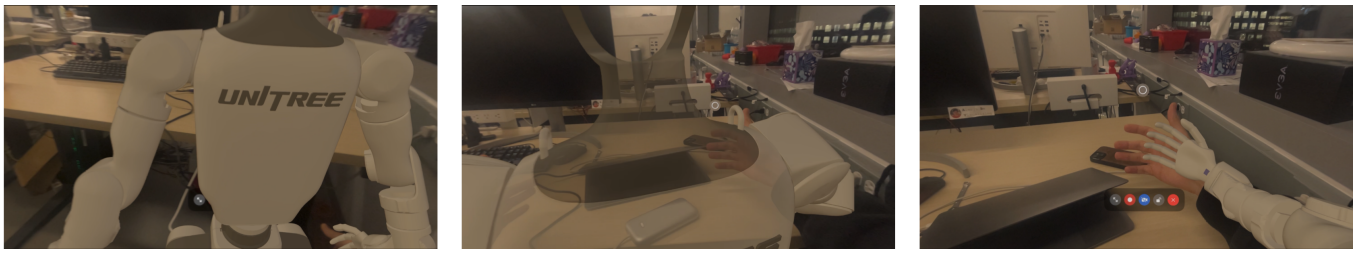


Figure 1: Teleoperation of the G1 Humanoid Robot via VisionProTeleop

4 Problem Statement and Method Overview

Within this broader context, the objective of this project is focused and concrete:

> *Design an upper-body teleoperation interface for the Unitree G1 and make the robot accurately track the resulting trajectories, assessing whether those trajectories are physically feasible and suitable as demonstration data.*

Let $q(t) \in R^n$ denote the joint configuration of the upper body and let $q_d(t)$ denote the joint-space trajectory induced by teleoperation. The goal is to design a control and validation loop such that

$$q(t) \approx q_d(t) \quad \text{and} \quad \dot{q}(t) \approx \dot{q}_d(t), \tag{4}$$

while respecting actuator limits and generating torques that are consistent with the robot's dynamics.

4.1 Teleoperated Demonstration Acquisition and Trajectory Generation

A teleoperation module generates joint-space demonstration trajectories for the humanoid arms. The module connects to an Apple Vision Pro device and receives 6D hand poses for the left and right hands. Each hand pose is represented as a homogeneous transform

$$T_{\text{hand}}(t) \in SE(3), \tag{5}$$

obtained from the wrist tracking output of the headset.

To align the Vision Pro hand frame with the robot's kinematic frame, a fixed rotation correction $R_{\text{corr}} \in SO(3)$ is applied. The desired end-effector pose for each arm is then

$$T_d(t) = \begin{bmatrix} R_{\text{hand}}(t)R_{\text{corr}} & p_{\text{hand}}(t) \\ 0 & 1 \end{bmatrix}, \tag{6}$$

where $R_{\text{hand}}(t)$ and $p_{\text{hand}}(t)$ are the orientation and position of the tracked wrist. Separate correction rotations are used for the left and right hands to account for their mirrored kinematics.

Given $T_d(t)$, a differential inverse kinematics (IK) solver computes joint velocities $\dot{q}(t)$ for the left and right 7-DOF arms. The solver is formulated as a stack of tasks: two end-effector frame tasks (one per wrist) enforce the desired position and orientation, and a posture task regularizes the solution toward a comfortable joint configuration. Joint position, velocity, and collision-avoidance limits are also enforced. At each control step,

$$\dot{q}^\star(t) = \arg\min_{\dot{q}} \sum_i w_i \left\| e_i(q(t), \dot{q}; T_{d,i}(t)) \right\|^2 \tag{7}$$

is solved subject to these limits, and the configuration is updated via

$$q(t + \Delta t) = q(t) + \dot{q}^\star(t)\, \Delta t. \tag{8}$$

While the operator moves their hands, the physics simulator integrates the robot dynamics in real time and the teleoperation module records the resulting motion. The joint positions $q_{\text{pos}} \in R^{T \times n_q}$ are stored at a recording rate of 200 Hz as an `NPZ` file. This file serves as the reference trajectory source for the follower controller described in the next subsection.

