ABB arm setup

Author: XXC *Date: 2023/12/21*

The following are instructions for using the ABB arm to gather data with the NeuroTac sensor and CRI python library.

These instructions are written for Linux systems (Ubuntu 18 and above) with all software installed in a conda virtual environment named “ABB”.

Required software

If gathering data from the desktop PC, log in to the username “ABB” with the password “tactip123”. All software should be pre-installed.

If gathering data from a laptop, create a python virtual environment using conda called ABB for gathering data and install the following software:

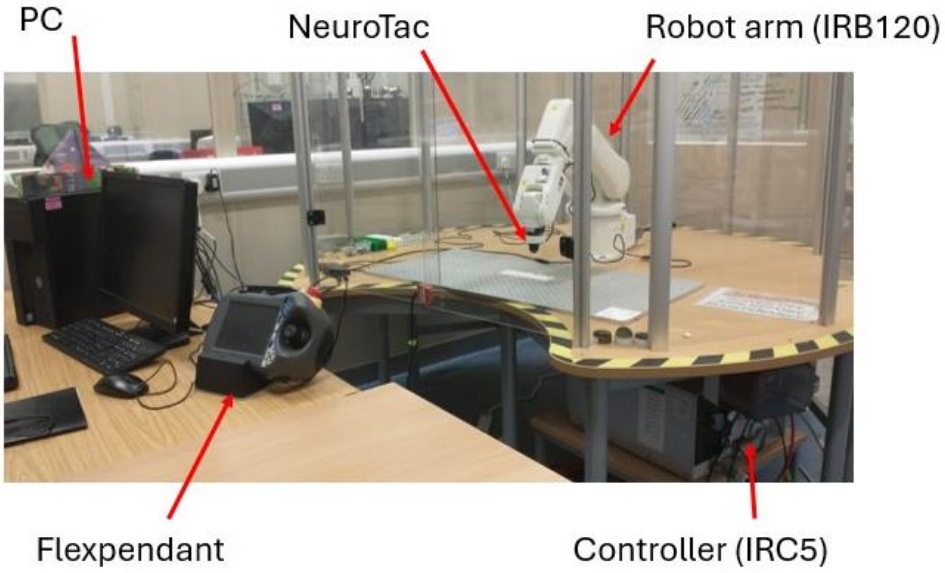
- CRI: Common robot interface for controlling Franka, UR, ABB and Dobot robot arms. Clone from https://github.com/jlloyd237/cri and follow installation instructions.

- Tactile-core-neuro: Python code to gather data with the NeuroTac sensor on the ABB, UR5 and Dobot robot arms. Clone from https://bitbucket.org/bw14452/tactile-core-neuro and follow installation instructions.

- dv-processing-python: backend code to stream event data from the NeuroTac sensor in python. Install according to instructions at Installation of dv-processing — dv-processing rel_1.6 documentation (inivation.com) (for Linux use “python -m pip install git+https://gitlab.com/inivation/dv/dv-processing”)

- DV: visualization software to quickly check output from the NeuroTac.Download and install from Get Started · DV (inivation.gitlab.io)

ABB robot components



Turning on the ABB robot

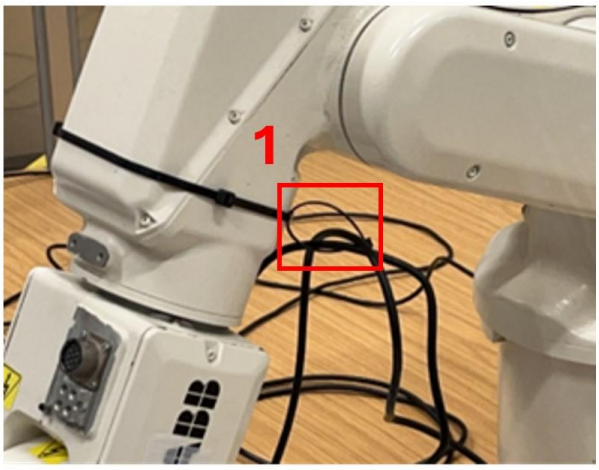
1. Connecting the NeuroTac sensor:

   - Plug the neuroTac's LED cable to the USB cable (black, USB 2.0) connected to the desktop PC.

