Homework 1 Report

La Scala Giovanni Maria Francesco 241561
Di Lorenzo Andrea 239221

1 Goal of the assignment

The goal of the assignment was to implement different types of controllers for robotics learned during lessons.

We want to compare in terms of performances under different conditions the followings:

- **Operational Space Controller**
- **Impedance Controller**

For the Operational Space Control (OSC) we are interested in testing the control applied to a postural task, which means we want to stabilize the position of the end-effector at a given point and then to a reference trajectory.

As for the Impedance Control (IC) the assignment asks us to compare different formulations (simplified and complete versions) and apply them to a postural task which will be, again, the stabilization at a given point for the end-effector and after we want to stabilize the end-effector around a given trajectory.

2 Stabilize desired position with different versions of IC

With impedance control we want to regulate the relationship between motion and force by means of the following control law:

$$\tau = J^T \left(Ke + B\dot{e} + \mu \right) \tag{1}$$

Where e is the error between the desired position and the actual one, $\dot{e}$ the error in velocity, the matrices K and B contain **stiffness** and **damping** parameters of the robot and are user-defined.

If we want to add the postural task in this control law we need to minimize the null space of $J^{T\dagger}$ and therefore the control law becomes:

$$\tau = J^T \left(Ke + B\dot{e} + \mu \right) + \left(I - J^T J^{T\dagger} \right) \tau_0 \tag{2}$$

Where τ_0 is the value of the torque which doesn't affect $\ddot{x}$.

If we want to further simplify the control law in (1), we can neglect the biased forces (due to gravity, Coriolis and centrifugal forces) therefore avoiding to compute μ:

$$\tau = h + J^T \left(Ke + B\dot{e} \right) \tag{3}$$

While the control law in (2) becomes:

$$\tau = h + J^T \left(Ke + B\dot{e} + \mu \right) + \left(I - J^T J^{T\dagger} \right) \tau_0 \tag{4}$$

For our purposes, since, as said, K and B are user defined, we have decided for our purposes to fix matrix K to be a diagonal matrix of order $\mathbb{R}^{3\times3}$ with magnitude 10^3.

$$K = 10^3 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

As for the damping matrix B instead, we have set the damping ratio ξ to be the critical damping ($\xi = 1$) in order not to have oscillations. From there, we imposed B to be an identity matrix whose diagonal elements are calculated extracting the elements on the diagonal of Λ which represents our desired mass matrix. Therefore each element on the diagonal is calculated as:

$$B[i,i] = 2\xi\sqrt{km_i} \quad \text{where: } m_i = \Lambda[i,i]$$

Also, during our simulation we have kept k_p as it was in the original template with $k_d = \sqrt{k_p}$. Same for the values of friction needed later.

2.1 No Friction

Among the four different implementation of the same controller, we can say that every controller succeeds in making the position converge to the desired one, except for the `IC_0` (exact version of the IC). In fact:

`IC_0_simpl`	`IC_0_simpl_post`	`IC_0`	`IC_0_post`
0.011 m	0.008 m	Nan	0.008 m

Table 1: Tracking errors in case of no friction

The reason why that `IC_0` diverges is to find in the definition of the bias forces μ. In fact it is defined as:

$$\mu = \Lambda \left(JM^{-1}h - \dot{J}\dot{q} \right)$$

Where h is a matrix containing all the non linear effects while $\dot{J}\dot{q}$ is a second order differential which grows exponentially. Usually this term can be neglected under the assumption of small velocities.

The other controls manage to stabilize the desired position because they are imbued with the postural task, the first case instead, even though it doesn't have the postural task, doesn't diverge because we made the assumption that joint velocities won't be high, doing so we don't multiply the term $\dot{J}\dot{q}$ by J^T and we don't encounter the numerical divergence.

Non linear effects are portrayed by h which is simply added to the control law remaining reasonably bounded.

2.2 Friction

When friction is introduced, the controller faces a challenge in consistently achieving the desired end effector position. This issue arises because the control torque is primarily influenced by the positional error.

As the error decreases and approaches zero, the control torque also diminishes. In such cases, the control torque can eventually become nearly equal in magnitude but opposite in direction to the combined effect of friction torque and gravity compensation.

This leads to a state of torque equilibrium, preventing the robot from reaching and maintaining the desired position due to the counteracting forces of friction and gravity.

This also makes the `IC_0` work because the friction acts like a mechanical impedence.

