

# Safety Guidelines

The combination of torque sensing in the robot's joints and impedance controllers allow for research into human robot collaboration and interaction. This combination of torque sensing and impedance control help with safety when a person is in the workspace of a robot but **does not guarantee it**.

Below, we outline some of the main safety guidelines and conditions for using the robots in the lab setting to help minimize risks while using the robots. Although this will never eliminate all risks.

**Please, always keep the safety of others and yourself a priority while using the robots in the lab.**

## Default Robot operating conditions

To use the robot in a collaborative way that allows safe human robot interaction, the robot **must** be run with the follow list of conditions as default. If you need to run the robot with conditions that vary from these, you will need to get permission from your daily supervisor.

- A cartesian impedance controller is used by default.
- Never use the robot alone, always have someone else in the room with you while operating the robot, incase of an accident or emergency.
- The joint velocity limit should be set to:
  - Start with a very low velocity/gains/forces/stiffnesses/range of movements etc., if that works safely and reliably, ramp it up iteratively until the final desired values (or restart at low values if it becomes unsafe/unreliable and you fix something).
- Stay away out the workspace of moving robots, only be in the workspace when necessary i.e. when performin human robot interaction research.
- Someone has their hand on the emergency stop button at all time that the robot is unlocked and live.
- Under no circumstances can you leave the robot operating in the lab without supervision.
- No controllers with integral gains e.g. position or velocity based controllers from **franka\_ros**.
- No MOVEIT!!... Due to the torus nature of the robots joint space, the majority of the path planning algorithms in MOVEIT, plan in the joint space, this can give very unintuitive paths for the robot joints to follow and cause it to whip around the long way. There are examples of point-to-point linear interpolation in the the human friendly controllers library, these should be the default trajectory planner. MSc students, talk to your Daily supervisor if this is not sufficient for the scope of your project.
- If possible, test your code and/or experiments in simulation first.

- Leave the robot and workspace clean and tidy after each use, so that next person doesn't have to clean up or disassemble the robot.
- Don't change the software configuration on the computers where the controller is running. Set up an isolated environment for your controller, as instructed by your daily supervisor. If this is necessary, ask permission from your daily supervisor and document and undo all configuration changes
- Where possible, only run the controller of the Franka Emika on the computer in the lab running the real time kernel. This is for practical reasons, as the load of other processes can cause the process to run slower not send the command to the robot in time, which will cause the robot to stop, along with respect to other people using the robot, as any global changes on that computer can cause other peoples code to break. The best practice here is to run everything other than the controller on your own computer, which is on the same subnet as the controller computer.

## General lab safety

- No loose cables going between the robots and areas other people can walk. If you need to run cables along the floor, ask the technicians for a cover to safely conceal the cables.
- No open drinks and food close to the robot and computers

## Acknowledgment and Signature

I acknowledge that I have read and understand the safety guidelines for using the robots and working conditions outlined in the lab in this document. I agree to follow these guidelines when working with the robots in the lab.

---

**Daily Supervisor**

---

Print Name

---

Date

---

**Student**

---

Print Name

---

Date