R&D Project Report

Arm-to-Arm Bilateral Control of Robotic Manipulators

Farhaan Nasir

Under the Guidance of

Prof. Ameer Mulla

Academic Year: Autumn–2025

Contents

Abstract

This report presents the development of an arm-to-arm bilateral control framework using the Interbotix Reactor X150 as the master manipulator and the Interbotix Viper X300 as the slave. A joint-space bilateral architecture was implemented to enable real-time synchronization of joint angles and force reflection when the slave interacts with the environment. The system was validated through staged simulation in RViz and PyBullet, demonstrating stable motion coupling and correct feedback behavior. Hardware deployment achieved partial bilateral motion, but consistent synchronized behavior was limited by control-loop execution challenges and hardware-related constraints. The findings highlight both the strengths of joint-space bilateral control in simulation and the practical considerations required when transferring such systems to physical robots. Future work includes implementing full inverse and forward kinematics to enable true Cartesian-space bilateral control.

1 Introduction

Bilateral control enables two robotic manipulators to exchange motion and force information in real time. Such systems help replicate human-like perception by allowing one robot (the master) to feel and react to forces experienced by another robot (the slave). This capability is fundamental for teleoperation, remote manipulation, and collaborative robotics.

In this project, bilateral control was explored using two Interbotix robotic arms: the Reactor X150 (5-DOF) as the master and the Viper X300 (6-DOF) as the slave. The central aim was to create a functional robotic–robotic bilateral control loop using joint-angle mapping for motion synchronization and torque-based feedback for haptic response.

The work follows a structured progression: formulation of the bilateral control model, implementation of the control loop in simulation, verification of bilateral behavior under contact scenarios, and experimental deployment on the physical robotic platform.

Project Repository: `github.com/avocadoxD0/RnD-Project`

2 System Overview

2.1 Hardware Setup

- **Master Arm:** Interbotix Reactor X150 (5-DOF)
- **Slave Arm:** Interbotix Viper X300 (6-DOF)
- **Actuators:** Dynamixel servo motors

- **Control Interface:** Interbotix Python API with ROS integration

The master arm is operated directly, and its joint angles are streamed to the slave arm through a real-time control loop. The slave arm encounters external objects in the workspace, and the resulting interaction forces form the feedback signal to the master.

2.2 Bilateral Control Architecture

The bilateral control loop consists of:

1. **Forward Path:** Master joint angles θ_m mapped to slave joint commands θ_s
2. **Environment Interaction:** Slave experiences external contact forces
3. **Feedback Path:** Reaction torques τ_s mapped back to the master
4. **Master Response:** Force feedback influences the master's motion

The basic mapping relation is:

$$\theta_s = M\,\theta_m$$

where M is a predefined joint-angle mapping matrix.

