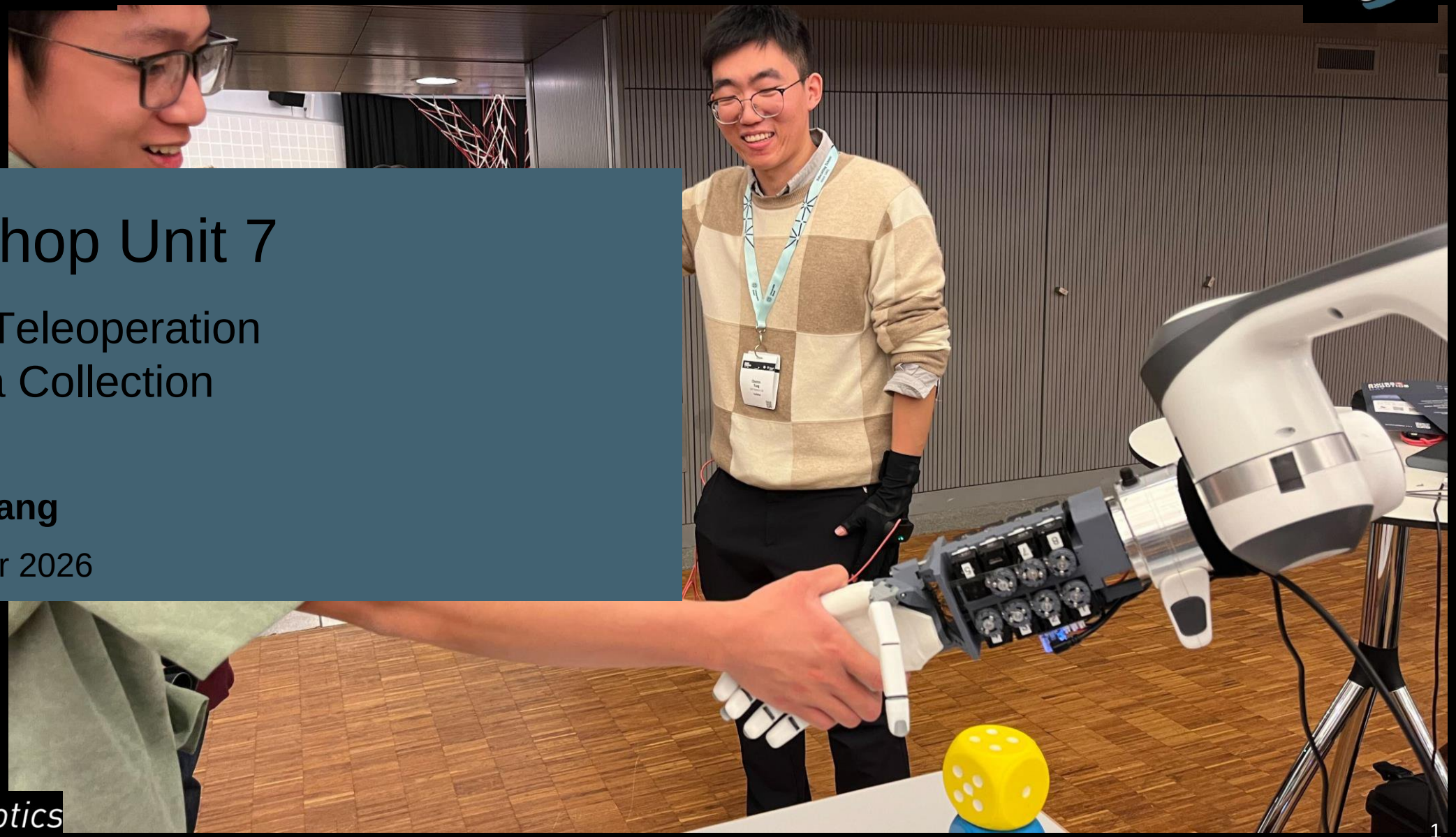
ETH zürich



Workshop Unit 7

