RB-A630-2 Homogeneous Teleoperation Guide

Table of Contents

- 1. Software Overview
- 2. Document Objective and Scope
- 3. High-Risk Warning
  - 3.1 Mandatory Checks Before Starting
  - 3.2 Risks During Operation
- 4. Functional Topology
  - 4.1 Web Quick Start
- 5. Command-Line Startup and Inspection
  - 5.1 Environment Setup
  - 5.2 Start One Arm Pair
  - 5.3 Start Four Arms
  - 5.4 Shutdown Flow
- 6. Runtime Confirmation and Success Criteria
- 7. Common Issues and Abnormal Handling
- 8. Related Paths

1. Software Overview

RB-A630-2 homogeneous teleoperation provides master-slave joint-space following for four six-axis robot arms. The system contains four arm instances: `robot_1`, `robot_2`, `robot_3`, and `robot_4`. `robot_1` and `robot_2` are master arms that perform zero-force dragging, while `robot_3` and `robot_4` follow their corresponding master arms.

The fixed pairing is:

- Pair 1: `robot_1` master dragging, `robot_3` slave following.
- Pair 2: `robot_2` master dragging, `robot_4` slave following.

The teleoperation node reads the master arm joint angles and publishes target positions to the slave position controllers. The master arm zero-force torque is output by the RB-A630-2 zero-force drag node in `raybot_dynamics_control`.

2. Document Objective and Scope

This document guides on-site debugging and operation personnel through RB-A630-2 homogeneous teleoperation startup, runtime confirmation, shutdown, and abnormal handling.

Applicable scope:

- Workspace: `/home/raybot/raybot_core_ws`
- Model workspace: `/home/raybot/raybot_models`
- Homogeneous teleoperation package: `raybot_homogeneous_teleop`
- Zero-force drag package: `raybot_dynamics_control`
- Robot model: `rb_a630_2`
- Arm instances: `robot_1` , `robot_2` , `robot_3` , `robot_4`

Out of scope:

- Teleoperation algorithm design and secondary development
- Zero-force drag parameter calibration details
- Non-RB-A630-2 four-arm topologies

3. High-Risk Warning

> ***High-risk warning***: *Homogeneous teleoperation causes the slave arms to follow the master arms and therefore involves real robot motion risk. Before starting, confirm that no one remains in the robot motion range, the emergency-stop button is available, and assume that the software chain may experience delay, direction errors, state loss, or abnormal exit. If unintended motion, collision risk, personnel risk, or abnormal status feedback occurs, the physical emergency-stop always takes priority over any command-line operation.*

3.1 Mandatory Checks Before Starting

Before starting, confirm:

- The hazardous area around all four arms has been cleared.
- A person is holding the master arms before zero-force drag starts, to prevent sudden sinking due to gravity or parameter mismatch.

- The physical emergency-stop is immediately reachable.
- The ROS2 Humble environment, `raybot_models`, and `raybot_core_ws` have all been sourced correctly.
- No old `/controller_manager` instance is still running.
- `/joint_states` keeps updating.
- `/raybot_robot/hardware_ready` becomes `true`.
- The first run must be validated with a single pair, small motion, and a low-risk direction.

If Ray Studio or another terminal is already controlling the robot, stop the old control chain first to avoid starting a duplicate `/controller_manager`.

You can use:

```
curl -fsS http://127.0.0.1:8889/api/robot-control/stop || true
```

3.2 Risks During Operation

Pay special attention during operation:

- `robot_3` follows `robot_1`, and `robot_4` follows `robot_2`. No one may stand near the slave arms.
- Zero-force dragging does not mean completely weightless, hovering, or absolutely safe.
- Slave following involves control-cycle delay, communication delay, state refresh delay, and speed limiting.
- Under `follow_mode:=absolute`, if the initial joint-angle difference between master and slave is too large, the teleoperation node may refuse to start or may produce a large target jump.
- Do not make large motions before the master-slave pairing and direction are confirmed.
- If zero-force drag becomes abnormal, a slave arm jitters, the follow direction is wrong, or the state times out, stop immediately.

