Setup Franka Simulation

1 Introduction

The simulation scenario for exercises for the rest of the course will no longer involve AUVs, but robotic manipulators. In the Teams channel *AY2526 - Cooperative Robotics* you have a folder called *Exercises Manipulators* in which you can find this guide together with:

- The Python simulator source file inside the folder `python_simulator`.
- The datasheet of the robot.
- The guide for the setup of the virtual machine (if needed).

The robot adopted for this assignment is the Franka by Emika, a versatile 7-DOF manipulator for researchers (see Figure 1 and datasheet). The Python simulator will be used to visualize two manipulators and perform some cooperative manipulation scenarios, by implementing the *Task Priority Inverse Kinematic* algorithm in Matlab.

2 System Requirements and Installation

To complete the exercises, you will need the following tools and software:

2.1 Matlab Requirements

- Matlab version > 2019a with the following toolboxes installed:
  - Robotic System Toolbox
  - DSP System Toolbox

2.2 Python and Ubuntu Requirements

- Ubuntu version > 18.04 (the simulator may work on older distributions, but additional Python packages might be required):
  - Ensure `python3` is installed by running:

    ```
    python3 --version
    ```

  - Install the `pyBullet` package:

    ```
    pip install pybullet
    ```

2.3 Using the Simulator

Run the following command to launch the simulator:

`python3 franka_panda_simulation.py`

This will display a scene with two Franka Panda robotic arms (Left and Right Arm, see Figure 2). Use the following mouse controls for navigation:

- Rotate: `ctrl+left_button` + drag
- Move: `ctrl+Alt+left_button` + drag

2.4 Using a Virtual Machine

If you do not have Ubuntu installed, consider using a virtual machine. Matlab can run on Windows, as data is sent to the simulation using UDP communication. For an easy configuration follows the following steps:

- Read the guide `setup_virtual box_for_UDP_communication.pdf`
- Test the UDP connection between Matlab and Python simulator by running the following code on Matlab.

```
hudps = dsp.UDPSender('RemoteIPPort',1500);
hudps.RemoteIPAddress = '127.0.0.1';
q = [0,0,0,0,0,0,0]';

for t=1:0.1:30
    disp(t);
    step(hudps,[q;q]);
    pause(1);
end
```

The robots should assume the configuration depicted in Figure 1.

2.5 Using WSL2 (not recommended)

PORT FORWARDING with WSL2
https://superuser.com/questions/1789240/wsl2-port-forwarding-between-windows-and-linux

1. netsh interface portproxy add v4tov4 listenport=1505 listenaddress=¡windows ip¿ connectport=1505 connectaddress=((wsl hostname -I).Trim())
2. New-NetFirewallRule -DisplayName "WSL2 Port Bridge" -Direction Inbound -Action Allow -Protocol TCP -LocalPort 1505
3. PUT THE VIRTUAL MACHINE ADDRESS -¿ ((wsl hostname -I).Trim()) both on Matlab and Python !!!
4. Remember to PUT THE virtual machine address on Virtual Box !!!!