4.2 Trajectory Validation via Adaptive Joint-Space Control

The follower robot is controlled in joint space to track a desired trajectory obtained from a reference trajectory. The control design assumes a standard rigid-body manipulator model with n revolute joints,

$$M(q)\, \ddot{q} + C(q, \dot{q})\, \dot{q} + g(q) = \tau, \tag{9}$$

where $q \in R^n$ denotes the joint angles, $M(q)$ is the positive definite inertia matrix, $C(q, \dot{q})\, \dot{q}$ collects Coriolis and centrifugal effects, $g(q)$ is the gravity torque vector, and $\tau \in R^n$ is the vector of commanded joint torques.

The terms $M(q)$, $C(q, \dot{q})\, \dot{q}$, and $g(q)$ are obtained at each time step from the physics engine using inverse dynamics, so that the controller can explicitly compensate for nonlinear dynamics rather than treating them as unknown disturbances.

Tracking errors and desired acceleration. For a given desired trajectory $q_d(t)$, $\dot{q}_d(t)$, $\ddot{q}_d(t)$, the joint-space tracking errors are defined as

$$e(t) = q_d(t) - q(t), \tag{10}$$

$$\dot{e}(t) = \dot{q}_d(t) - \dot{q}(t). \tag{11}$$

The second derivative $\ddot{q}_d(t)$ is numerically differentiated from the desired velocity and then filtered to reduce high-frequency noise. Denote the filtered desired acceleration as $\ddot{q}_{d,\text{filt}}(t)$.

Structure of the control law. The control input is decomposed into three components:

$$\tau = \tau_{\text{dyn}} + \tau_{\text{PD}} + \tau_{\text{adapt}}, \tag{12}$$

where

- τ_{dyn} compensates for the nominal nonlinear dynamics,
- τ_{PD} is a fixed-gain joint-space PD feedback,
- τ_{adapt} is an adaptive term that estimates unknown parameters such as joint friction, and optionally controller gains.

The dynamic compensation term uses the model in (9):

$$\tau_{\text{dyn}}(q, \dot{q}) = M(q)\, \ddot{q}_{d,\text{filt}} + C(q, \dot{q})\, \dot{q} + g(q). \tag{13}$$

This term drives the robot as if it were exactly following the desired motion, leaving the feedback terms to correct residual errors and unmodeled effects.

The fixed-gain feedback term is a standard joint-space PD controller,

$$\tau_{\text{PD}} = K_p\, e + K_d\, \dot{e}, \tag{14}$$

where $K_p, K_d \in R^{n \times n}$ are diagonal and positive definite. These gains provide basic stiffness and damping in joint space.

Parameterization and regression matrix. The adaptive component targets parametric uncertainties that are not captured by the nominal model, primarily joint friction. Let $\theta^\star \in R^m$ denote a vector of unknown parameters, and $\hat{\theta} \in R^m$ its online estimate. The adaptive torque is constructed from a regression matrix $Y(q, \dot{q}, e, \dot{e}) \in R^{n \times m}$ as

$$\tau_{\text{adapt}} = Y(q, \dot{q}, e, \dot{e})\, \hat{\theta}. \tag{15}$$

In the simplest configuration, the parameter vector contains the viscous friction coefficients for each joint,

$$\theta^\star = [\, b_1, \ldots, b_n \,]^\top, \tag{16}$$

so that the regressor has the diagonal structure

$$Y(q, \dot{q}, e, \dot{e}) = \begin{bmatrix} \dot{q}_1 & 0 & \ldots & 0 \\ 0 & \dot{q}_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \dot{q}_n \end{bmatrix}, \tag{17}$$

so that $Y\,\hat{\theta} \approx [\, b_1\dot{q}_1, \ldots, b_n\dot{q}_n \,]^\top$.

An extended configuration also includes the PD gains themselves in the parameter vector, for example

$$\theta^\star = [\, b_1, \ldots, b_n, K_p, K_d \,]^\top, \tag{18}$$

in which case Y is augmented with additional columns built from the error signals e and $\dot{e}$, so that the same estimated gains are shared across all joints. The exact structure of Y depends on which parameters are chosen to be adaptive.

Adaptive update law and Lyapunov function. Define the parameter estimation error

$$\tilde{\theta} = \hat{\theta} - \theta^\star. \tag{19}$$

The adaptive update law is chosen in gradient form,

$$\dot{\hat{\theta}} = -\,\Gamma\, Y(q, \dot{q}, e, \dot{e})^\top \big(e + \lambda\, \dot{e}\big), \tag{20}$$

where $\Gamma \in R^{m \times m}$ is a positive-definite adaptation gain matrix and $\lambda > 0$ is a design constant. In discrete time this is implemented using an Euler step with additional smoothing and saturation to avoid numerical oscillations, but the continuous form in (20) captures the underlying control design.