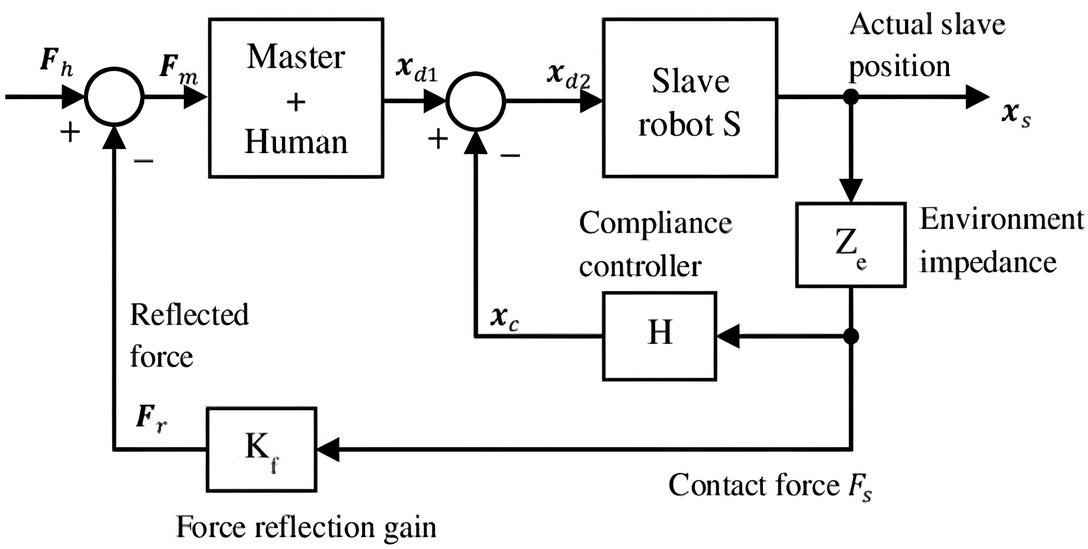

Figure 1: Overall bilateral control architecture showing forward joint-angle mapping, force reflection, and compliance control.

3 Mathematical Formulation

This section presents the complete mathematical structure of the bilateral control system implemented between the Reactor X150 (master) and Viper X300 (slave) robotic manipulators. The formulation covers joint-space mapping, forward and feedback control flows, dynamic modeling, force reflection, admittance control on the master side, and stability considerations.

3.1 System Representation

Let

$$\theta_m \in \mathbb{R}^{n_m}, \quad \theta_s \in \mathbb{R}^{n_s}$$

denote the joint position vectors of the master and slave manipulators, respectively. Their corresponding joint velocities and torques are represented as

$$\omega_m, \omega_s, \quad \tau_m, \tau_s.$$

The bilateral control loop consists of two coupled flows:

1. **Position flow:** Master → Slave
2. **Force flow:** Slave → Master

3.2 Joint-Space Mapping

Because the Reactor X150 (5-DoF) and Viper X300 (6-DoF) differ in degrees of freedom, a mapping matrix

$$M \in \mathbb{R}^{n_s \times n_m}$$

is used to map the master joint angles into the slave joint space:

$$\theta_s = M\theta_m.$$

For this project, the mapping matrix was defined as:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Inverse or mirror-reverse mapping is also supported:

$$\theta_s = -M\theta_m.$$

3.3 Manipulator Dynamics

Both manipulators follow the standard joint-space rigid-body dynamic equation:

$$M_i(\theta_i)\,\ddot{\theta}_i + C_i(\theta_i, \dot{\theta}_i)\,\dot{\theta}_i + G_i(\theta_i) = \tau_i, \quad i \in \{m, s\}$$

where:

- M_i is the inertia matrix,
- C_i contains Coriolis and centrifugal terms,
- G_i is the gravity vector,
- τ_i is the applied joint torque vector.

These dynamics influence responsiveness during bilateral motion—particularly in hardware execution where friction, communication delays, and actuator limits introduce asymmetries.

3.4 Force Reflection Model

When the slave end-effector makes contact with the environment, the interaction force F_{ext} generates joint torques according to:

$$\tau_s = J_s^\top F_{\text{ext}},$$

where J_s is the slave Jacobian matrix.

These torques are mapped back to the master using:

$$\tau_m = K_f \, M^\top \tau_s,$$

where K_f is a gain factor controlling the intensity of perceived force feedback.

3.5 Admittance Control on the Master

If the master arm does not directly accept torque commands, an admittance model is used to convert incoming feedback torque into a compliant positional offset:

$$B \, \Delta\dot{\theta}_m + K \, \Delta\theta_m = -\tau_m,$$

where B and K are virtual damping and stiffness parameters.

The discrete-time update law is:

$$\Delta\theta_m(t + \Delta t) = \Delta\theta_m(t) - \frac{\Delta t}{B} \left(K \, \Delta\theta_m(t) + \tau_m \right),$$

and the commanded master joint configuration becomes:

$$\theta_{m,\text{cmd}} = \theta_m + \Delta\theta_m.$$

3.6 Stability Considerations

A key requirement for bilateral control is passivity, enforced through:

$$\dot{E} = \tau_m^\top \dot{\theta}_m + \tau_s^\top \dot{\theta}_s \leq 0,$$

ensuring the system does not inject energy.