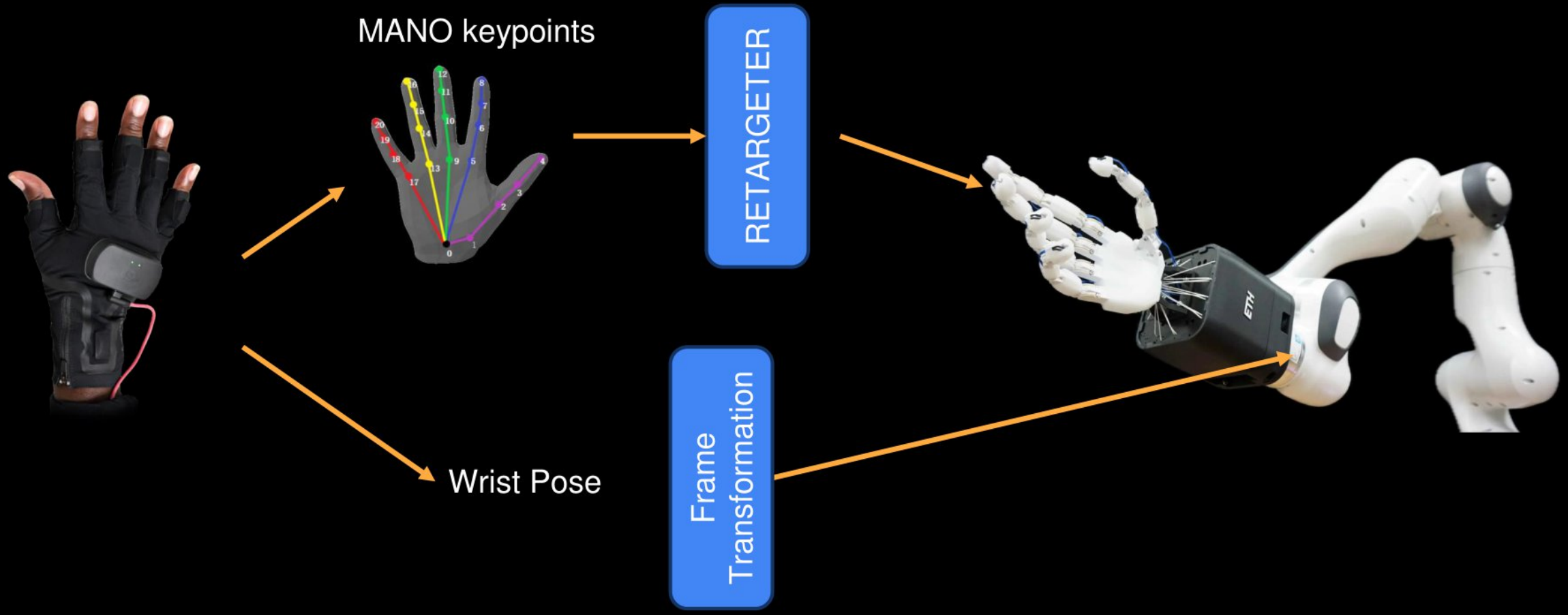
Franka, Teleoperation and Data Collection

