Assignment 01: Comparison between different controllers

Jianming Han [230401], Basem Shaker [230637]

October 19, 2021

Question 1

Simulation results are shown in Figure 1. The tracking errors are listed here:

Test #	Controller	Kp	Frequency	Average Track Error [m]
1	OSC	50	[1.0,1.0,0.3]	0.031
2	IC	50	[1.0,1.0,0.3]	0.036
3	IKID	50	[1.0,1.0,0.3]	0.031
4	OSC	100	[1.0,1.0,0.3]	0.02
5	IC	100	[1.0,1.0,0.3]	0.024
6	IKID	100	[1.0,1.0,0.3]	0.02
7	OSC	50	[2.0,2.0,0.6]	0.064
8	IC	50	[2.0,2.0,0.6]	0.058
9	IKID	50	[2.0,2.0,0.6]	0.065
10	OSC	100	[2.0,2.0,0.6]	0.032
11	IC	100	[2.0,2.0,0.6]	0.044
12	IKID	100	[2.0,2.0,0.6]	0.034

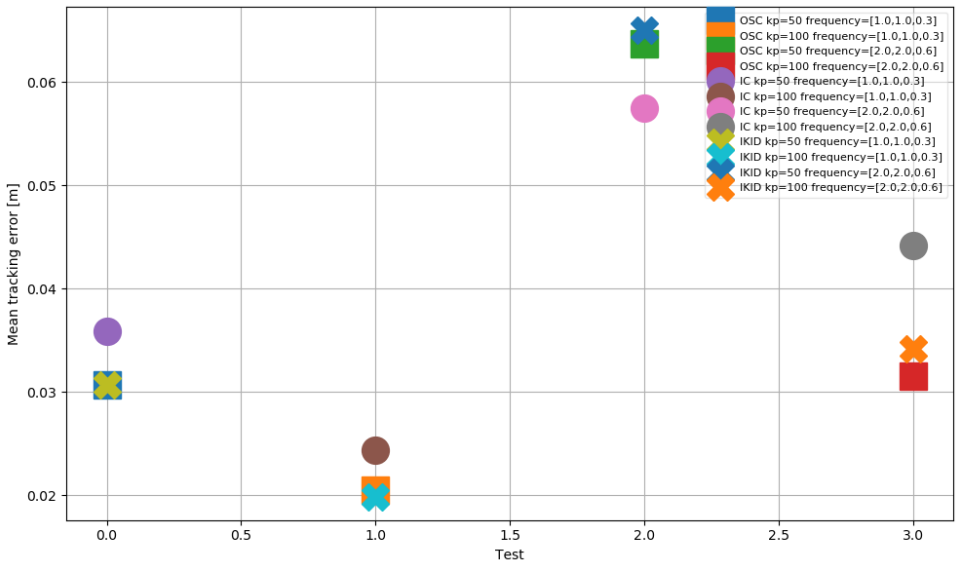

Figure 1: Mean tracking errors

OSC and IKID have similar performance in the 4 settings,. IC performs the worst in 3 out the 4 settings.

In the two higher frequency settings, IC performs best when $k_p = 50$, OSC performs best when $k_p = 100$.

In the lower frequency settings, controllers have smaller mean tracking errors, while in the higher frequency settings, the mean tracking errors increase a lot for all the controllers.

Question 2

We ran the simulations four times, and the results are shown in Figure 2.