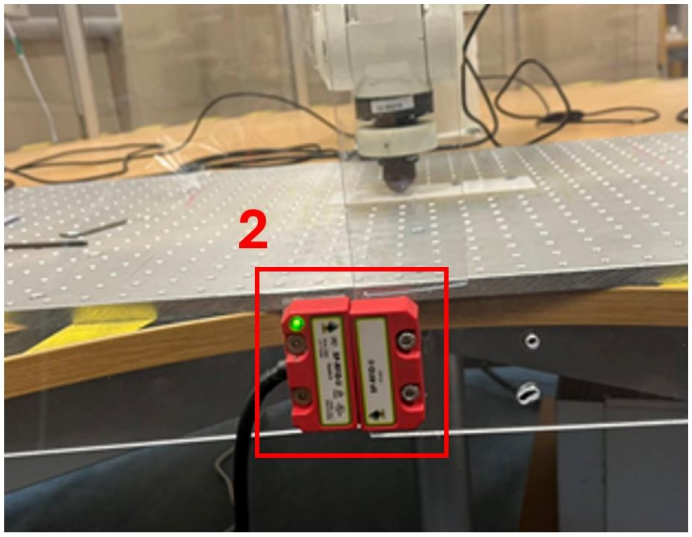
   - Plug the neuroTac's camera cable to the USB cable (blue, USB3.0) connected to the PC.

   Both USB cables should be fed through the black cable tie on the robot arm to avoid interfering with robot movement. If gathering data from a laptop, disconnect both USB cables from the desktop PC and connect

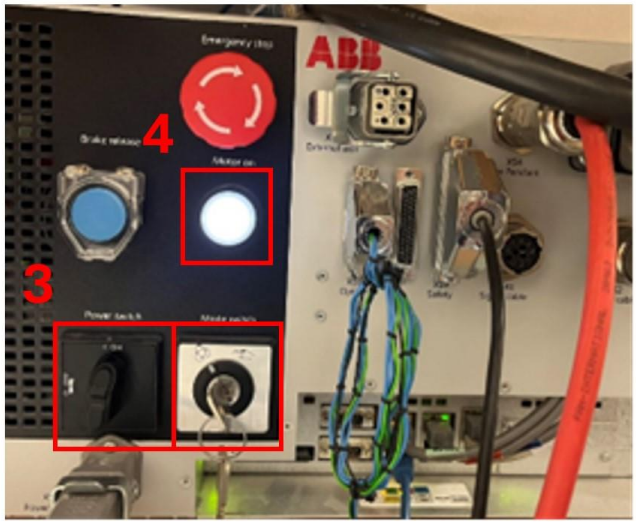
them to your laptop



2. Close the plexiglass windows in front of the robot and ensure the red sensors touch each other (green light on).

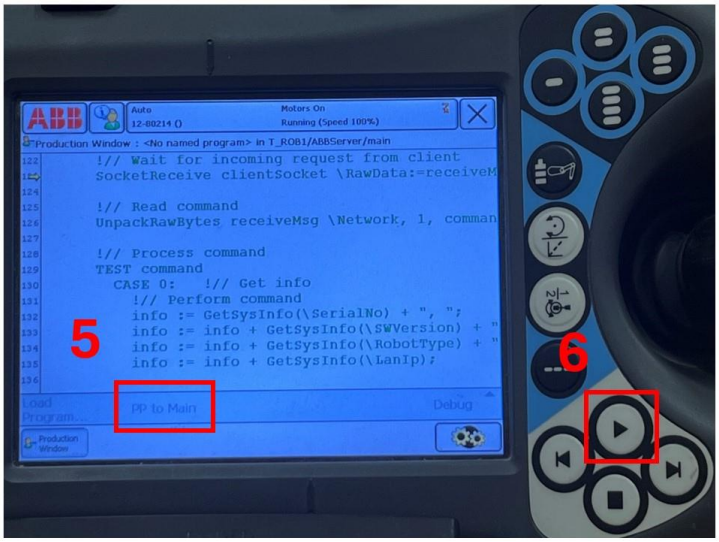


3. Switch on the ABB robot on the controller box under the desk (black on/off button). Check the key is set to “Automatic mode”.
4. The Flexpendant is the small screen attached to the control module by a thick red cable. Once this has booted up (after a minute or so), switch on the motors by pressing the white “motors” button on the controller. The button should go from flashing to constantly lit.