In the table below we report the tracking errors for the controller:

`IC_0_simpl`	`IC_0_simpl_post`	`IC_0`	`IC_0_post`
0.015 m	0.019 m	0.012 m	0.020 m

Table 2: Tracking errors in case friction is present

3 Stabilize reference trajectory with IC and OSC

With the Operational-Space control we want to regulate the robot's movement in its operational space, which is the space where the robot performs its actual work of the following control law:

$$\tau = J^T f_d + \left(I - J^T J^{T\dagger} \right) \tau_1 \tag{5}$$

Where f_d represents the desired force, which depends on a desired acceleration that can be described using a PD controller. The formula is as follows:

$$\ddot{x}_{fb} = k_p \left(x_d - x \right) + k_d \left(\dot{x}_d - \dot{x} \right) \tag{6}$$

Obviously k_p and k_d are the coefficient of the PD controller.

3.1 IC vs OSC

The current objective is to test the two controllers, OSC and IC, in two distinct configurations, each with different frequencies.

The OSC controller has demonstrated overall superior performance compared to the IC controller. However, it's important to note that, as highlighted in `Figure16`, the OSC controller exhibited significantly higher torque peaks in the initial phases of the simulation, which could raise concerns in a practical implementation.

Furthermore, both the OSC and IC controllers, as indicated in table below, showed a deterioration in performance as the frequency increased, with a change in joint configurations leading to a higher average tracking error.

In summary, the OSC controller appears to offer better trajectory tracking performance than the IC controller, but it may require special attention to manage the initial torque peaks. Both controllers struggled to maintain high performance at higher frequencies.

In the table, we report the tracking errors for the controller, in `Figure 23` there is a summary for the errors.

OSC_f1	IC_f1	OSC_f3	IC_f3
0.018 m	0.021 m	0.049 m	0.060 m

Table 3: Comparison at different frequencies

3.2 Improvements

In order to improve the performances of the controllers we look at the control laws defined in (5) and (4).

As we may see in the OSC implemented, the feedback acceleration is controlled with a PD controller. We can improve it by adding to the control law an integral term to better estimate the dynamics of the error between the desired position and the actual one. Thus we get a PID controller:

$$\ddot{x}_{fb} = k_p\left(x_d - x\right) + k_d\left(\dot{x}_d - \dot{x}\right) + k_i \int \left(x_d - x\right) dt \qquad \Rightarrow \tau = J^T f_d + \left(I - J^T J^{T\dagger}\right)\tau_1 \tag{7}$$

Also, if we increase the value of k_p we improve the proportional term of the PID. This all contributes to reduce the error for the OSC control.

For the IC, instead, we can try to make a more reasonable tuning of the stiffness and damping matrices in order to improve the dynamics of the system. In order to do that, keeping the implementation for B as before, we simply increase the value of the stiffness up to a certain feasible value.

We have set $k_p = 550$, $k_i = 30$, and the magnitude of the stiffness matrix K has been increased to $k = 10^4\ Nm$. The errors obtained are reported in the table below and in `Figure 24` we can find a summary for the errors.

OSC_f1	IC_f1	OSC_f3	IC_f3
0.004 m	0.002 m	0.012 m	0.013 m

Table 4: Errors for Improved Controllers

3.3 Randomize

Our conservative tuning approach paid off. Enabling the "randomize robot model" feature increased average tracking errors, as expected. However, through multiple script runs, we consistently observed the controllers' resilience, affirming that our tuning efforts produced two robust controllers.

In the plots in figures `Figure 25` to `Figure 30` we have reported 6 tests for our controllers in random mode. This shows that our design is robust with respect to the robot model.