Additional measures include:

- **Gain tuning:** small K_f and sufficiently large B to avoid oscillation.
- **Filtering:** low-pass torque filtering,

$$\tau_s^{(t)} = (1-\alpha)\,\tau_s^{(t-1)} + \alpha\,\tau_s,$$

- **Dead-zone:** ignore small torques,

$$|\tau_i| < \tau_{\text{threshold}} \Rightarrow \tau_i = 0.$$

3.7 Complete Bilateral Control Flow

The final bilateral loop can be summarized as:

$$\begin{aligned}
\text{(1) Master} \to \text{Slave:}\ & \theta_s = M\theta_m, \\
\text{(2) Slave} \to \text{Environment:}\ & \tau_s = J_s^\top F_{\text{ext}}, \\
\text{(3) Environment} \to \text{Master:}\ & \tau_m = K_f M^\top \tau_s, \\
\text{(4) Master Response:}\ & \theta_{m,\text{cmd}} = \theta_m + \Delta\theta_m.
\end{aligned}$$

3.8 Tuning Parameters

Symbol	Meaning	Typical Range
K_f	Force feedback gain	0.05–0.2
B	Damping coefficient	3–10
K	Stiffness (admittance)	20–50
$\tau_{\text{threshold}}$	Torque dead-zone	0.02–0.05 Nm
α	Filter constant	0.05–0.2

3.9 Summary

The bilateral control architecture integrates forward joint-angle mapping, force reflection computed through Jacobian-transformed torques, and an admittance-based correction on the master. This structure enables real-time, responsive, and stable robot–robot teleoperation in simulation and forms the basis for extension to full Cartesian bilateral control.

4 Methodology

4.1 Simulation Workflow

- Initial testing of master–slave and inverse master–slave motion in **RViz**.
- Transition to **PyBullet** for dynamic simulation and full bilateral loop validation.
- Verification of correct contact detection and force reflection behavior in simulation.