5. On the Flexpendant, the screen should look like the following image (if it doesn't, go to the "ABB" menu top left of the screen, and select "Production window"). Then select "PP to main".
6. Press the play button on the Flexpendant. You should see the message "T_ROB-1->SERVER: Waiting for incoming connections … " come up on the screen. The robot is now ready to connect to via the PC.

If connecting from your laptop rather than the desktop PC, disconnect the blue ethernet cable that goes from the robot controller to the back of the PC and connect it to your laptop.



Creating the experiment file

Create your own experiment file. You can copy an example file in the tactile-core-neuro repo at "python/experiments/NeuroTac_datasets" and change the

relevant parameters in the *make_meta* function (as described below). For example, if your experiment involves vertical tapping motions with the NeuroTac_DVXplorer sensor, use “tactile-core-neuro/python/experiments/NeuroTac_datasets/NeuroTac_DVXplorer/ABB_taps.py” as a base file.

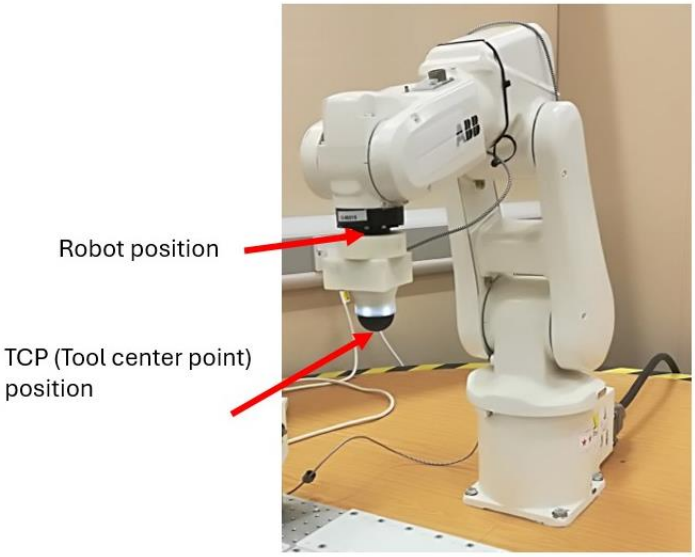
⚠ **Remember to create your own data file path and change the file path in your experiment file!**

Parameters in the *make_meta* function:

- *robot_tcp*: the relative position (x, y, z and 3 Euler angles) of the tool tip to the center of the flange of the robot (see image below). NeuroTac sensors will have this set a z value corresponding to the sensor’s overall height.

EXAMPLE The NeuroTac_DVXplorer with a round tip and the Robotiq force sensor on the back has an overall height of 120mm, so we would set the robot_tcp to “[0,0,120,0,0,0]”

⚠ **The robot_tcp value can be slightly off its real value, since all robot positions will be relative to the set TCP. However it is CRUCIAL that the robot_tcp parameter in the experiment file matches the value in the CRI GUI when setting the work_frame parameter (see next section)**



- *base_frame*: the origin world frame of the robot. Don't modify this parameter.
- *home_pose:* starting point of the robot (x, y, z and 3 Euler angles) when switched on. This parameter usually doesn't need to be modified.
- *work_frame:* starting point (x, y, z and 3 Euler angles) and reference coordinate frame for your experiment . This will need to be set for your specific experiment (see next section) and will usually be a position near the first object in your experiment.
- *linear_speed/angular_speed:* speed parameters. These usually don't need to be modified, unless your experiment has variations in speed.
- *tap_move:* sequence of positions for each "tap" or contact between the sensor and objects in your experiment. These positions are relative to the robot's current pose (positive z value goes down).