Controllers IC kp=100 friction = 0

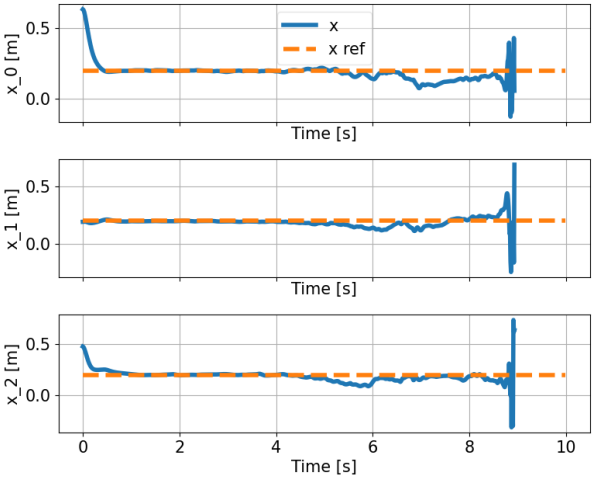

Figure 1: Position of the EE for IC_0

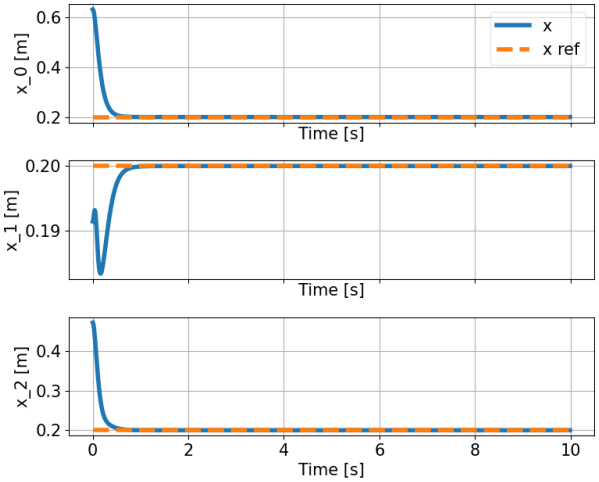

Figure 2: Position of the EE for IC_0_post

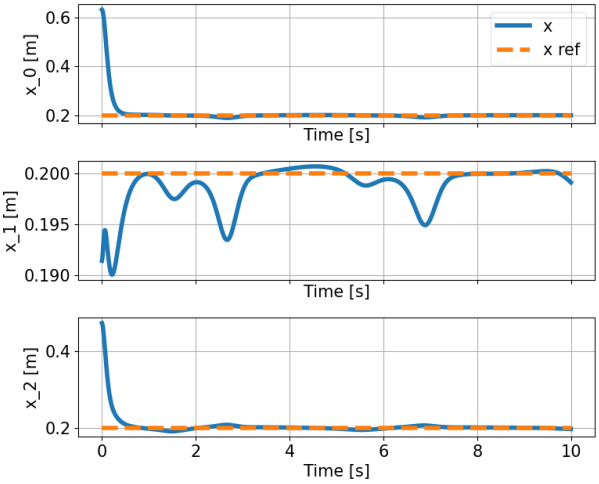

Figure 3: Position of the EE for IC_0_simpl

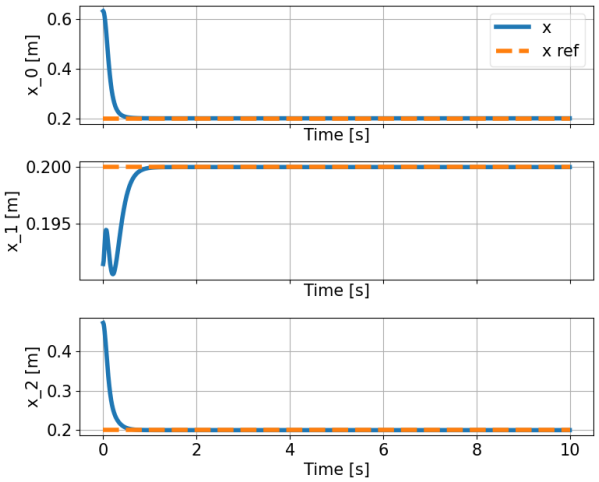

Figure 4: Position of the EE for IC_0_simpl _post

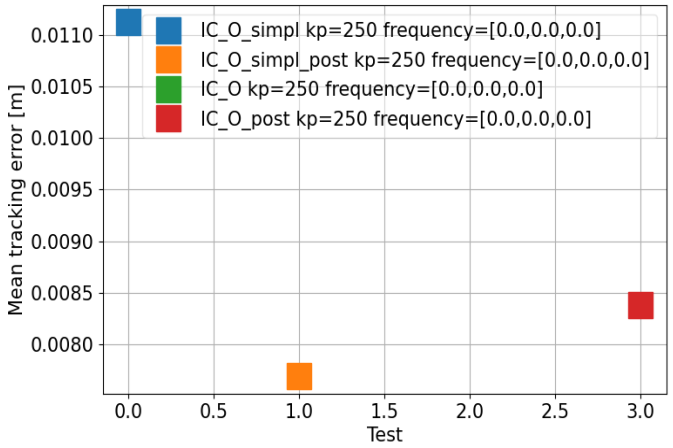