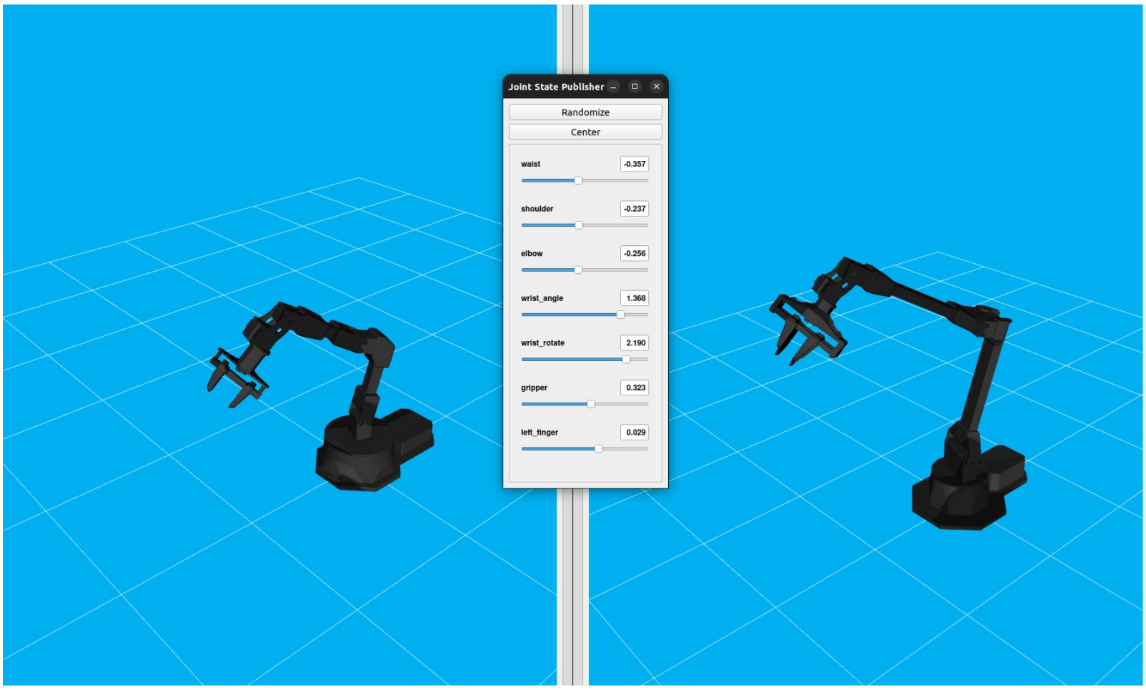

Figure 2: Initial master–slave joint-space mapping tested in RViz using Interbotix URDF models.

4.2 Control Loop Implementation

- Real-time reading of master joint angles.
- Mapping of angles to slave commands via matrix M.
- Computation of reaction torques in the slave.
- Force reflection to master using gain-controlled feedback.

4.3 Hardware Deployment

The final stage involved deploying the bilateral control loop on the physical X150–X300 pair. Both arms were able to move out of rest positions and execute partial bilateral motion, but maintaining stable synchronization was challenging due to software timing, actuator behavior, and hardware constraints.

5 Observations

- Full bilateral control in PyBullet showed stable synchronization, obstacle detection, and consistent force-feedback behavior.
- Simulation demonstrated robust tracking and predictable activation of the bilateral loop during wall contact events.
- Hardware trials achieved partial bilateral motion; however, sustained synchronized operation was limited by control-loop implementation issues and hardware-related constraints.
- The contrast between simulation and hardware highlights the practical challenges of real-world actuation, communication delays, and physical uncertainties.

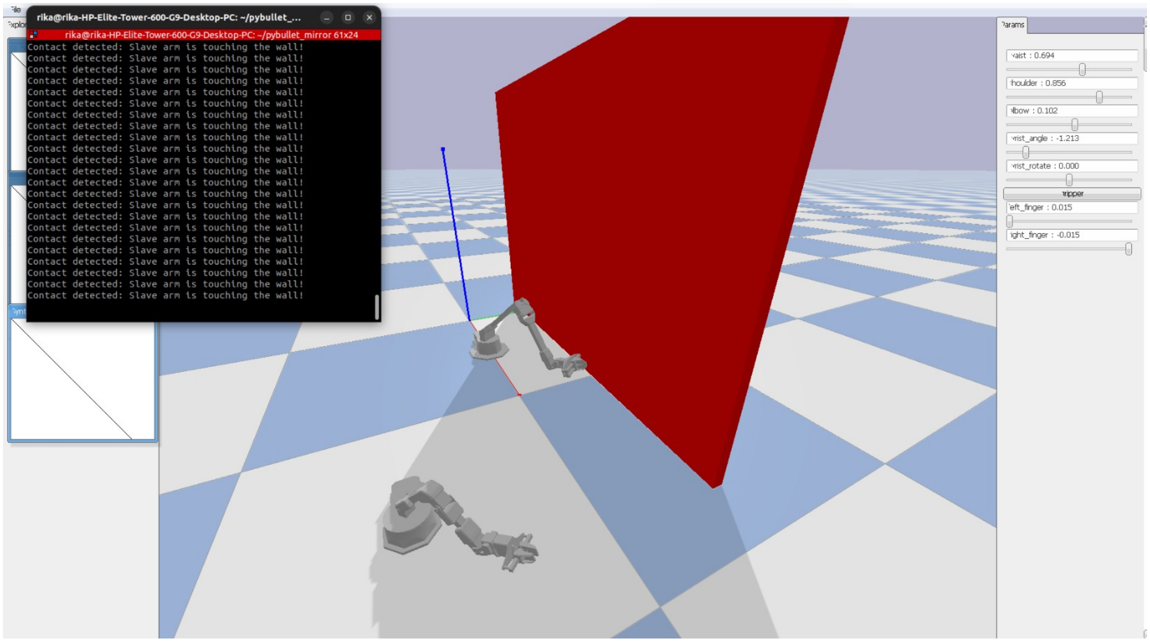

Figure 3: Full bilateral control in PyBullet showing real-time obstacle contact detection and activation of the force-feedback loop.