EXAMPLE If I want the robot to press 4 mm into an object then return to its current position, I can set its position to just above the object, and set *tap_move = [[0, 0, 4, 0, 0, 0], [0, 0, 0, 0, 0, 0]]*.

- *obj_poses:* positions (x, y, z and 3 Euler angles) of the different poses/objects relative to the work frame coordinate system, i.e. [0,0,0,0,0,0] corresponds to the position set in *work_frame*.

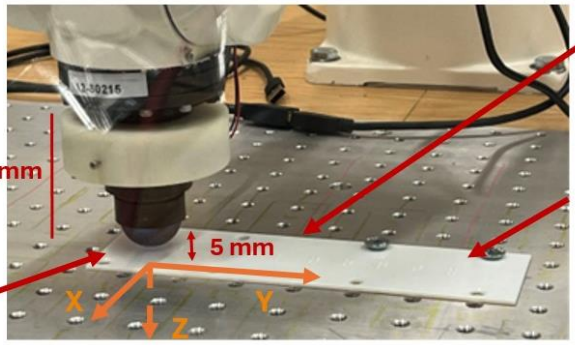
  EXAMPLE If 5 objects are listed in a line along the table's Y axis with gaps of 10 mm between them, we would set this parameter to *obj_poses = [[0, 0, y, 0, 0, 0] for y in range(0, 41, 10)]*.

- *n_trials*: total number of trials for the experiment (i.e. number of times each "tap" is repeated on each "object").

Example experiment:

1. The sensor used here measures 62mm from robot arm tip to sensor tip, so *robot_tcp = [0,0,62,0,0,0]*

2. The starting point of the experiment is above the first object (Braille letter). Here, *work_frame = [450,-61,19,180,0,180]*



3. The interaction with each object is a 5 mm downward tap in the z direction so *tap_move = [[0,0,5,0,0,0][0,0,0,0,0,0]]*

4. This experiment features 11 objects (Braille letters), with 10mm spacing in the y direction. So *obj_poses = [[0, 0, y, 0, 0, 0] for y in range(0, 101, 10)]*

Setting the starting position for an experiment

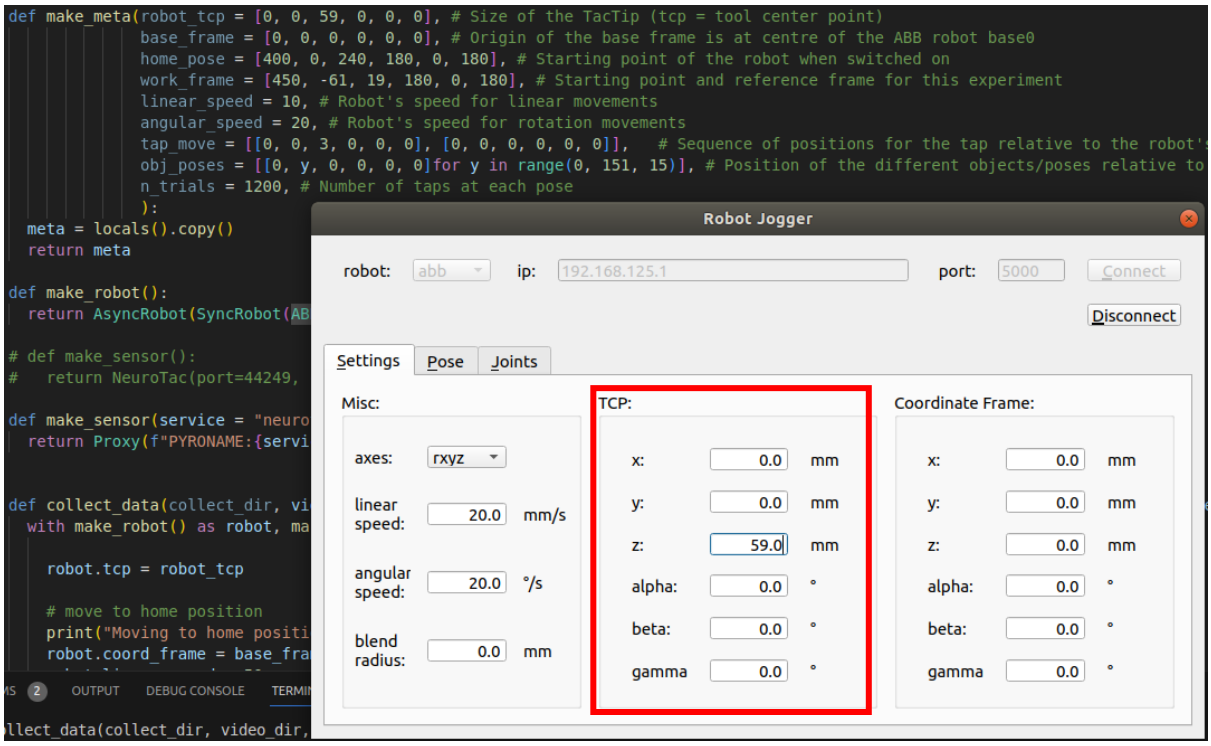
1. Run the CRI robot jogger. If gathering data using Ben's username on the desktop PC, the file path is: "/home/ben/repos/cri/tools/robot_jogger/robot_jogger.py". Otherwise, it will be within your downloaded CRI repo at: "cri/tools/robot_jogger/robot_jogger.py"