Figure 5: Errors compared

Controllers IC kp=100 friction = 2

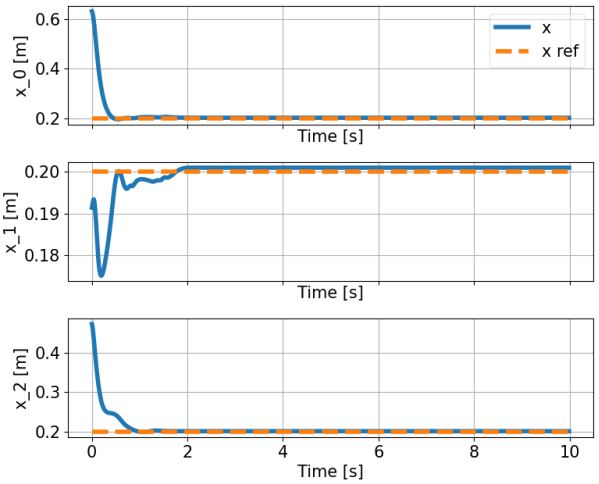

Figure 6: Position of the EE for IC_0

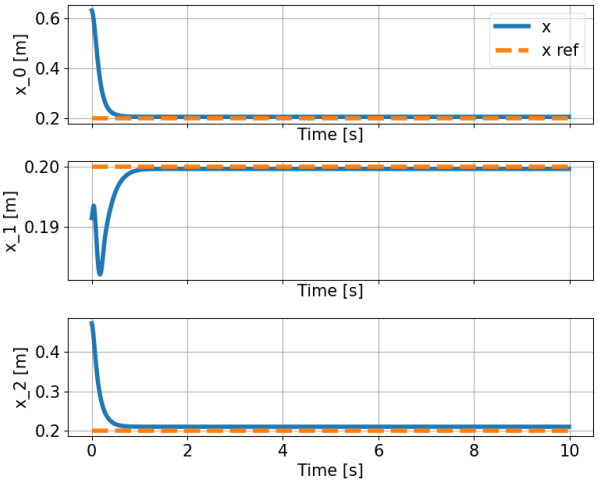

Figure 7: Position of the EE for IC_0_post

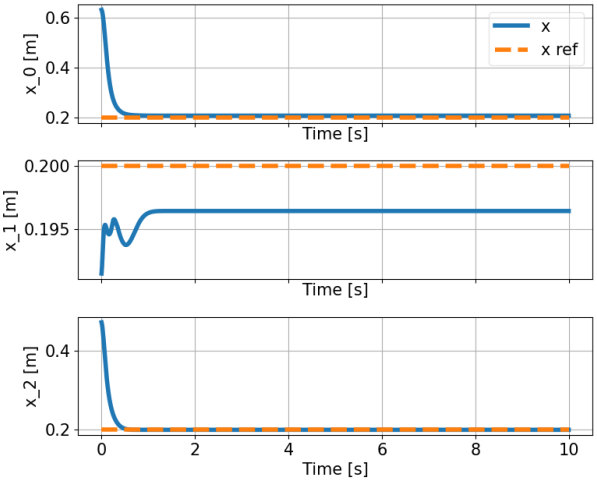

Figure 8: Position of the EE for IC_0_simpl

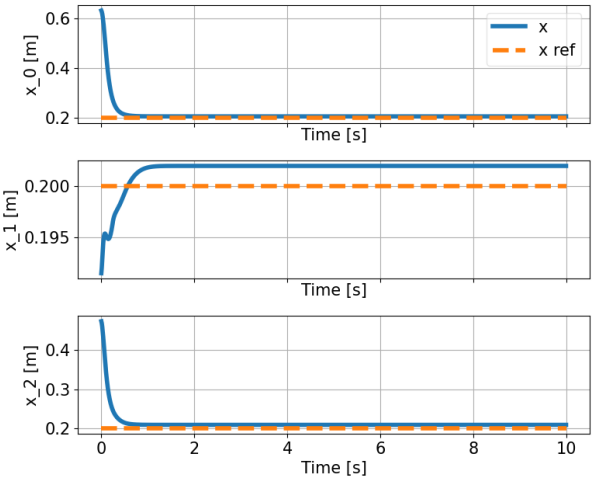

Figure 9: Position of the EE for IC_0_simpl _post

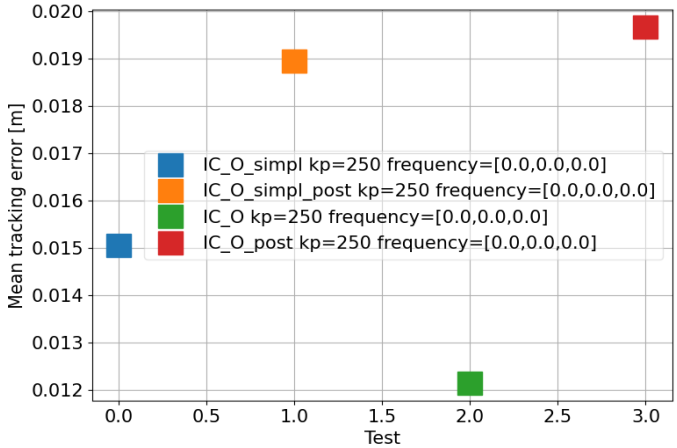

