Cooperative Robotics

Authors: "names"
Email: "email"
Date: "date"

General notes

- Exercise 1 is done with the ROBUST matlab main and unity visualization tools. Exercises 2-3 are done with the Franka simulation tools.
- Comment and discuss the simulations, in a concise scientific manner. Further comments, other than the questions, can be added, although there is no need to write 10 pages for each exercise.
- Aid the discussion with screenshots of the simulated environment (compress them to maintain a small overall file size), and graphs of the relevant variables (i.e. activation functions of inequality tasks, task variables, and so on). Graphs should always report the units of measure in both axes, and legends whenever relevant.
- Report the thresholds whenever relevant.
- Report the mathematical formula employed to derive the task jacobians and the control laws when asked, including where they are projected.
- If needed, part of the code can be inserted as a discussion reference.

Use the following template when you need to discuss the hierarchy of tasks of a given action or set of actions:

Table 1: Example of actions/hierarchy table: a number in a given cell represents the priority of the control task (row) in the hierarchy of the control action (column). The type column indicates whether the objective is an equality (E) or inequality (I) one.

Task	Type	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$
Task A	I	1		1
Task B	I	2	1	
Task C	E		2	2

1 Exercise 1: Implement a Complete Mission

Implement several actions to reach a point, land, and then perform the manipulation of a target object.

Initialize the vehicle at the position:

$$\begin{bmatrix} 8.5 & 38.5 & -36 & 0 & -0.06 & 0.5 \end{bmatrix}^\top$$

Use a "safe waypoint navigation action" to reach the following position:

$$\begin{bmatrix} 10.5 & 37.5 & -38 & 0 & -0.06 & 0.5 \end{bmatrix}^\top$$

Then land, aligning to the nodule. In particular, the x axis of the vehicle should align to the projection, on the inertial horizontal plane, of the unit vector joining the vehicle frame to the nodule frame. Once landed, implement a "fixed-based manipulation action" to reach the target nodule (mimicking the scanning of the nodule). During this manipulation phase, the vehicle should not move for any reason.

1.1 Q1: Report the unified hierarchy of tasks used and their priorities in each action.

1.2 Q2: Comment the behaviour of the robots, supported by relevant plots.

1.3 Q3: What is the Jacobian relationship for the Alignment to Target control task? How was the task reference computed?

1.4 Q4: Try changing the gain of the alignment task. Try at least three different values, where one is very small. What is the observed behaviour? Implement a solution that is gain-independent guaranteeing that the landing is accomplished aligned to the target.

1.5 Q5: After the landing is accomplished, what happens if you try to move the end-effector? Is the distance to the nodule sufficient to reach it with the end-effector? Comment the observed behaviour.

1.6 Q6: Implement a solution that guarantees that, once landed, the nodule is certainly within the manipulator's workspace.

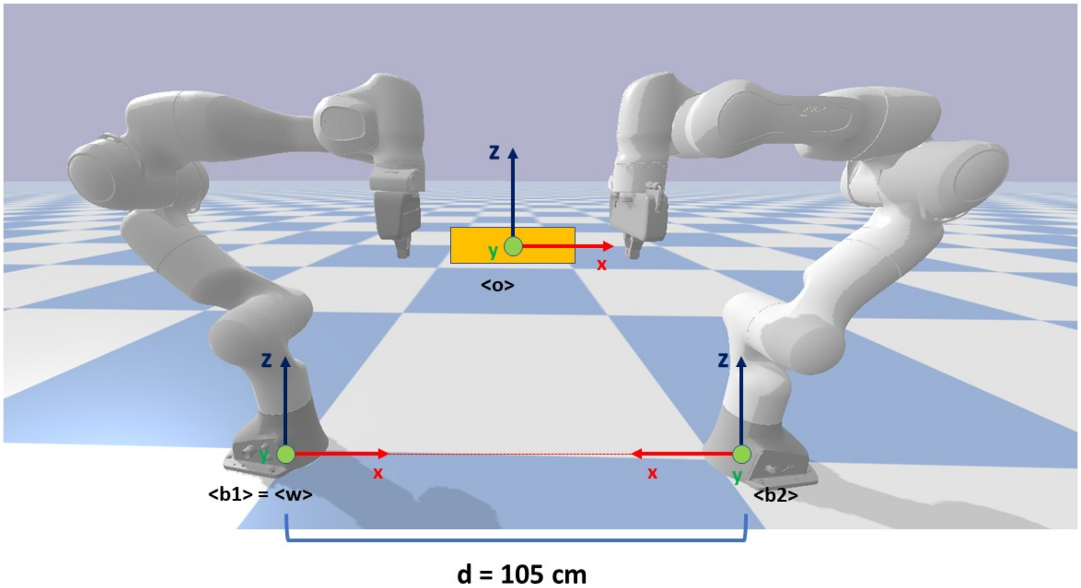

Figure 1: Basic transformation matrices

2 Exercise 2: Bimanual manipulation

In this exercise it is required to perform a bimanual manipulation by implementing the task priority algorithm considering two manipulators as a single robot. The manipulator adopted for this assignment is the Franka Panda by Emika. A simulation in python is provided, in order to visualize the two robots and test your Matlab implementation.

