# ARC Assignment 02: Operational-Space Control VS Impedance Control

Andrea Carollo [206343] - Matteo Tomasi [206334]
Advanced Optimization-based Robot Control
University of Trento

May 3, 2020

## Question 1

**What can you say about the relative performance of OSC and IC in the different settings? Which controller performed best without friction? Which one performed best with friction? Report the values of the tracking errors that you got in simulation.**

We can start from the tracking error of the two control methods - OSC and IC - using different gains in the PD controller and subjected to different levels of friction.

| Test | Controller | kp | Friction [%] | Tracking errors [m] |
|---|---|---|---|---|
| 0 | OSC | 50 | 0 | 0.043 |
| 1 | IC | 50 | 0 | 0.014 |
| 2 | OSC | 100 | 0 | 0.023 |
| 3 | IC | 100 | 0 | 0.008 |
| 4 | OSC | 100 | 2 | 0.029 |
| 5 | IC | 100 | 2 | 0.017 |
| 6 | OSC | 200 | 2 | 0.017 |
| 7 | IC | 200 | 2 | 0.010 |
| 8 | OSC | 200 | 4 | 0.021 |
| 9 | IC | 200 | 4 | 0.016 |
| 10 | OSC | 400 | 4 | 0.012 |
| 11 | IC | 400 | 4 | 0.009 |

Table 1: Table with the experiments

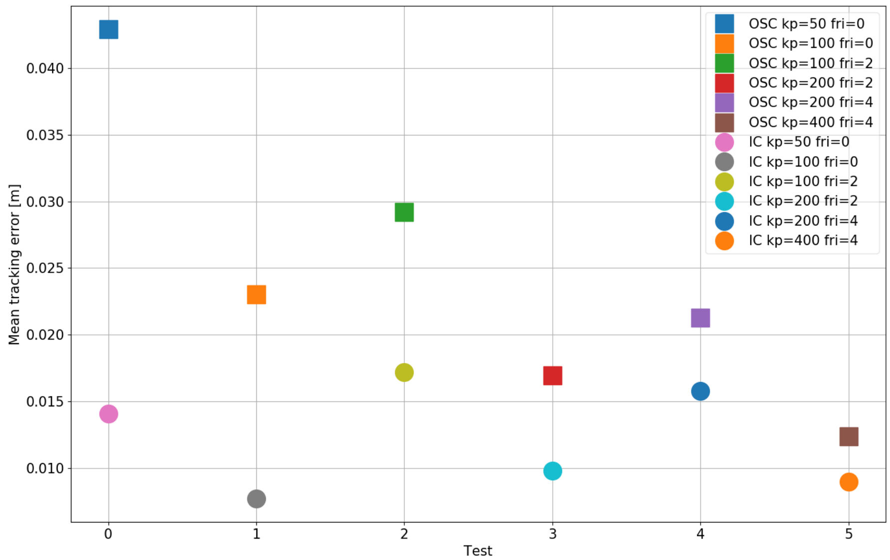


Figure 1: Plot of the experiments - Tab. 1

From table 1 and figure 1, we can see that the tracking error is lower if we use the Interaction Control - IC - regardless the presence of friction. However, using the plot in figure 1, we can argue that the difference in tracking error decreases, increasing the friction level. This means that, for sure, IC works better at low level of friction but, if we increase the friction level, the OSC starts to work better, while IC keeps its tracking error around a constant value. If we test the two algorithms at high value of friction - 20% of the maximum joint torque - the tracking errors assume almost the same value - Tab. 2 -. However, this happens for a very high value of joint friction that should be never reached in a real application.

| Test | Controller | kp | Friction [%] | Mean tracking errors [m] |
|---|---|---|---|---|
| 12 | OSC | 1100 | 20 | 0.014 |
| 13 | IC | 1100 | 20 | 0.013 |
| 14 | OSC | 1400 | 20 | 0.011 |
| 15 | IC | 1400 | 20 | 0.011 |

Table 2: Table with additional the experiments

Additionally, if we try to reach an unreachable point using the OSC, the algorithm is not able to solve the problem.