Figure 10: Errors

Controller OSC kp=100 frequency=[1.0, 1.0, 0.3] friction = 2

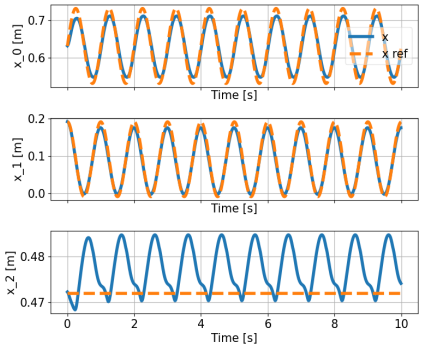

Figure 11: Position of the EE

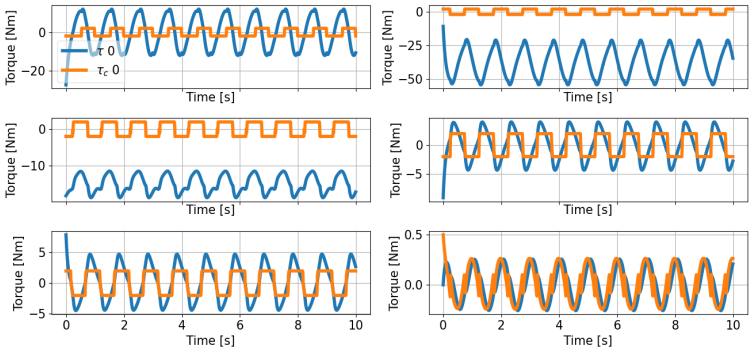

Figure 12: Torque

Controller IC kp=100 frequency=[1.0, 1.0, 0.3] friction = 2

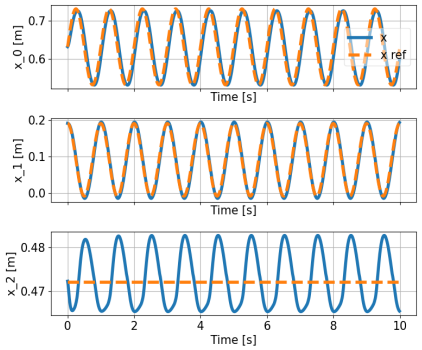

Figure 13: Position of the EE

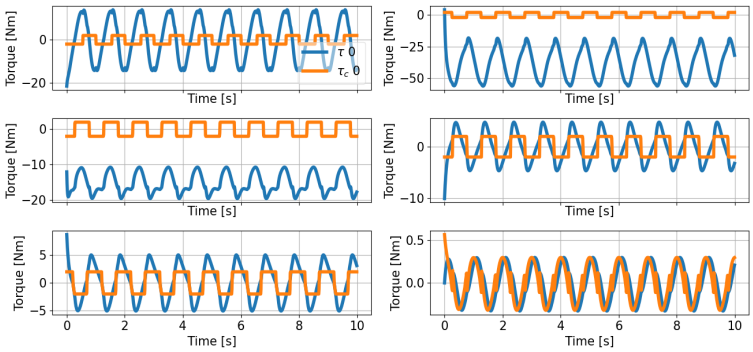

Figure 14: Torque