4. Functional Topology

Control relationships:

Role	Arm	Control Mode	Controller
Master 1	robot_1	Zero-force drag	robot_1_arm_eff_controller
Master 2	robot_2	Zero-force drag	robot_2_arm_eff_controller
Slave 1	robot_3	Position following	robot_3_arm_pos_controller
Slave 2	robot_4	Position following	robot_4_arm_pos_controller

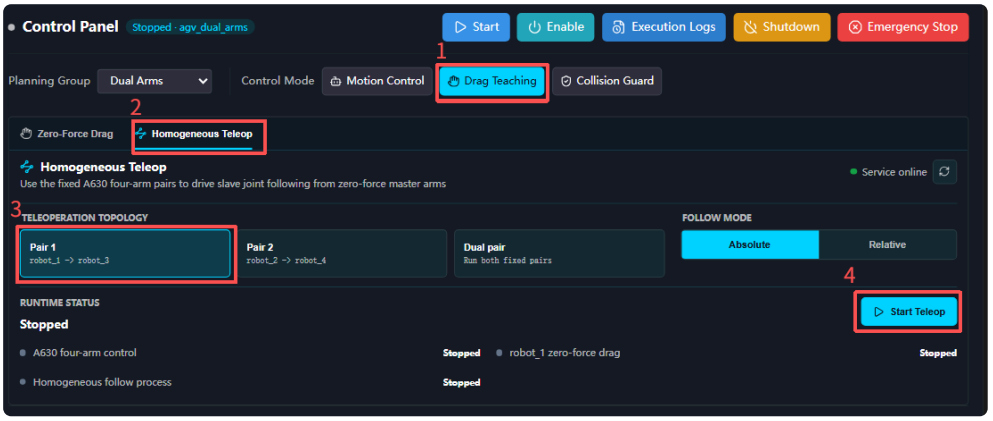
Key nodes and topics:

Item	Name
Single-pair teleop launch	single_pair_homogeneous_teleop.launch.py
Four-arm teleop launch	four_arms_homogeneous_teleop.launch.py
Zero-force drag launch	zero_force_drag_rb_a630_2.launch.py
Master 1 zero-force node	zero_force_drag_robot_1
Master 2 zero-force node	zero_force_drag_robot_2
Master 1 effort command	/robot_1_arm_eff_controller/commands
Master 2 effort command	/robot_2_arm_eff_controller/commands
Slave 1 position command	/robot_3_arm_pos_controller/commands
Slave 2 position command	/robot_4_arm_pos_controller/commands

4.1 Web Quick Start

If Ray Studio is already available on site, start from the Drag Teaching area on the Robot Control page and enter the homogeneous teleoperation panel.

Follow the sequence shown in the figure below:



Recommended steps:

1. Open the Robot Control page and click Drag Teaching .

2. Click `Homogeneous Teleoperation`.
3. Select the target pair.
4. Click `Start Homogeneous Teleoperation` to enable the function.

5. Command-Line Startup and Inspection

5.1 Environment Setup

In each terminal, enter the workspace and source the environment:

```
cd /home/raybot/raybot_core_ws
source /opt/ros/humble/setup.bash
source /home/raybot/raybot_models/install/setup.bash
source install/setup.bash
```

Notes:

- `raybot_models` must be sourced first, otherwise the `rb_a630_2_four_arms` model resources may not be found.
- Do not close the terminal that starts the hardware bringup.
- Do not close the terminals that start zero-force drag or homogeneous teleoperation.

5.2 Start One Arm Pair

This section uses pair 1 as the example: `robot_1` as the master arm and `robot_3` as the slave arm.

TERMINAL 1: START SINGLE-PAIR HARDWARE BRINGUP

```
cd /home/raybot/raybot_core_ws
source /opt/ros/humble/setup.bash
source /home/raybot/raybot_models/install/setup.bash
source install/setup.bash

ros2 launch raybot_homogeneous_teleop single_pair_homogeneous_teleop.launch.py \
  pair:=1 \
  start_bringup:=true \
  start_teleop:=false
```

Wait until the following appears:

```
Hardware ready state changed: READY
```

Do not close terminal 1.