Chenyu Yang

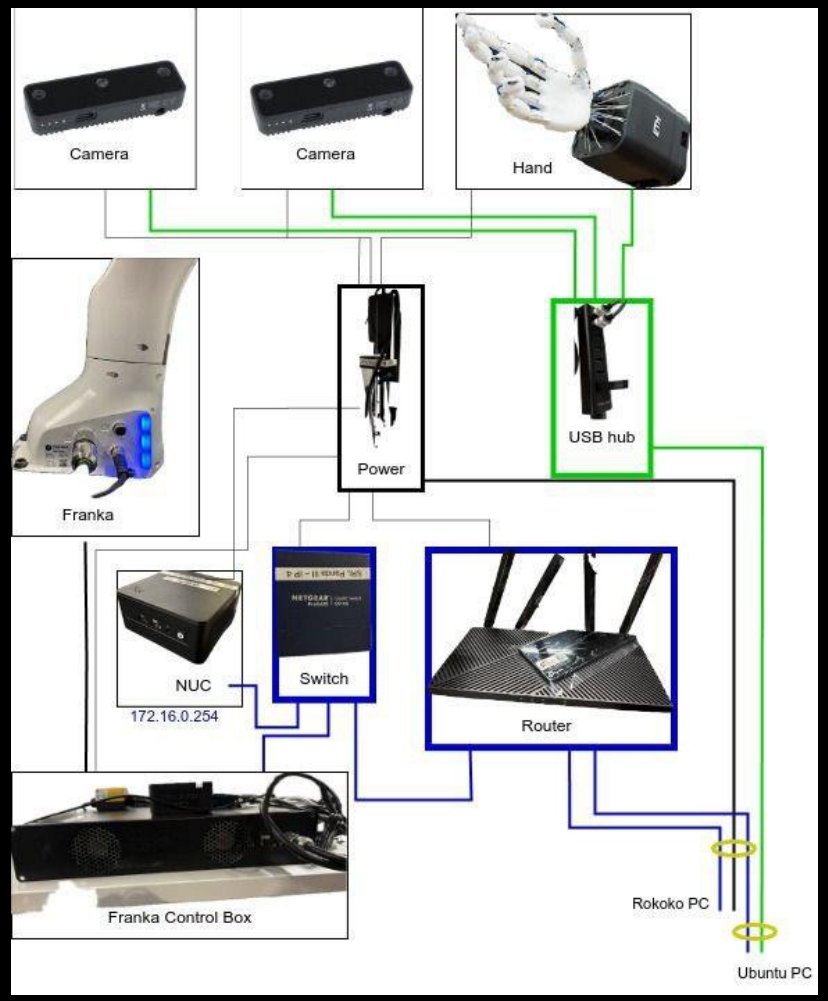
2 November 2026

SoftRobotics Laboratory

Teleoperation overview



Hardware of the RWR system



- Franka arm with your robot hand
- Rokoko glove and coil for hand and wrist tracking
- OAK-D cameras for observations
- Robot PC running multipanda_ros2

Use the IP address assigned to your station.

Black and red stop buttons

Black: user stop



- Press to stop robot motion.
- The arm stays powered and holds position.
- Clear the workspace before re-enabling.

Red: emergency stop

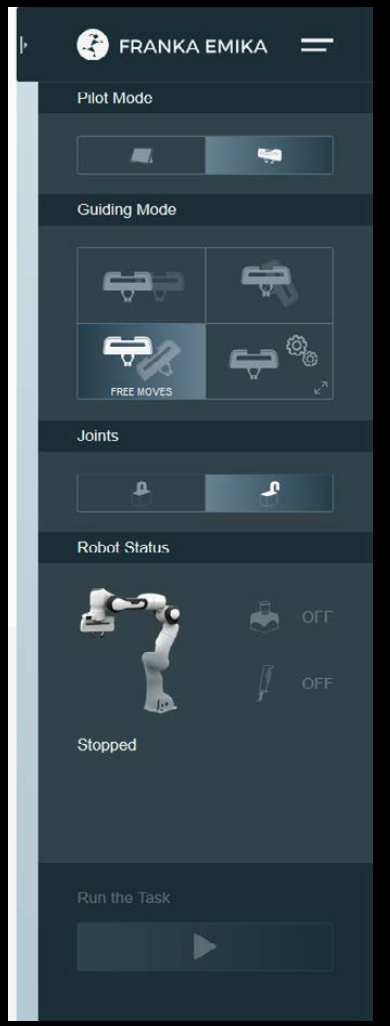


- Press immediately in an emergency.
- Power is cut and the joints lock.
- Ask the TA before restarting.

Franka Desk

1. Open https://<robot-ip> and log in.
2. Under Joints, unlock the brakes.
3. Clear the workspace, release the black button, then select Activate FCI from the menu.
4. Keep the “FCI is active” pop-up open.