A Lyapunov candidate for the closed-loop system is

$$V(e, \dot{e}, \tilde{\theta}) = \frac{1}{2} e^\top e + \frac{1}{2} \dot{e}^\top \dot{e} + \frac{1}{2} \tilde{\theta}^\top \Gamma^{-1} \tilde{\theta}. \tag{21}$$

Substituting the dynamics (9), control law (12)–(15), and adaptation rule (20) into the time derivative of V yields

$$\dot{V} \leq 0, \tag{22}$$

under standard assumptions on the regressor Y and the desired trajectory. This guarantees boundedness of the parameter estimates and asymptotic convergence of the tracking errors $e(t)$ and $\dot{e}(t)$ to zero.

In summary, the follower robot employs a joint-space controller that combines model-based dynamic compensation, fixed-gain PD feedback, and an adaptive term driven by a regressor in the joint velocities and tracking errors. The adaptive law updates friction (and optionally gain) estimates in a Lyapunov framework, providing robust tracking of the reference motion despite unmodeled joint dissipation and parameter uncertainty.

5 Experimental Evaluation

5.1 Reference Trajectories

Two types of reference trajectories are used to evaluate the adaptive joint-space controller and to assess the feasibility of the teleoperated motions.

Teleoperated upper-body motion. A first trajectory is obtained from the teleoperation interface. The operator moves both arms of the Unitree G1 upper body using the VisionPro-based teleoperation mapping, producing a multi-joint, time-varying motion that includes coupled rotations of the shoulder, elbow, and wrist joints. The resulting joint-space reference is denoted

$$q_d(t) \in R^n,$$

and is recorded over a finite horizon $t \in [0, T_{\text{tele}}]$. This trajectory represents a typical "rich" teleoperated motion with varying speed and direction.

Sinusoidal shoulder motion. A second trajectory provides a simpler, more structured reference. Only the first joint of the left and right arms (a shoulder joint in each arm) is commanded to follow a sinusoidal trajectory of the form

$$q_{d,1}^{(\text{L/R})}(t) = q_0 + A\sin(\omega t), \tag{23}$$

while the remaining joints are held near a nominal configuration. This produces a single-degree-of-freedom periodic motion in each arm, with all other joints remaining approximately constant. This sinusoidal reference is used as a baseline case with simple kinematics and a predictable frequency content.

In all experiments, the same adaptive joint-space controller is used, and only the reference trajectory $q_d(t)$ is changed between the teleoperated and sinusoidal cases.

5.2 Evaluation Metrics and Plots

Tracking performance is evaluated in joint space using the error

$$e(t) = q_d(t) - q(t), \tag{24}$$

where $q(t)$ is the actual joint configuration produced by the controller. Four types of plots are generated.

Per-joint tracking for teleoperated and sinusoidal motion. For the teleoperated trajectory, the desired and actual joint positions $q_d^i(t)$ and $q^i(t)$ are plotted for each joint of the right and left arms. These time series visualize how closely each joint follows its reference during a complex, multi-joint motion. For the sinusoidal trajectory the plots emphasize how the controller handles a simple, single-joint periodic motion and allow direct visual comparison of phase lag and amplitude accuracy.

Lyapunov function evolution. Finally, the value of the Lyapunov function candidate $V(t)$, defined in the control design section, is plotted over time for both the teleoperated and sinusoidal trajectories. This provides a qualitative view of stability and convergence: a generally decreasing $V(t)$ indicates that the tracking errors and parameter estimation errors remain bounded and tend to decrease.

6 Results and Discussion

The experiments indicate that the adaptive joint-space controller is able to track both the complex teleoperated motion and the simpler sinusoidal motion with good accuracy, while maintaining consistent Lyapunov behavior.