Terminal 2: confirm hardware and controllers

```
cd /home/raybot/raybot_core_ws
source /opt/ros/humble/setup.bash
source /home/raybot/raybot_models/install/setup.bash
source install/setup.bash

ros2 topic echo --once /raybot_robot/hardware_ready
ros2 control list_controllers
```

You should see:

```
data: true
```

The controllers should include:

```
joint_state_broadcaster        active
robot_1_arm_pos_controller     active
robot_1_arm_eff_controller     inactive
robot_3_arm_pos_controller     active
robot_3_arm_eff_controller     inactive
```

Terminal 3: start robot_1 zero-force drag

Hold the master arm before startup and make sure the emergency-stop is available.

```
cd /home/raybot/raybot_core_ws
source /opt/ros/humble/setup.bash
source /home/raybot/raybot_models/install/setup.bash
source install/setup.bash

ros2 launch raybot_dynamics_control zero_force_drag_rb_a630_2.launch.py \
  robot_id:=robot_1
```

Do not close terminal 3.

Terminal 4: start robot_1 to robot_3 teleoperation

```
cd /home/raybot/raybot_core_ws
source /opt/ros/humble/setup.bash
source /home/raybot/raybot_models/install/setup.bash
source install/setup.bash

ros2 launch raybot_homogeneous_teleop single_pair_homogeneous_teleop.launch.py \
  pair:=1 \
```

```
  start_bringup:=false \
  start_teleop:=true \
  master_effort_source:=external \
  follow_mode:=absolute
```

For pair 2, use the same startup flow but change `pair:=1` to `pair:=2`, and start zero-force drag with:

```
ros2 launch raybot_dynamics_control zero_force_drag_rb_a630_2.launch.py \
  robot_id:=robot_2
```

Pair 2 corresponds to `robot_2` as the master and `robot_4` as the slave.

5.3 Start Four Arms

Four-arm mode runs both teleoperation pairs at the same time:

- `robot_1 -> robot_3`
- `robot_2 -> robot_4`

Terminal 1: start four-arm hardware

```
cd /home/raybot/raybot_core_ws
source /opt/ros/humble/setup.bash
source /home/raybot/raybot_models/install/setup.bash
source install/setup.bash

ros2 launch raybot_homogeneous_teleop four_arms_homogeneous_teleop.launch.py \
  start_bringup:=true \
  start_teleop:=false
```

Wait for terminal 1 to finish starting. Do not close it.

Terminal 2: check hardware and controllers

```
cd /home/raybot/raybot_core_ws
source /opt/ros/humble/setup.bash
source /home/raybot/raybot_models/install/setup.bash
source install/setup.bash

ros2 topic echo --once /raybot_robot/hardware_ready
ros2 control list_controllers
```

You must see:

```
data: true
```

Terminal 3: start robot_1 zero-force drag

```
cd /home/raybot/raybot_core_ws
source /opt/ros/humble/setup.bash
source /home/raybot/raybot_models/install/setup.bash
source install/setup.bash

ros2 launch raybot_dynamics_control zero_force_drag_rb_a630_2.launch.py \
  robot_id:=robot_1
```

Terminal 4: start robot_2 zero-force drag

```
cd /home/raybot/raybot_core_ws
source /opt/ros/humble/setup.bash
source /home/raybot/raybot_models/install/setup.bash
source install/setup.bash

ros2 launch raybot_dynamics_control zero_force_drag_rb_a630_2.launch.py \
  robot_id:=robot_2
```

Terminal 2: switch both master arms to effort

Make sure terminals 3 and 4 are running normally before executing:

```
cd /home/raybot/raybot_core_ws
source /opt/ros/humble/setup.bash
source /home/raybot/raybot_models/install/setup.bash
source install/setup.bash

ros2 service call /raybot_robot/switch_robot_1_arm_to_effort \
  std_srvs/srv/Trigger "{}"

ros2 service call /raybot_robot/switch_robot_2_arm_to_effort \
  std_srvs/srv/Trigger "{}"
```

Check the controllers:

```
ros2 control list_controllers
```

You should see these controllers as active :

```
joint_state_broadcaster
robot_1_arm_eff_controller
robot_2_arm_eff_controller
robot_3_arm_pos_controller
robot_4_arm_pos_controller
```

First verify that robot_1 and robot_2 zero-force drag are normal. After confirming that the master-arm drag feel and direction are correct, start homogeneous teleoperation.