ETH zürich SoftRobotics Laboratory

Controller Docker environment



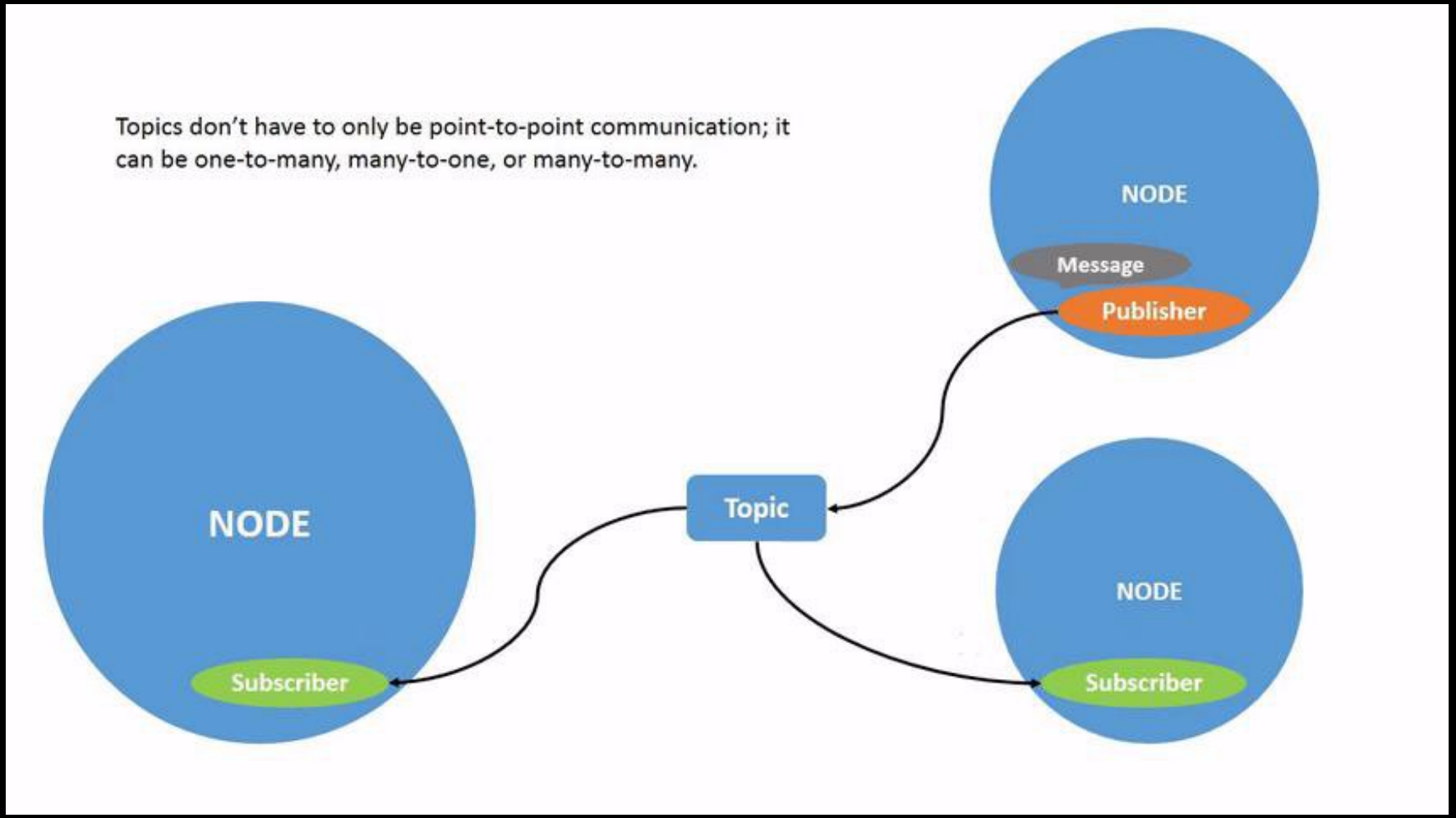
On the robot PC, open the multipanda_ros2 folder.

```
./run
```

This opens the prepared Docker environment.