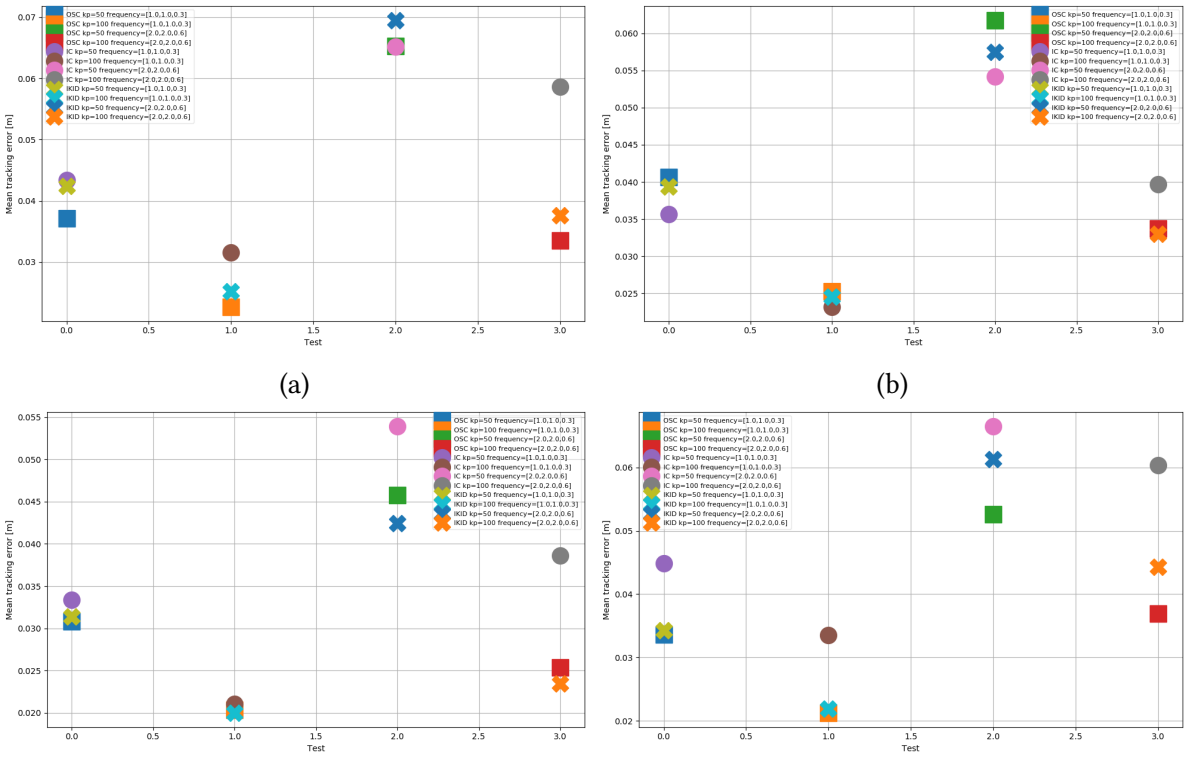

Figure 2: Different runs with randomization on.

We averaged the results of our four runs and compared it to the first results achieved in Question 1. Figure 3 shows the difference in the error between the average result from our random four runs and the result without randomization.

The figure shows that the tracking error has consistently increased across all controllers in the lower frequency. However, the tracking error became more erratic with the higher frequency. For IC, the mean tracking errors increased across both frequencies. We added all the errors together as shown in Figure 4. Based on the results, OSC is the most robust to the introduced modeling errors.

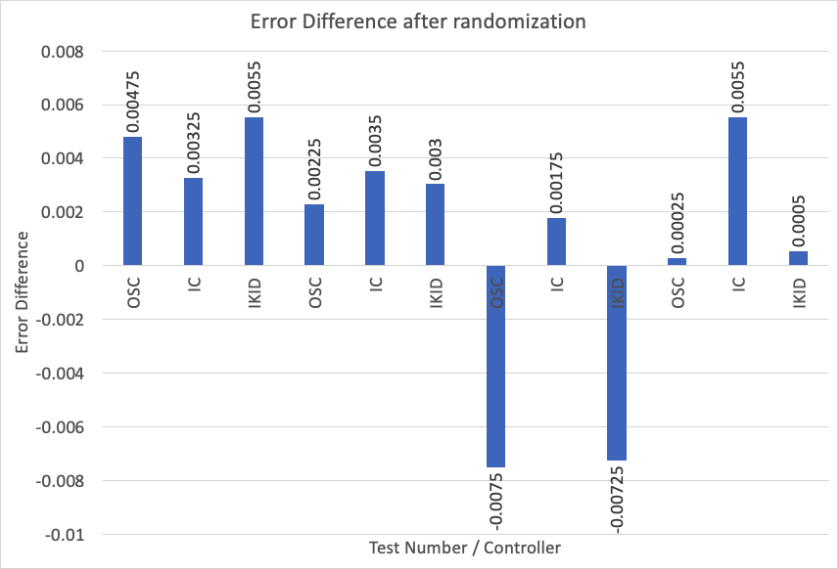

Figure 3: Difference in error after randomization

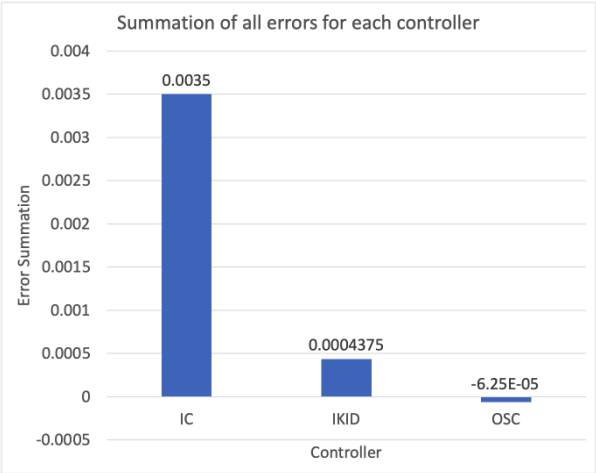

Figure 4: Summation of Errors for each controller

Question 3

The robot makes some abrupt moves because the control signal to stabilize a particular joint configuration is not enough to make it stable all the time. In order to avoid such abrupt moves, we can tune the proportional gains of the joint space, which is kp_j, to make it more stable in the joint space.