1. The transformations between the world and the base of each robot must be computed knowing that:
   - The left arm base coincides with the world frame.
   - The right arm base is rotated of π w.r.t. z-axis and is positioned 1.05 meters along the x-axis, see figure 1
2. The transformation from each base to the respective end-effector is given in the code.
3. Define the tool frame for both manipulators, placing it at 10 cm along the z-axis of the end-effector frame, and rotated with an angle of -43.1937 degrees around the z-axis of the end-effector.
4. Implement a "move to action" that includes the joint limits task.
5. The first objective is to move the tool frame of both manipulators to the grasping points, using the "move to action". Define the goal frames for each manipulator such that their origin correspond to the grasping points depicted in Figure 2, and their orientation corresponds to the frame depicted in the same Figure. HINT: the position of the grasping points can be computed by knowing the origin of the object frame ${}^{w}\mathrm{O}_o$ and the object length reported in Figure 2

$$ {}^{w}\mathrm{O}_o = [0.5, 0, 0.59]; \tag{1} $$

$$ length = 10cm; \tag{2} $$

Once the manipulators reach the grasping points, the second phase of the mission should start. Now, implement the Bimanual Rigid Grasping Constraint task to carry the object as a rigid body.

1. Define the object frame as a rigid body attached to the tool frame of each manipulator.

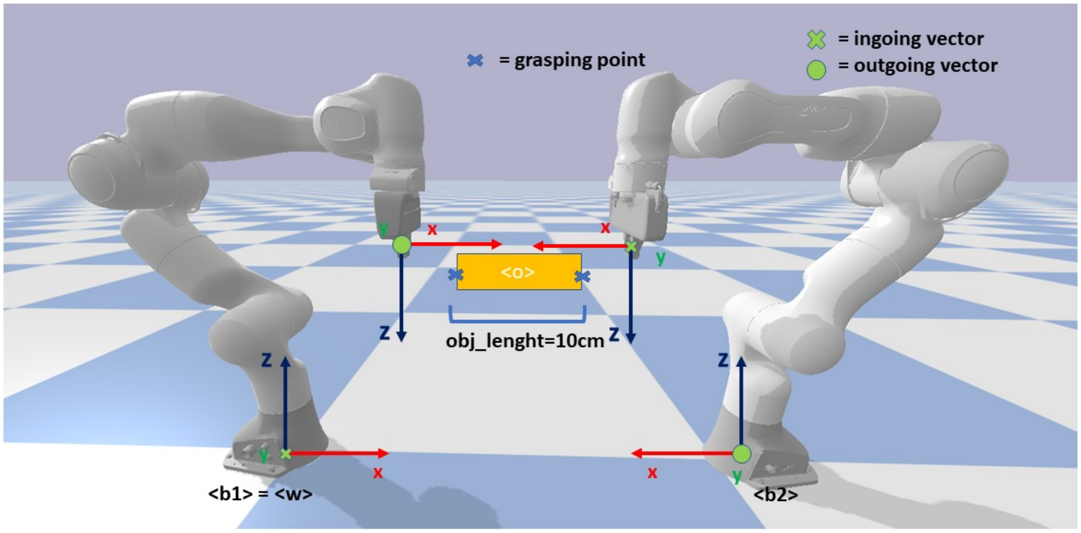

Figure 2: Tool frame positioning

2. Define the rigid grasp task.
3. Then, move the object to another position while both manipulators hold it firmly, e.g. ${}^{w}O_g = [0.5, -0.5, 0.5]$
4. Finally, find a position where one of the two manipulators reaches a joint limit, and move the object there to stress the constraint task.

2.1 Q1: Report the unified hierarchy of tasks used and their priorities in each action.

2.2 Q2: What is the Jacobian relationship for the Joint Limits task? How was the task reference computed?

2.3 Q3: What is the Jacobian relationship for the Bimanual Rigid Grasping Constraint task? How was the task reference computed?

2.4 Q4: Comment the behaviour of the robots, using relevant plots. In particular, show the difference (if any) between the desired object velocity, and the velocities of the two end-effectors in the two cases.

3 Exercise 3: Cooperative manipulation

In this exercise, it is required to perform cooperative manipulation by implementing the task priority algorithm considering the two Franka Panda manipulators as a two distinct robot.

Using the same specifications of Exercise 2, implement the cooperative manipulation policy. Again, test it twice, once with a reachable goal, and then with a goal that triggers the activation of multiple joint limits by one manipulator. The idea is that this second position should trigger the cooperation policy (i.e., the cooperative velocity should be different than the original desired object velocity).

3.1 Q1: Report the unified hierarchy of tasks used and their priorities in each action.

3.2 Q2: How many task hierarchies you have for each robot? If more than one, what is the difference?

3.3 Q3: Comment the behaviour of the robots, using relevant plots. In particular, show the difference (if any) between the desired object velocity, the non-cooperative Cartesian and the cooperative velocities of the two end-effectors in the two cases.