GitLab link: https://gitlab.lrz.de/tum-impl-ss24/assignment_3_group_2

# Report for Assignment 3 IMPL

**Group 2: Jiabao Song, Adnan Fidan, Rui Tu, Dorian Fülöp**

In the second task, a copy of the provided cartesian_impedance_control.cpp (record_trajectory.cpp) in the LibFranka examples folder was made. The stiffness and damping was set to 0 to simulate free-drive functionality. In the hand guiding mode an infinity symbol was demonstrated by physically moving the robot through the motions and the joint angles and Cartesian end-effector coordinates were recorded in text files.

For the third task, a copy of the provided generate_cartesian_pose_motion.cpp (infinity_draw.cpp) was made. The file was setup to draw a 90 degree arc and extended to draw two consecutive circles to form an infinity sign.

For the fourth task, initially, a copy of the file generate_cartesian_pose_motion.cpp was made (replay_trajectory.cpp), and the cartesian coordiantes were returned to the CartesianPose callback. However, due to the strictness of the accepted input to the struct CartesianPose, inverse kinematics where not able to be generated, which resulted in reflex mode errors. So, instead, a copy of the provided joint_impedance_control.cpp example file was made. The joint angles were imported and returned to the torque callback struct. The actual end-effector coordinates were saved again for comparison.

Below, are the actual and desired coordinates in 3D space shown in Figure 1 and 2, respectively:

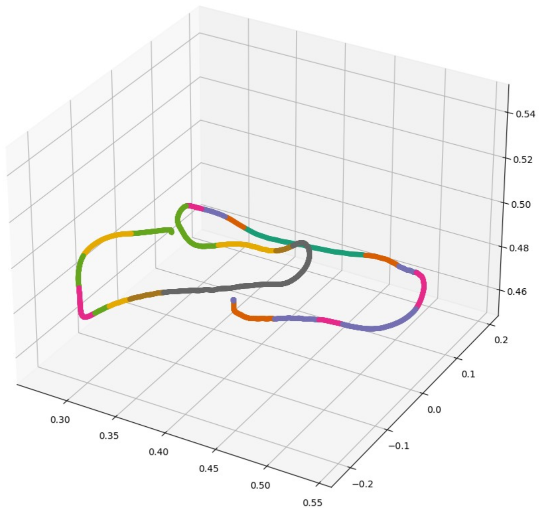

*Figure 1: Desired Trajectory*

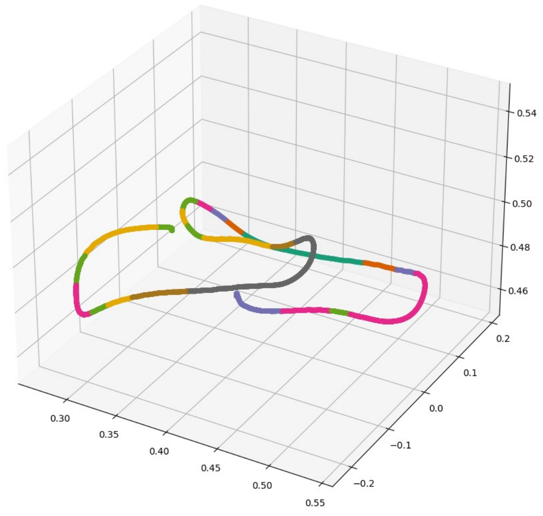

*Figure 2: Actual Trajectory*

The error in each axis was calculated and plotted below:

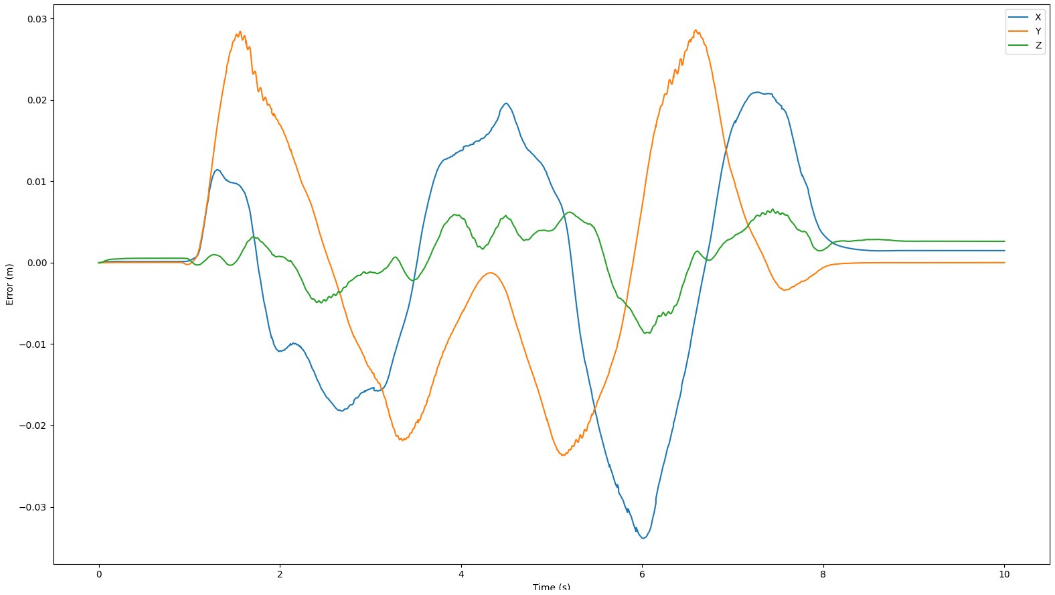

*Figure 3: (x,y,z) Error Over Time*

For this implementation, robot positions were sampled at a rate of 1000Hz. So the error should not be from a deviation in number of trajectory points. When replaying the trajectory motion, the stiffness and damping parameters that are set affect the trajectory the robot will take. This will lead to error between the desired and actual trajectories. Furthermore, as shown in figure 3, initially a small error is already visible before any robot motion. This indicates the affect of sensor errors. After the motion has ended we can see that the error increases. This indicates that there is some accumulated error from moving in respective axes.