github.com/srl-ethz/multipanda_ros2

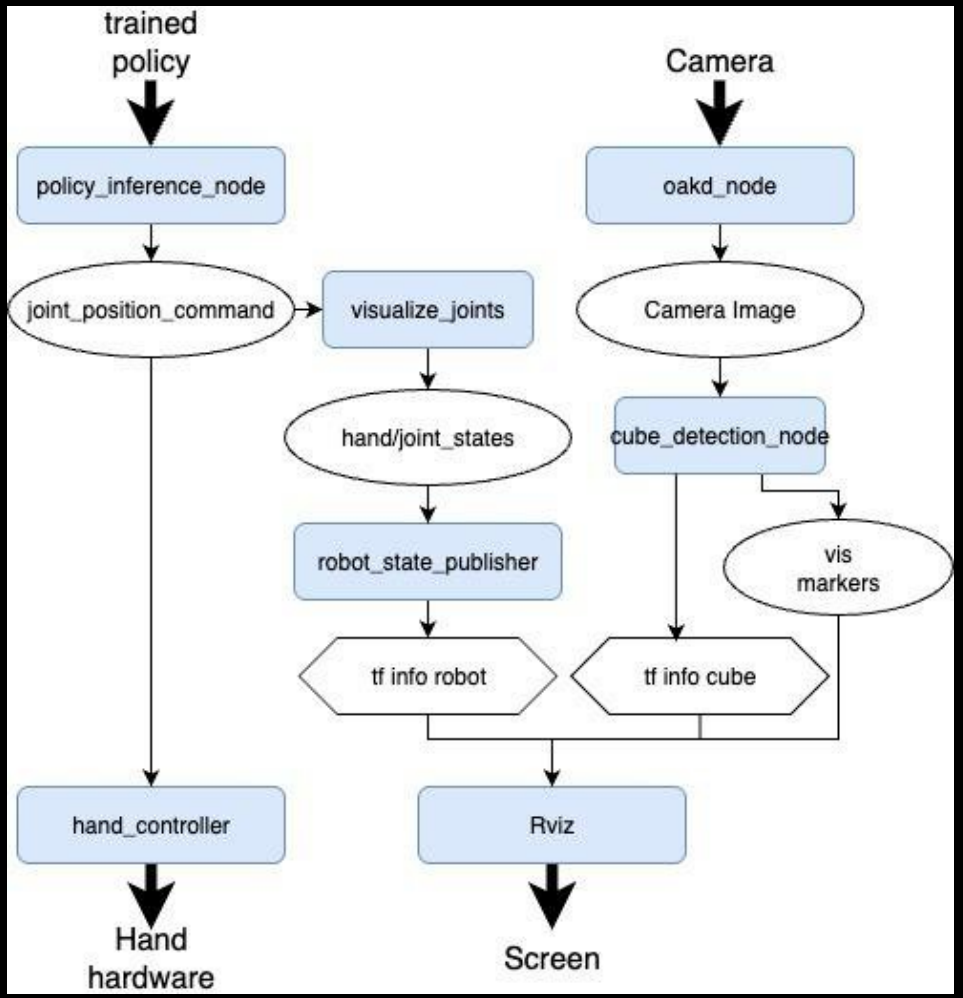
Teleoperation: ROS topics



Software of the RWR system: Task 1



- Policy commands the hand.
- Camera provides observations.
- RViz shows the robot and scene.



Software of the RWR system: teleoperation



Input	Processing and output
Hand keypoints	Retargeter produces robot-hand joint commands.
Wrist pose	Arm teleoperation produces an end-effector target.
Arm target	multipanda_ros2 controls the Franka arm.
Camera and robot state	Visualization shows the robot and scene.

The hand and arm follow separate command paths.

Demo 1: Rokoko glove and coil



- Connect the glove and coil.
- Check the live hand keypoints and wrist pose.
- Confirm the tracking responds to your motion.

Retargeting



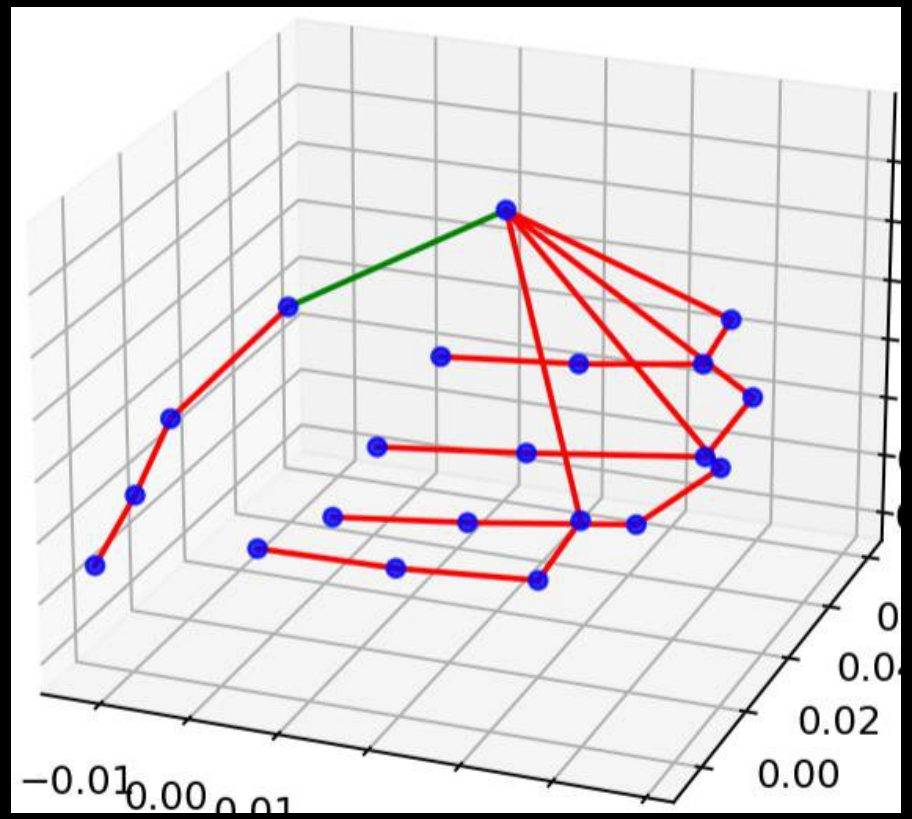
21 hand keypoints

Different geometry
Different joint limits



Robot-hand joint commands

Option 1: direct joint-angle mapping



Estimate angles from the keypoints and map them to robot joints.