2. Fill in the following in the GUI that pops up:
   - Ip address: 192.168.125.1
   - Port: 5000

   Click "connect". This should connect you to the ABB, and you should see the following message on the Flexpendant: "T_ROB-1->SERVER: Connected to IP address 192.168.125.2"

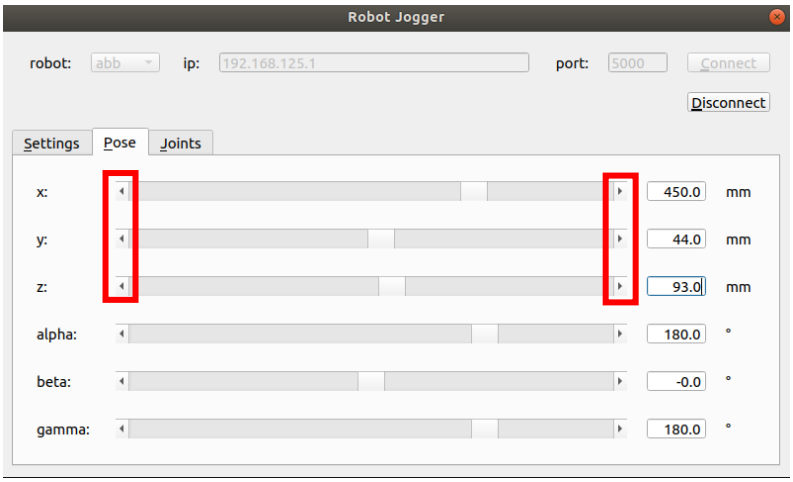
3. Check the TCP parameters in the Settings tab, middle column. Make sure they match your experiment file values before proceeding.

⚠ **TCP parameters MUST match the *robot_tcp* in your experiment file.**



- On the pose tab, adjust the position by clicking on either side of the x,y and z scroll bars to move 10mm in either direction, or on the arrows at the end of the scroll bar to move 1mm in each direction. Bring the sensor tip to the starting location of your experiment (close to the first object in your experiment).

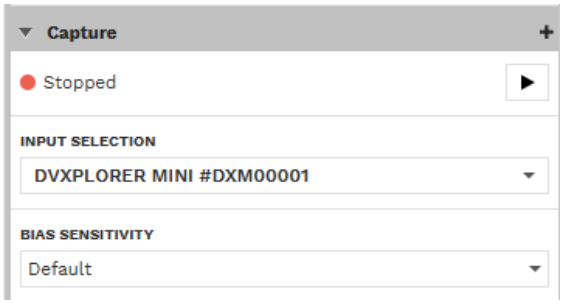
⚠ **Only move the robot with the scroll bar and arrow clicks! Do not drag the scroll bar or enter numbers in the boxes at the end of each scroll bar as this could cause the robot to collide with the table.**



- Once you have reached the starting position of your experiment, copy the current values in x,y,z,alpha,beta and gamma to the

work_frame parameter in your experiment file. In the above example, you would set *work_frame = [450,44,93,180,0,180]*

- To confirm the precise height of the tool tip in the z direction you can use the DV software. You should find "Capture" on the right side. Select your camera type and you can view your events on the screen. Adjust the tool tip by clicking the arrow buttons in the downward z direction until you see a burst of events indicating contact, then set your *work_frame* starting point relative to this point of initial contact.