Controller OSC kp=100 frequency=[3.0, 3.0, 0.9] friction = 2

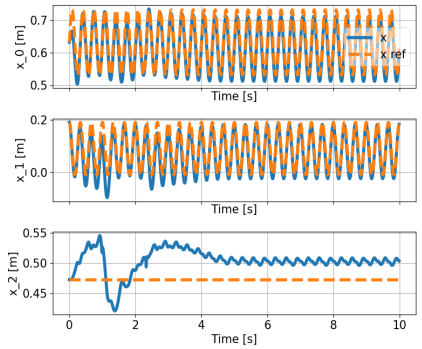

Figure 15: Position of the EE

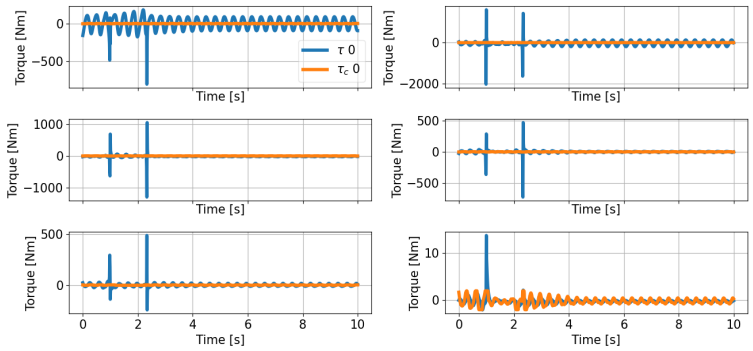

Figure 16: Torque

Controller IC kp=100 frequency=[3.0, 3.0, 0.9] friction = 2

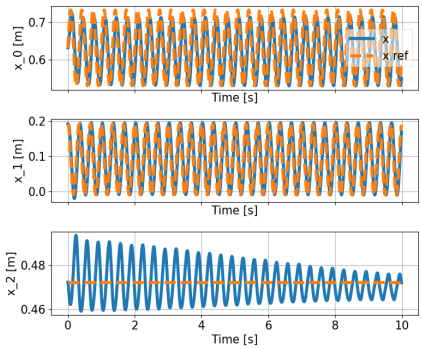

Figure 17: Position of the EE

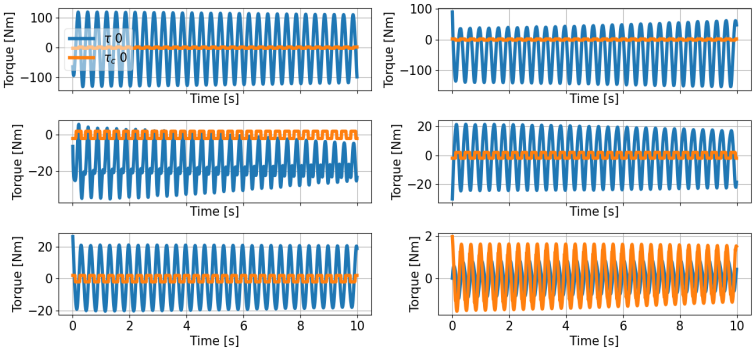

Figure 18: Torque

Controller's Improvements

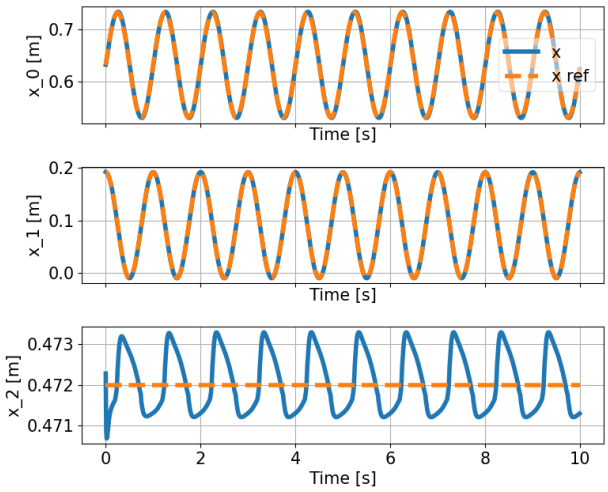