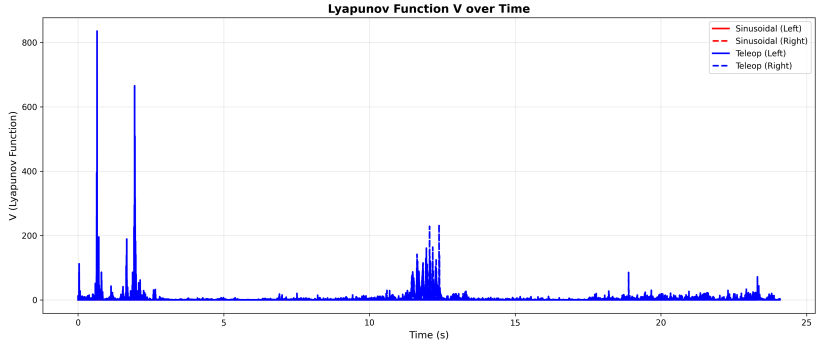


Figure 2: Measured Lyapunov function V(t) for a teloperated trajectory and a simple sinusoidal trajectory.

The Lyapunov function $V(t)$ exhibits distinct behavior for the two trajectories. For the sinusoidal case, after a short initial transient $V(t)$ quickly settles to values very close to zero (on the order of 10^{-2}) and then oscillates tightly around this small baseline. This indicates that, once the controller has synchronized with the periodic motion, both the tracking error and the parameter error remain very small. For the teleoperated trajectory, $V(t)$ stays near zero for most of the motion but displays two pronounced spikes, reaching values between approximately 200 and 800 at specific moments. These peaks coincide with segments where the reference changes more abruptly, and are followed by a rapid return of $V(t)$ to small values rather than sustained growth. Overall, $V(t)$ remains nonnegative and mostly close to zero in both experiments, which is consistent with the Lyapunov-based design and indicates that the adaptive controller maintains stability while adjusting its parameters, even when subjected to more irregular, human-generated references.