- Close DV. You should now have your experiment file ready!

Test runs:

1. As an additional precaution, add 100 to the z value of the *work_frame* parameter in your experiment file so the robot is not in contact with objects for the first test run. Set *n_trials* to 2, as we just want to test that the robot is moving as expected here.
2. Ensure the robot_jogger GUI is disconnected (click the disconnect button) and the DV software is shut down.
3. Ensure the ABB virtual environment is activated. (If not already active, in the terminal type > conda activate ABB)
4. Activate the pyro server on the Desktop PC (this will stream events from the NeuroTac sensor). To do so, open a terminal and type:

   > cd ~/repos/tactile-core-neuro/python/core/sensor

   > python tactile_sensor_neuro_pyro_server.py

5. Wait a few seconds until "Service neurotac_service_1 running " is displayed in the terminal. The sensor is now ready to go.
6. Run your experiment file. Hold your hand on the E-Stop button of the Flexpendant in case anything unexpected happens.

7. If everything goes as expected, correct the *work_frame* parameters by subtracting the 100 that was added in (1) and rerun your experiment. Check that any required data, meta information and videos were saved for all trials and are in the correct format.

Actual experiment run:

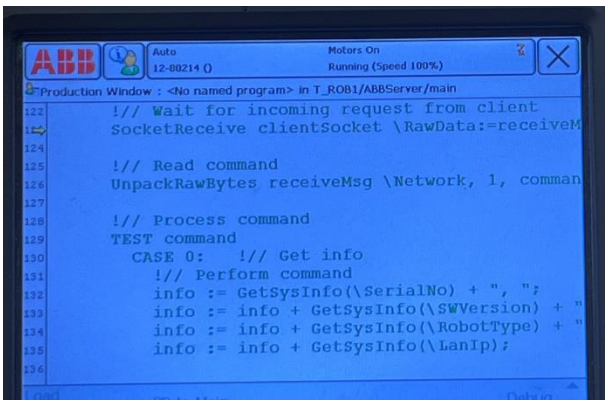
You are now ready to run your experiment!

Ensure the *robot_tcp*, *work_frame* (especially the Z position value) and *n_trials* parameters are set to the correct values and run your experiment python file.

Keep your hand on the E-stop button of the Flexpendant for at least the first few trials.

Turning off and resetting the robot:

- In the case of an emergency stop (you pressed the E-stop button), the robot needs to be reset by releasing the E-stop button (twist it clockwise) and restarting the motors (white button on the controller).
- To turn off the robot,press the red “ABB” button on the top left on the Flexpendant. Press the red “Restart” button on the bottom right. Select “Advanced” on the bottom left and “Shutdown”. Then switch off the robot by turning the black button on the controller to the OFF position.



FAQs

1. *Why can't I start the robot by pushing the white motors button?*
    - Check whether the glass windows are closed, and the red sensors are in contact with each other (green light on).
    - Check whether the E-Stop button is pressed. Release it by turning it clockwise if so.

2. *Why can't I connect to the robot with the CRI GUI?*

- Check that the robot controller is connected to the desktop PC or your laptop (blue ethernet cable)
- Check whether you have pushed the white motors button.
- Check that the IP address and port number are correct in the GUI.

3. *Why can't I start the pyro server or run the robot?*

   - Check whether you have disconnected the GUI and shut down the DV software (you cannot connect simultaneously to the robot or sensor with several different pieces of software)

4. *How can I determine whether the tool tip almost touch the material to set the best coordinates for work_frame?*

   - You can simply place a piece of paper between the material and tool tip. When you just feel there is a little bit of friction when pulling the paper, it is place you want.

   - You can also use DV software to determine it. When the screen shows events from none, the tool tip touches the material.