Question 4

In our experiments, when we increased kp_j from 20 to 50, the robot was stable in the joint space for all the 12 test settings. Increasing it to 100 did not improve any further, while decreasing it to 10 gave way worse results.

Figure 5 shows the percent improvement in the mean tracking error after increasing kp_j from 20 to 50. The most controller that benefited the most was the IKID controller. In general, the

mean tracking errors don't change a lot because tuning kp_j mainly affects the torque in the joint space, it doesn't give much impact in operational space.

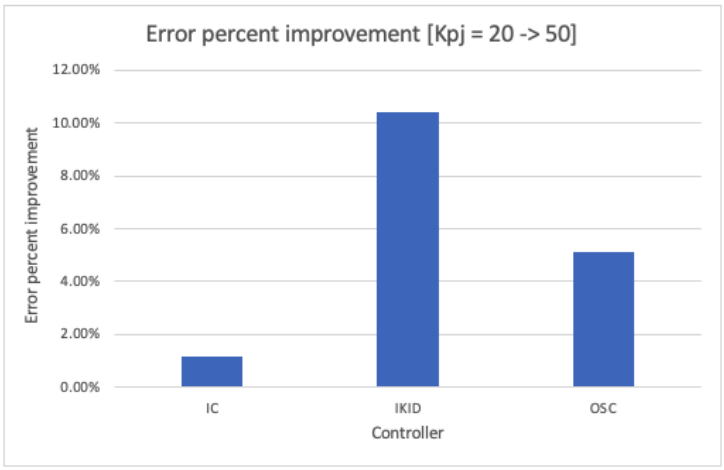

Figure 5: Mean tracking error percent improvement

Question 5

We will implement OSC control law in a real robot, because it's the most robust one according to our experiments. One way to improve the performance is to increase kp_j because it will make the robot more stable, but it won't decrease the tracking errors a lot. The other way to improve the performance is to increase kp which will help decrease the tracking errors. Figure 6 shows the results of the comparison between two different kp values 50 and 1000. We can see that the mean tracking errors decrease a lot.

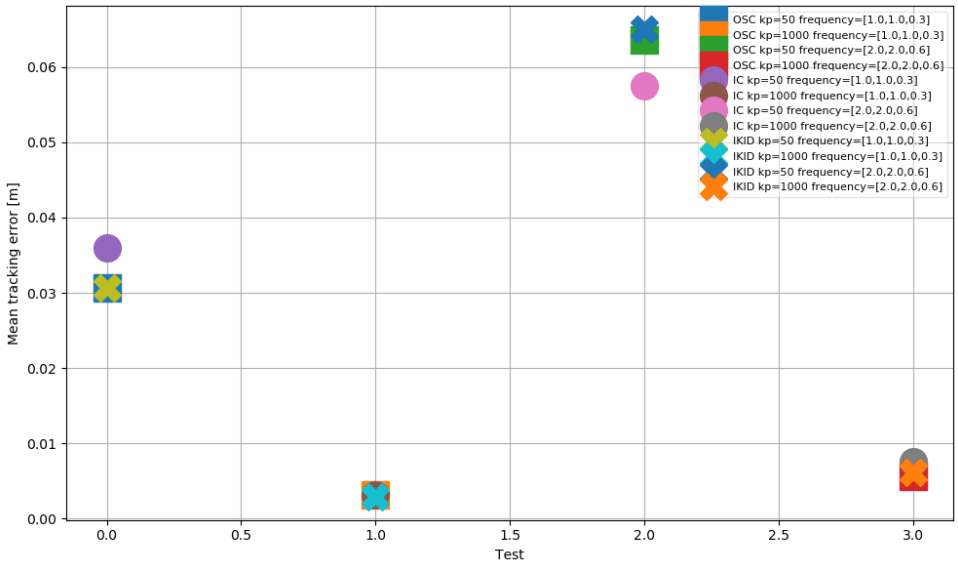

Figure 6: Performance comparison by increasing kp.