Advantages

- Easy to implement
- Useful for testing joint motion

Limitations

- Different hand geometry gives different fingertip positions.

Option 2: inverse kinematics



Algorithm 1 Numerical Inverse Kinematics

$\mathbf{q} \leftarrow \mathbf{q}^0$ ▷ Start configuration

while $\|\boldsymbol{\chi}_e^* - \boldsymbol{\chi}_e(\mathbf{q})\| > tol$ **do** ▷ While the solution is not reached

  $\mathbf{J}_{eA} \leftarrow \mathbf{J}_{eA}(\mathbf{q}) = \frac{\partial \boldsymbol{\chi}_e}{\partial \mathbf{q}}(\mathbf{q})$ ▷ Evaluate Jacobian for $\mathbf{q}$

  $\mathbf{J}_{eA}^+ \leftarrow (\mathbf{J}_{eA})^+$ ▷ Calculate the pseudo inverse

  $\Delta\boldsymbol{\chi}_e \leftarrow \boldsymbol{\chi}_e^* - \boldsymbol{\chi}_e(\mathbf{q})$ ▷ Find the end-effector configuration error vector

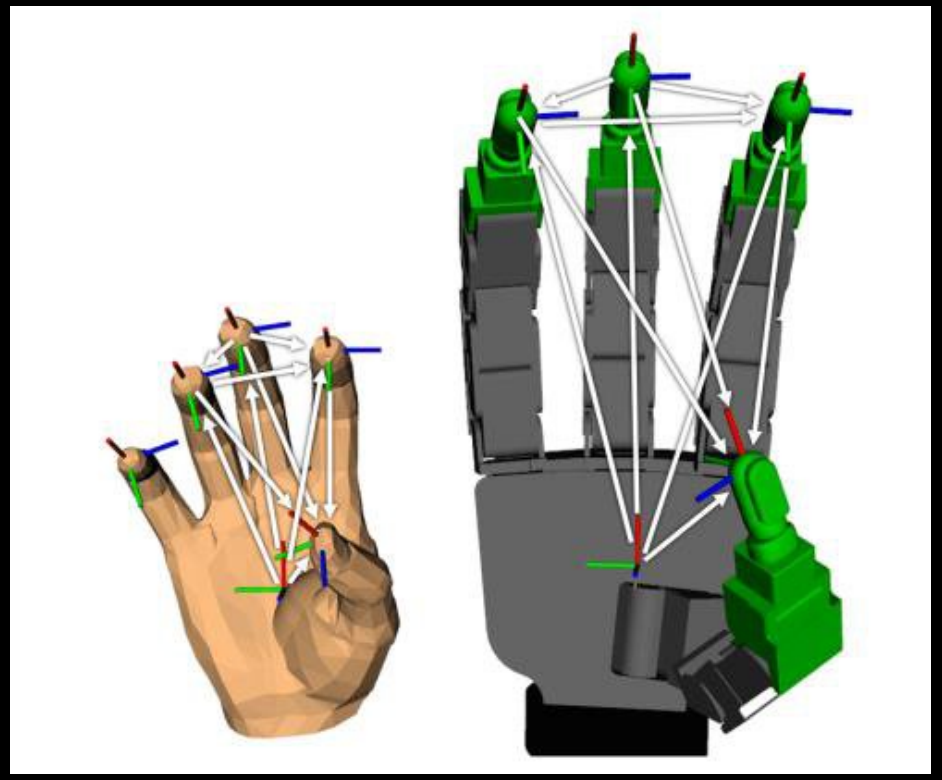
  $\mathbf{q} \leftarrow \mathbf{q} + \mathbf{J}_{eA}^+ \Delta\boldsymbol{\chi}_e$ ▷ Update the generalized coordinates

end while



- Track fingertip targets using the robot model.
- Update the joint angles with the Jacobian.
- Handle singularities and joint limits.

Option 3: keyvector optimization



Define Loss function

$$L(q) = \sum_{i=1}^{N} \left\| v_i^h - (c_i \cdot v_i^r) \right\|_2^2$$

Scaling coefficient

Robot joint angles

Hand keyvector

Robot keyvector

- Choose corresponding keyvectors.
- Minimize their mismatch over robot joints.
- Tune scale, limits and solver settings.