# Question 2

**Set to 1 the flag *randomize_robot_model* in the configuration file and re-run the script. This flag enables the random modification of the inertial parameters of the simulated robot, making it different (at most 30%) from the model used by the controllers. How did the performance of OSC and IC change? Which of the two controllers was most robust to the introduced modeling errors?**

To see which control is more robust with respect to the model randomization, some tests are performed in order to be sure that the simulation taken in consideration will not provide a particular result. In the following table, we show the mean differences - in tracking error - between the randomize and non-randomize model simulation, defined as 1.

$$\Delta e = \frac{1}{n} \sum_{i=1}^{n} (e_{rand,i} - e_i) \tag{1}$$

Where $n$ is the number of simulations runned.

| Test | Controller | kp | Friction [%] | Average differences of tracking error [m] |
|---|---|---|---|---|
| 0 | OSC | 50 | 0 | -0.005 |
| 1 | IC | 50 | 0 | 0.015 |
| 2 | OSC | 100 | 0 | 0.010 |
| 3 | IC | 100 | 0 | 0.007 |
| 4 | OSC | 100 | 2 | 0.014 |
| 5 | IC | 100 | 2 | 0.003 |
| 6 | OSC | 200 | 2 | 0.006 |
| 7 | IC | 200 | 2 | 0.002 |
| 8 | OSC | 200 | 4 | 0.004 |
| 9 | IC | 200 | 4 | 0.000 |
| 10 | OSC | 400 | 4 | 0.002 |
| 11 | IC | 400 | 4 | 0.001 |

Table 3: Differences in tracking error with randomized model

From table 3, we can see that the minor difference between the randomize case and the ideal one, is when we use the IC. For this reason, we can argue that the IC is more robust to the randomization.
The first row of the table corresponds to the only case in which the OSC gives better results with respect to the IC. Note that this happens in the case in which the difference between IC and OSC is greater with the non-randomize test only. This could be due to the randomization: there is an experiment in which the difference with respect to the ideal case assumes a very large value, for IC, and a low value for OSC. This could move the average in that direction. However, also removing that experiment form the computation, the OSC gives a smaller deviation from the ideal case. This suggests that OSC is more robust to the

randomization w.r.t. the IC but the robustness is greater only in this specific case - low gains and no friction -. Nevertheless, the mean tracking error is lower if we use the IC.

# Question 3

**The impedance control law $\tau = h + J^T(B\dot{e} + Ke)$ does not exactly achieve the desired impedance behavior $\Lambda\ddot{x} + B\dot{e} + Ke = f_{ext}$. Why? How could you modify the control law to achieve exactly the desired behavior?**

From the mathematical passages done to extract the torque control law, some simplifications were done because of negligible or small dimensional order terms. To compensate these cancellations, we can find a constant $\mathbf{C}$ that makes equal the two equations, so getting the same behaviour. We add the constant $\mathbf{C}$ to the first equation of $\tau$:

$$M\ddot{q} + h - J^T f_{ext} = \tau \tag{2}$$

$$\begin{aligned} \tau &= h - J^T(B\dot{e} + Ke) + \mathbf{C} \\ e &= x - x_{ref} \end{aligned} \tag{3}$$

Comparing the two values of torque we get:

$$M\ddot{q} + -J^T f_{ext} = -J^T(B\dot{e} + Ke) + \mathbf{C} \tag{4}$$

Pre-multiply equation 4 by $JM^{-1}$:

$$JM^{-1}(M\ddot{q} + -J^T f_{ext}) = JM^{-1}(-J^T(B\dot{e} + Ke) + \mathbf{C}) \tag{5}$$

Knowing that:

$$\begin{aligned} J\ddot{q} &= \ddot{x} - \dot{J}\dot{q} \\ \Lambda^{-1} &= JM^{-1}J^T \end{aligned} \tag{6}$$

Then, we substitute 6 into 5:

$$\Lambda\ddot{x} - \Lambda\dot{J}\dot{q} - f_{ext} = -(B\dot{e} + Ke) + \Lambda JM^{-1}\mathbf{C} \tag{7}$$

Reordering the equation 7, we obtain:

$$\Lambda\ddot{x} + B\dot{e} + Ke = f_{ext} + \Lambda\dot{J}\dot{q} + \Lambda JM^{-1}\mathbf{C} \tag{8}$$

Using the definition of $f_{ext}$ given in the question, the equation 8 becomes:

$$\Lambda\dot{J}\dot{q} + \Lambda JM^{-1}\mathbf{C} = 0 \tag{9}$$

Finally, we solve 9 for the constant $\mathbf{C}$ (where $\dagger$ is the pseudo-inverse operator):

$$\mathbf{C} = -(JM^{-1})^{\dagger}\dot{J}\dot{q} \tag{10}$$

In conclusion, to have the same impedance on the control law, we must add the contribute of 10 to the equation 3:

$$\tau = h - J^T(B\dot{e} + Ke) - (JM^{-1})^{\dagger}\dot{J}\dot{q} \tag{11}$$

# Question 4

**Try to modify the impedance control law that you implemented to include also the term you added in the previous question. Does this lead to an improvement in the tracking performance? Why?**