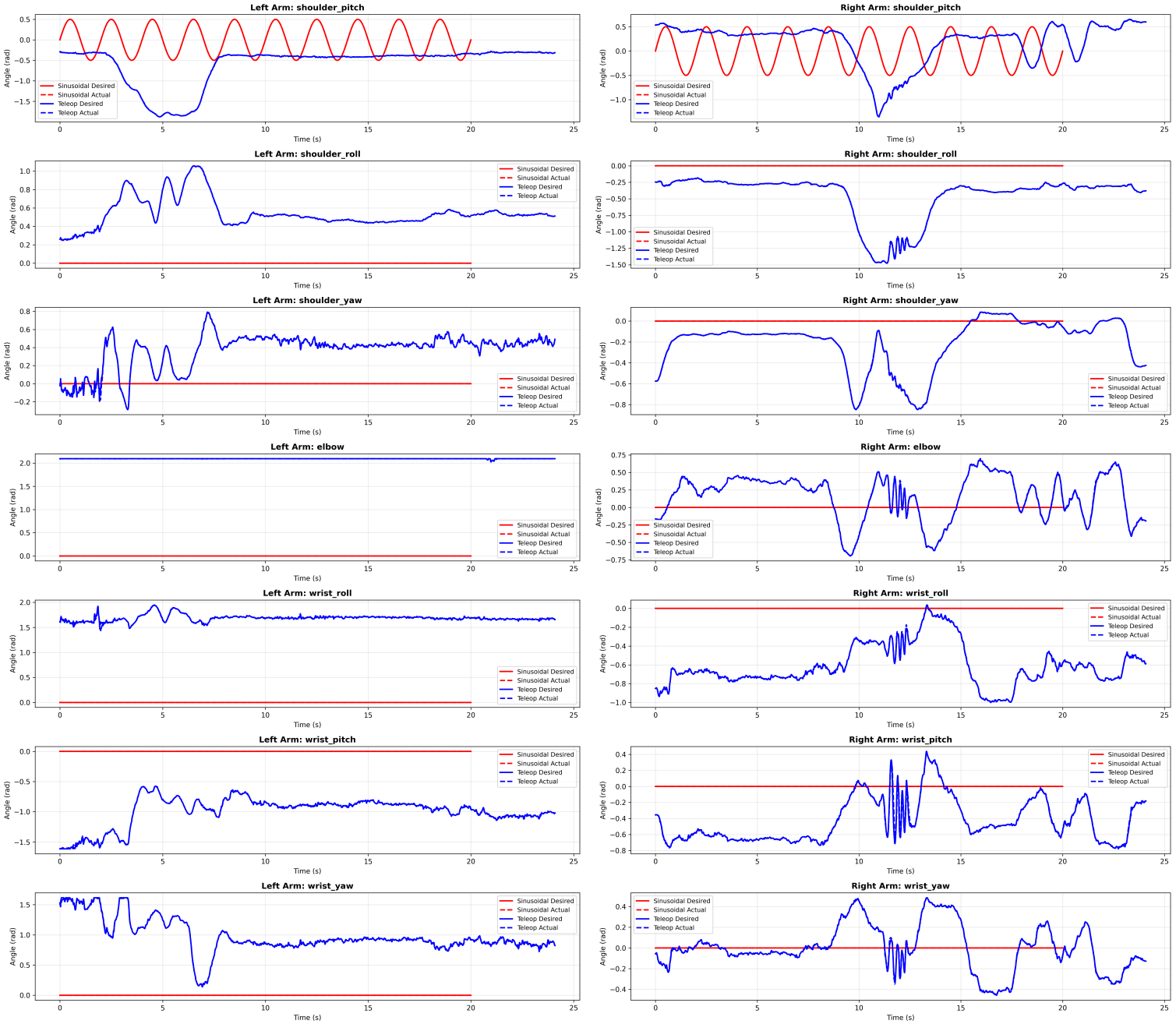


Figure 3: Per-joint plots of desired trajectory versus actual trajectory.

For the teleoperated trajectory, the per-joint plots show that the actual motion closely follows the desired joint trajectories for both arms, including during segments where multiple joints move simultaneously. Deviations between $q_d^i(t)$ and $q^i(t)$ remain bounded and typically occur when the reference changes rapidly, which is expected in human-

driven motion. The norm of the tracking error $\|e(t)\|_2$ remains relatively small over the entire teleoperation horizon, confirming that the generated teleoperation trajectories are kinematically and dynamically feasible for the model-based adaptive controller.

In the sinusoidal experiment, the first joint of each arm tracks the periodic reference with small amplitude error and limited phase lag. Since the remaining joints are nearly stationary, this case isolates the controller's behavior on a single, well-excited degree of freedom. The sinusoidal plots provide a baseline: they show that, for a simple reference, the controller produces smooth tracking with no instability or drift.

Taken together, these observations support the interpretation of the follower controller as a validation tool for teleoperated trajectories. The fact that the controller can track a human-generated, multi-joint upper-body motion with small joint-space error and well-behaved Lyapunov evolution suggests that the teleoperation mapping produces trajectories that are not only visually reasonable but also dynamically compatible with the robot model. This makes the recorded teleoperated motions suitable candidates for use as demonstration data in subsequent learning-based controllers.

7 Conclusion and Future Work

The primary goal of this project was not to tune a controller for optimal performance, but to establish a working teleoperation and validation pipeline for the Unitree G1 upper body. The emphasis was on (i) designing a VisionPro-based teleoperation interface for the G1 arms and (ii) using an adaptive joint-space controller to assess whether the resulting trajectories are physically feasible and usable as demonstration data. In this sense, the project was successful: for both the complex teleoperated motion and the simpler sinusoidal shoulder motion, the actual joint trajectories closely match their desired references, and the Lyapunov function remains near zero aside from short, bounded transients.

The current controller and experiments should be viewed as a starting point rather than a final design. The adaptive law so far focuses primarily on joint friction, and the gains were chosen to obtain reasonable behavior rather than systematically optimized. Future work should extend the adaptation to additional parameters (for example, selected inertial or gravity-related terms and controller gains), and perform more careful tuning of the feedback and adaptation gains to improve transient performance and robustness.

From a data perspective, only a single teleoperated trajectory and a single sinusoidal trajectory were used here to validate the teleoperation mapping and the basic feasibility of the approach. A natural next step is to collect a larger and more diverse set of teleoperated motions, including different arm configurations, speeds, and task contexts. These additional trajectories would allow more thorough validation of the teleoperation pipeline and provide a richer dataset for training or evaluating learning-based controllers such as SoftMimic-style policies. Ultimately, extending this framework to lower-body motion and multi-contact scenarios would move the system closer to full whole-body teleoperation and compliant control of the complete humanoid.

8 Appendix

For the full implementation, see the project repository on GitHub: `https://github.com/mau-pereira/UROP_ImprobableAI/tree/main/Project`.