Demo 2: retargeting

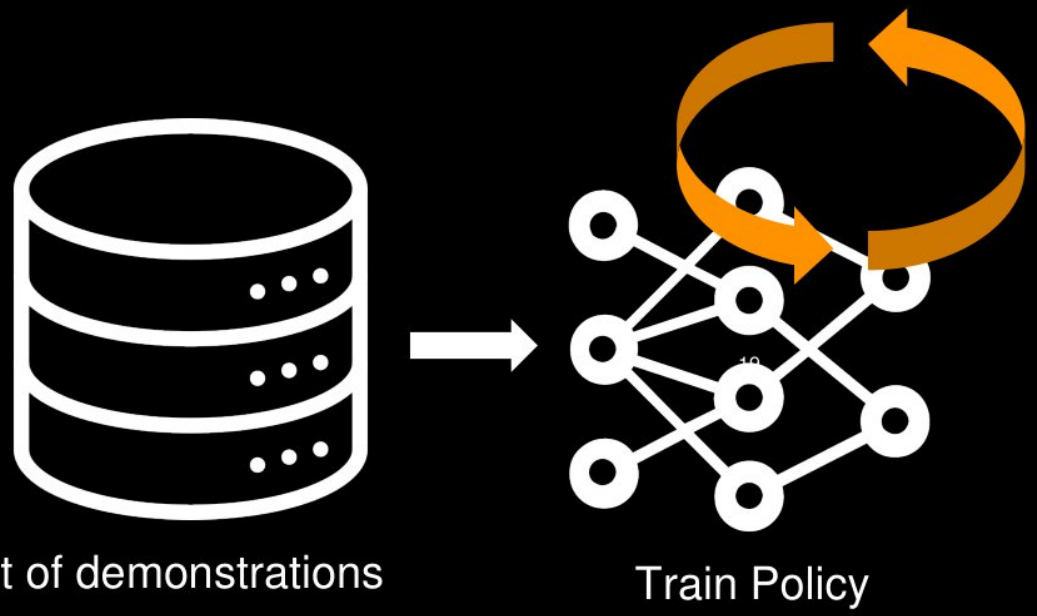


- Compare your hand with the robot visualization.
- Try finger flexion and a pinch.
- Check joint limits before enabling the hand.

Data collection for imitation learning



Learn from **expert demonstrations**



Deploy on the robot hand



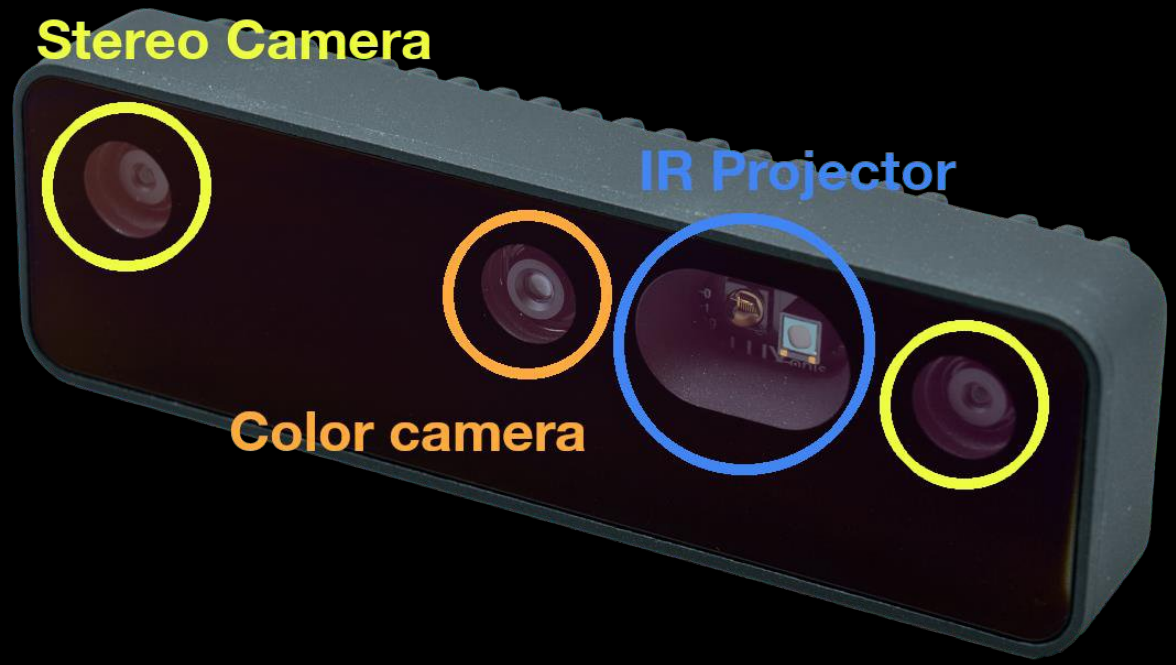
What do we record?