Adding the contribute of found term to the control law does not significantly increase the performance. This because the contribute of that term is very small and actually negligible. We increase the number of significant digits to see the small contribute of it and we test the control without noise on the model - *randomize_robot_model* = 0 - focusing only on the contribute of the modified control law.

| Test | Controller | kp | Friction [%] | Mean tracking error without correction [m] | Mean tracking error with correction [m] |
|---|---|---|---|---|---|
| 1 | IC | 50 | 0 | 0.014087 | 0.014580 |
| 3 | IC | 100 | 0 | 0.007715 | 0.007956 |
| 5 | IC | 100 | 2 | 0.017196 | 0.017331 |
| 7 | IC | 200 | 2 | 0.009796 | 0.009852 |
| 9 | IC | 200 | 4 | 0.015807 | 0.015570 |
| 11 | IC | 400 | 4 | 0.009003 | 0.008835 |

Table 4: Comparison of control laws with and without correction

By looking at the table 4, the contribute of the added term is negligible.

# Question 5

**(Optional) Which control law would you choose to implement on a real robot? Why? And how could you try to improve its performance? Discuss different ways, pros and cons, possibly showing plots and tracking errors that you obtained by testing your ideas in simulation.**

We would implement the IC control law on real robots because of the better performance in tracking error - with and without model randomization -. In addition, the IC computational load is lower than OSC. In fact, OSC implements the computation of $\Lambda$ that involves a double inversion of large matrices - neglecting the adding term that involves a similar computation of $\Lambda$ -. Moreover, if we try to reach an out-of-workspace point, the OSC law fails, while the IC law still gives us a solution.
Firstly, to improve the algorithm performances, we can easily increase the controller gains - *kp* and *kd* - up to a certain value in order to avoid overshoot and instability. This is confirmed also by the previous experiments, where we can see a decreasing of tracking error by increasing the gains.
Secondly, we can improve the performance by increasing the frequency of the control. However, testing the control with an higher frequency we did not get a significant improvement.
Thirdly, to improve the performance, we may include a friction compensation term. We

used an estimated joint friction torque based on a smooth version of the Coulomb model - $\tau_f = \tau_f^{MAX}\ tanh(k\ \dot{q})$ , with $k = 20$ -. Using this strategy, as we can see in table 5, we got an improvement in tracking error. Additionally, we note that the improvement due to the friction compensation is lower using the OSC. This strengthens our choice to use IC algorithm.

| Test | Controller | kp | Friction [%] | Tracking errors [m] | Tracking errors with friction compensation [m] |
|---|---|---|---|---|---|
| 4 | OSC | 100 | 2 | 0.029 | 0.027 |
| 5 | IC | 100 | 2 | 0.017 | 0.012 |
| 6 | OSC | 200 | 2 | 0.017 | 0.017 |
| 7 | IC | 200 | 2 | 0.010 | 0.006 |
| 8 | OSC | 200 | 4 | 0.021 | 0.018 |
| 9 | IC | 200 | 4 | 0.016 | 0.010 |
| 10 | OSC | 400 | 4 | 0.012 | 0.007 |
| 11 | IC | 400 | 4 | 0.009 | 0.005 |

Table 5: Comparison with and without friction compensation

Fourthly, in our tests, *kp* and *kd* are one-dimensional constants, but they could be generalized in matrix form to reach a different behaviour for different axis. Moreover, we could link two axis by using a non-diagonal matrix.
Fifthly, in the Interaction Control, the *kp* and *kd* have a physical meaning, i.e. they simulate a spring-damper system that constrains the end-effector position to a reference - system equation 12 -. We can use this physical meaning to make a modal analysis to tune the controller: if we focus on an interesting point - to keep fixed the system mass matrix -, we can study the modes of the system and find the best *kp* and *kd* - or the corresponding matrices - for any application. However, we must keep in mind that the mass matrix is configuration dependent, thus the modal analysis can be valid only around a given configuration.

$$\Lambda\ddot{x} + kd\ (\dot{x} - \dot{x}_{ref}) + kp\ (x - x_{ref}) = f_{ext} \tag{12}$$

Finally, using the IC, we can easily introduce a force controller to control also the force applied at the and-effector by a *Parallel Force-Impedance Control.*

$$\begin{aligned} \tau &= h - J^T(B\dot{e} + Ke) - J^T(f^*) \\ e &= x - x_{ref} \end{aligned} \tag{13}$$

The equation 13 can be split in three contribute:

- $J^T(B\dot{e} + Ke)$ is the Interaction Control contribute;
- $J^T(f^*)$ is the Force Control contribute;
- $h$ is the non linear term, common for the two methods