Figure 19: Position of the EE for IC with $f = 1$

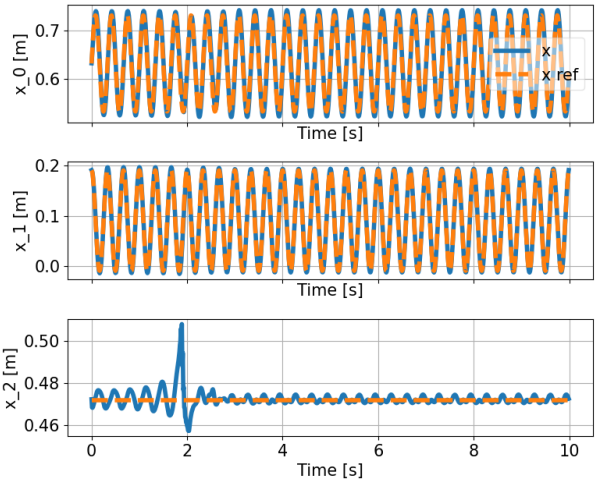

Figure 20: Position of the EE for IC with $f = 3$

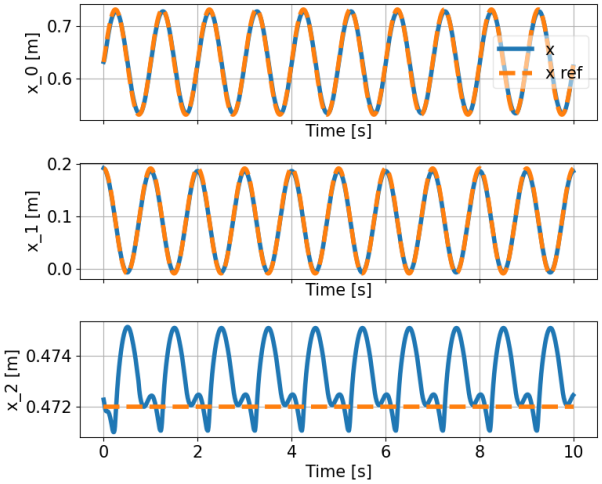

Figure 21: Position of the EE for OSC with $f = 1$

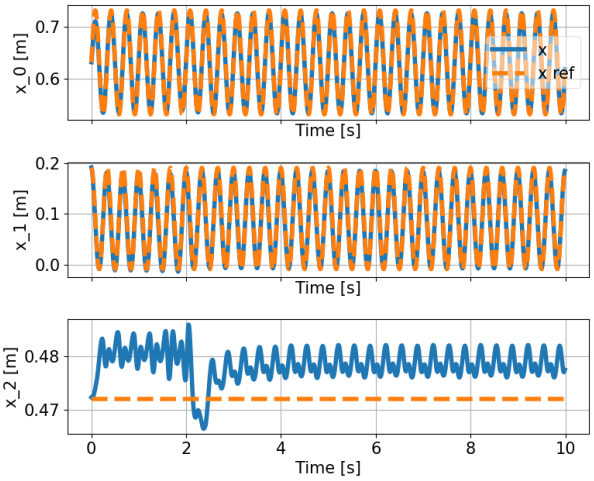

Figure 22: Position of the EE for OSC with $f = 3$

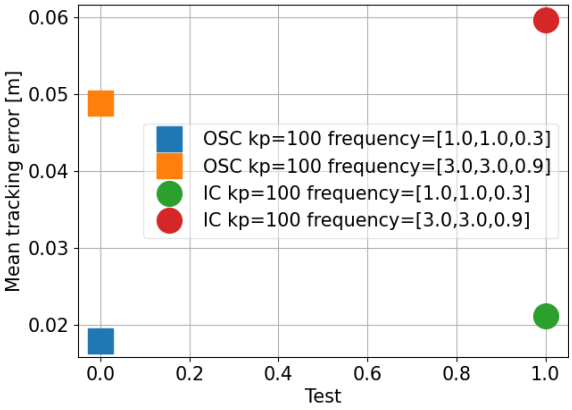

Figure 23: Comparison ICvsOSC at F&3F

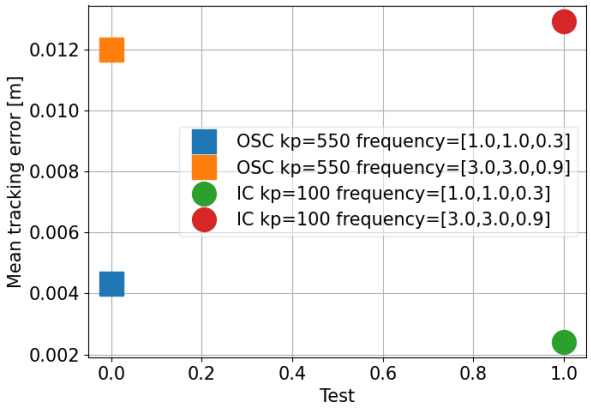