TERMINAL 5: START DUAL TELEOPERATION PAIRS

```
cd /home/raybot/raybot_core_ws
source /opt/ros/humble/setup.bash
source /home/raybot/raybot_models/install/setup.bash
source install/setup.bash

ros2 launch raybot_homogeneous_teleop four_arms_homogeneous_teleop.launch.py \
  start_bringup:=false \
  start_teleop:=true \
  master_effort_source:=external \
  follow_mode:=absolute
```

5.4 Shutdown Flow

Recommended shutdown order:

1. Stop manual dragging and confirm all four arms are stable.
2. Close the homogeneous teleoperation terminal.
3. Close the robot_1 and robot_2 zero-force drag terminals.
4. Close the hardware bringup terminal.
5. Confirm again that the robot is no longer receiving teleoperation or zero-force torque commands.

To stop zero-force drag by service:

```
ros2 service call /zero_force_drag_robot_1/shutdown std_srvs/srv/Trigger "{}"
ros2 service call /zero_force_drag_robot_2/shutdown std_srvs/srv/Trigger "{}"
```

For single-pair operation, only shut down the master-arm zero-force drag node that was actually started.

6. Runtime Confirmation and Success Criteria

The following conditions must all be satisfied to consider homogeneous teleoperation running normally:

- /raybot_robot/hardware_ready returns data: true .
- /joint_states keeps updating.

- joint_state_broadcaster is active .
- The master-arm eff_controller is active .
- The slave-arm pos_controller is active .
- The master arm can be dragged smoothly with no obvious inverse pushing, rapid sinking, or jitter.
- The slave arms follow in the correct direction, and the first small-move validation passes.
- No continuous errors appear about state timeout, controller switching failure, or excessive initial pose error after teleoperation starts.

  Common inspection commands:

```
ros2 topic echo --once /raybot_robot/hardware_ready
ros2 control list_controllers
ros2 topic echo --once /joint_states
ros2 node list | grep homogeneous_teleop
ros2 node list | grep zero_force_drag
```

7. Common Issues and Abnormal Handling

Symptom	Possible Cause	Handling	Immediate Shutdown Required
Bringup reports an existing /controller_manager	Ray Studio or another terminal is already controlling the robot	Stop the old control chain first, then restart; do not run two /controller_manager instances at the same time	No
/raybot_robot/hardware_ready is not true	Hardware not ready, motors not connected, control group mismatch, or bringup not finished	Check the bringup terminal logs and confirm that all required motors and control groups are ready	No
The master arm sinks or pushes back immediately after zero-force drag starts	Zero-force parameters, gravity compensation, joint state, or effort controller state are abnormal	Hold the master arm and stop zero-force drag immediately, then check the controller state and /joint_states	Yes
Homogeneous teleoperation fails to start and reports excessive initial pose error	Under follow_mode:=absolute, the initial master-slave joint-angle difference exceeds the safety threshold	Move the master and slave poses closer, then start teleoperation again	No
The slave arm does not follow	Teleoperation node not started, slave position controller not active, /joint_states incomplete, or the master state timed out	Check ros2 node list, ros2 control list_controllers, and /joint_states	No
The slave follow direction or pairing is not as expected	Wrong pair selected at startup, or the fixed topology was misunderstood	Confirm that pair:=1 is robot_1 -> robot_3, and pair:=2 is robot_2 -> robot_4	Yes
The slave suddenly stops following and holds position	The master /joint_states timed out or became incomplete	Check the state feedback chain and restore operation only after /joint_states updates normally again	Yes
Nodes still exist after shutdown	A launch terminal is still running, or zero-force drag is still active	Use ros2 node list | grep -E 'homogeneous_teleop|zero_force_drag' to confirm and close the	No

Symptom	Possible Cause	Handling	Immediate Shutdown Required
		corresponding terminal or service	

8. Related Paths

Item	Path
Homogeneous teleop package	/home/raybot/raybot_core_ws/src/raybot_homogeneous_teleop
Startup instructions source	/home/raybot/raybot_core_ws/src/raybot_homogeneous_teleop/doc/启动指令.md
Homogeneous teleop config	/home/raybot/raybot_core_ws/src/raybot_homogeneous_teleop/config/homogeneous_teleop.yaml
Homogeneous teleop controller config	/home/raybot/raybot_core_ws/src/raybot_homogeneous_teleop/config/ros2_controllers.yaml
RB-A630-2 zero-force drag launch	/home/raybot/raybot_core_ws/src/raybot_dynamics_control/launch/zero_force_drag_rb_a630_2.launch.py
RB-A630-2 zero-force drag config	/home/raybot/raybot_core_ws/src/raybot_dynamics_control/config/zero_force_drag_rb_a630_2.yaml