6 Difficulties Faced During Hardware Deployment

Implementing the bilateral control framework on the physical Reactor X150–Viper X300 setup introduced several challenges that were not observed in simulation.

- **Timing and control-loop execution:** The real-time command and feedback loop experienced inconsistencies arising from Python-level execution delays and communication jitter, affecting synchronized bilateral behavior.
- **Actuator and hardware constraints:** The Dynamixel servo motors exhibited thermal limitations during continuous operation, leading to temporary shutdowns or reduced responsiveness, which disrupted sustained bilateral motion.
- **Force estimation uncertainties:** Although torque-based feedback performed well in simulation, estimating reaction torques on hardware was affected by sensor noise, filtering delays, and actuator compliance.
- **Mismatch between simulation and reality:** Differences in friction, joint elasticity, and environmental contact behavior made it difficult for the physical system to reproduce the stable bilateral motion observed in PyBullet.

These challenges collectively limited the system to partial bilateral motion on hardware, even though the entire control loop functioned correctly in simulation.

Figure 4: Hardware deployment of the bilateral control system on the Reactor X150 (master) and Viper X300 (slave). Partial bilateral motion was achieved.

7 Conclusion

- Joint-space bilateral control was successfully implemented and validated in simulation using the Interbotix master–slave arm pair.
- PyBullet simulations confirmed stable synchronized motion and realistic force reflection under environmental interaction.
- Hardware deployment achieved partial bilateral movement but exposed limitations in timing accuracy, control implementation, and hardware reliability.
- Future work involves integrating full IK/FK for Cartesian bilateral control and enhancing low-level control, communication stability, and sensing robustness for practical real-world deployment.

8 Future Scope

The current implementation demonstrates the feasibility of joint-space bilateral control, but several improvements are required to achieve fully practical deployment on physical robotic systems.

- **Integration of IK/FK for Cartesian bilateral control:** Extending the system with complete inverse kinematics (IK) and forward kinematics (FK) will enable true Cartesian-space bilateral motion, improving usability, precision, and workspace accessibility.
- **Enhanced low-level control and communication:** Implementing high-frequency, latency-minimized control loops and optimizing ROS communication pathways will improve synchronization performance.
- **Improved sensing and force estimation:** Incorporating better torque estimation, filtering strategies, or external force sensors can enhance the robustness of force reflection.
- **Thermal and actuator performance management:** Actuator cooling improvements, duty-cycle optimization, or alternative actuation strategies can help mitigate thermal shutdowns during prolonged operation.
- **Real-world task integration:** Once stability is improved, the bilateral system can be extended to practical tasks involving teleoperation, collaborative manipulation, or dual-arm coordination.

References

[1] S. G. Tzafestas and A.-I. Mantelos, *Time Delay and Uncertainty Compensation in Bilateral Telerobotic Systems: State-of-the-Art with Case Studies*, Annual Reviews in Control, 2012.

[2] S. M. Yoon, W. J. Kim, and M. C. Lee, *Design of Bilateral Control for Force Feedback in Surgical Robot*, International Journal of Control, Automation, and Systems, 2015.

[3] A. Hattori, *Design and Implementation of a Pair of Robot Arms with Bilateral Teleoperation*, Bachelor's Thesis, Department of Mechanical Engineering, MIT, 2019.

[4] S. Echeandia and P. M. Wensing, *Numerical Methods to Compute the Coriolis Matrix and Christoffel Symbols for Rigid-Body Systems*, Journal of Mechanisms and Robotics, 2021.

[5] C. Zhou, L. Zhao, H. Wang, L. Chen, and Y. Zheng, *A Bilateral Dual-Arm Teleoperation Robot System with a Unified Control Architecture*, IEEE RO-MAN Conference, 2021.