Figure 24: Comparison ICvsOSC improved at F&3F

Randomize

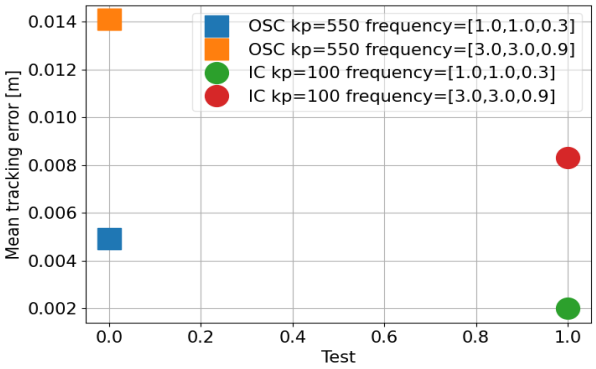

Figure 25: Test 1

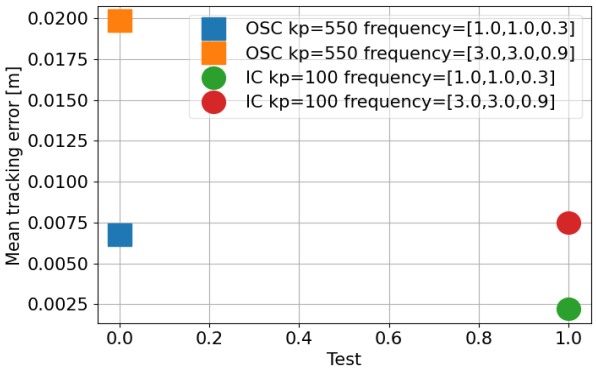

Figure 26: Test 2

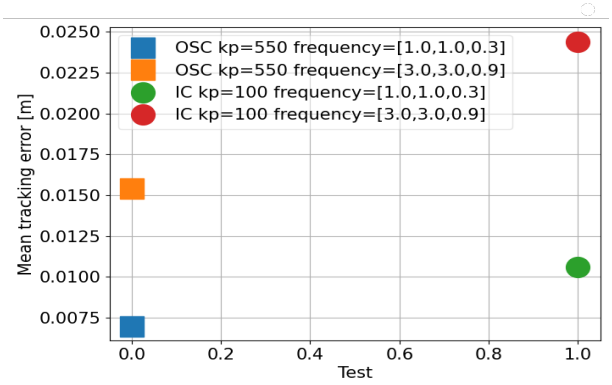

Figure 27: Test 3

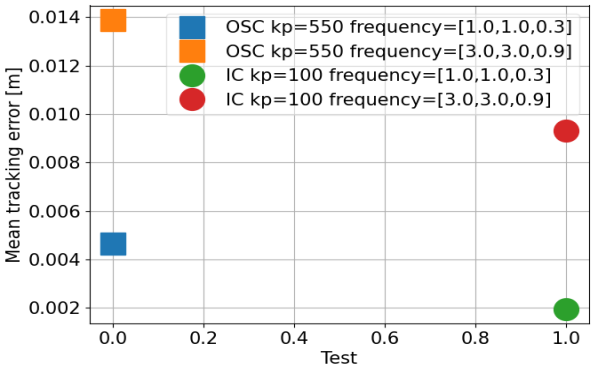

Figure 28: Test 4

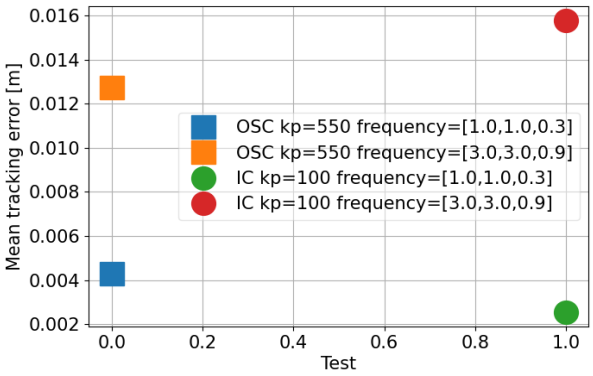

Figure 29: Test 5

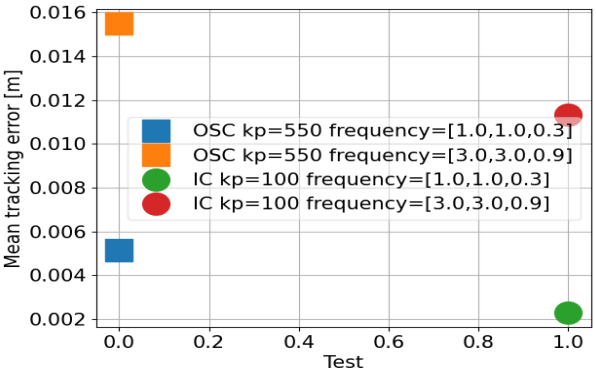

Figure 30: Test 6