- RGB images from wrist and external cameras
- Measured end-effector pose and robot state
- Commanded arm and hand actions
- Timestamps and episode boundaries
- Task labels and any additional sensors used by the policy

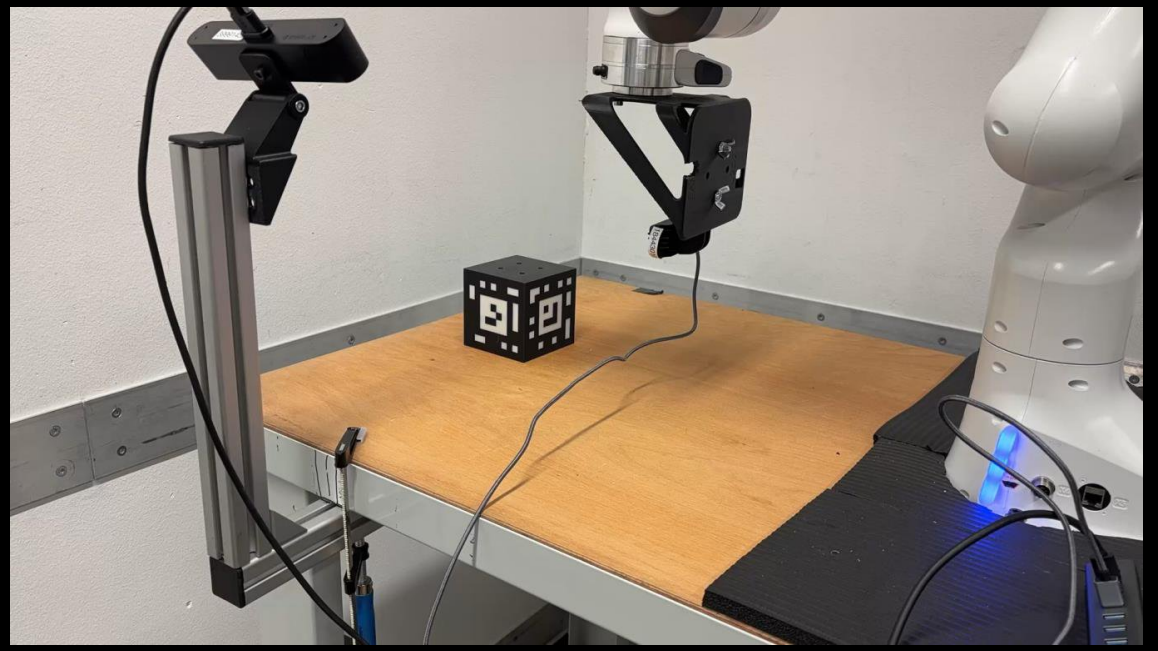
OAK-D cameras

RGB, depth and stereo observations



Camera calibration





Check the camera view and its alignment with the robot.

Software of the RWR system: recording



Stream	What the recorder saves
Cameras	Images and timestamps
Franka via multipanda_ros2	Measured end-effector pose and robot state
Teleoperation and retargeting	Commanded arm poses and hand joints
Episode controls	Start, stop and success/failure label

Review one saved demonstration before recording more.

Demo 3: data collection



- Start a new episode.
- Teleoperate a short task.
- Stop and label the episode.
- Replay the recording and check the data.

Software of the RWR system: policy inference



Stage	Role
Observe	Camera images and measured robot state
Infer	The learned policy predicts arm and hand actions.
Execute	multipanda_ros2 controls the arm; the hand controller drives the hand.
Monitor	Check the live scene and keep the stop buttons accessible.

Use one active command source at a time.

Workshop task overview



- Identify the black user stop and red emergency stop.
- Prepare Franka Desk and open Docker with ./run.
- Check glove tracking and hand retargeting.
- Record and inspect a short demonstration.

Good luck!