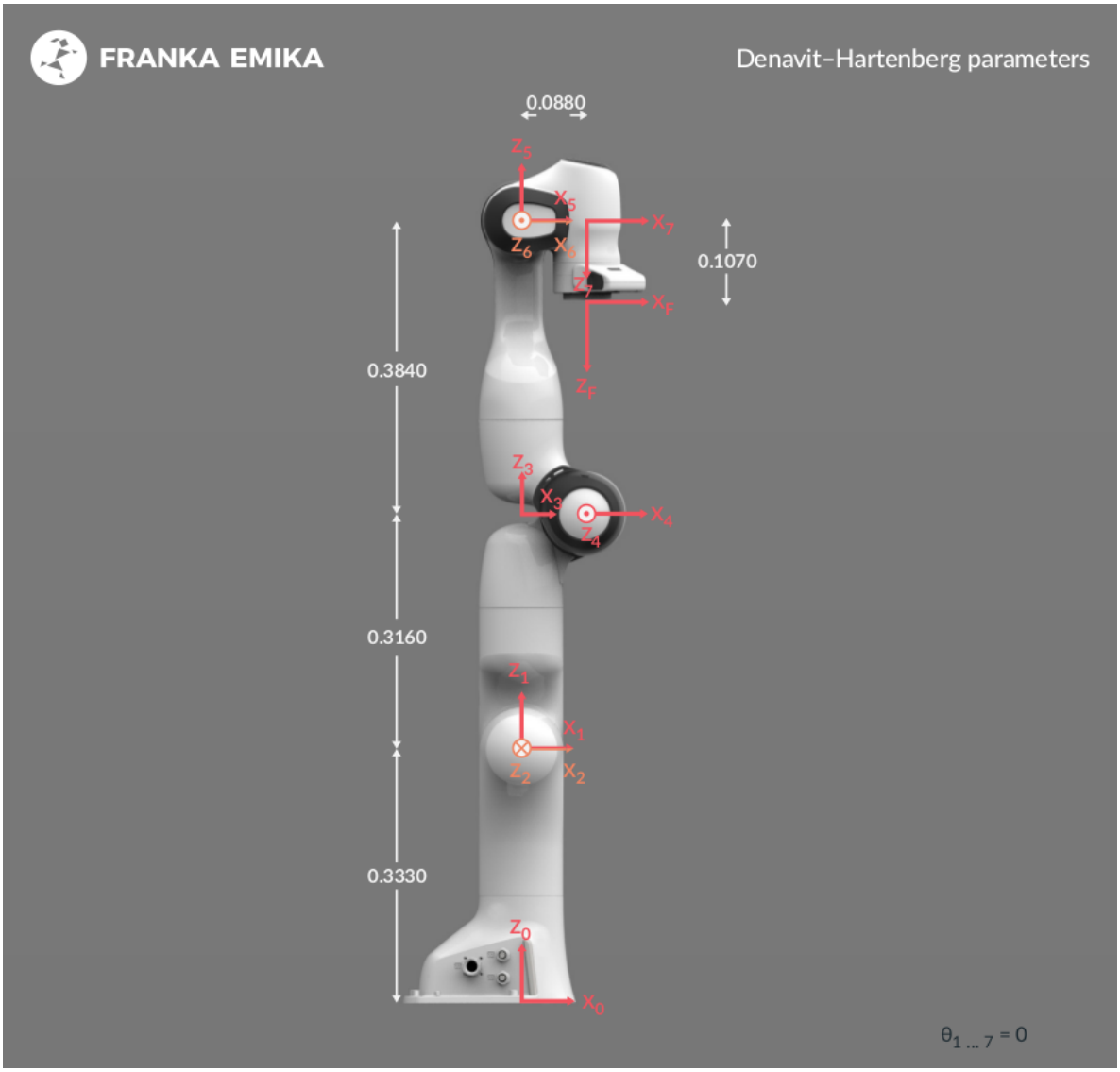

Figure 1: Robot Specifications

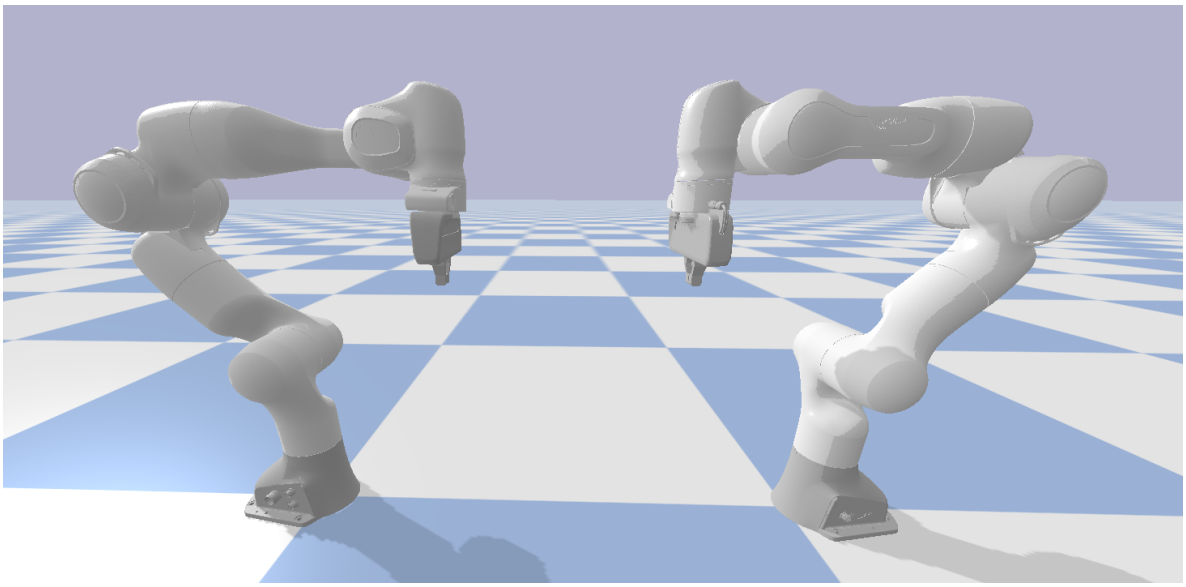

Figure 2: Left and Right robotic arms