

COOPERATIVE ROBOTICS

Authors: Astha Gupta
EMAILs: astha736@gmail.com
Date:

General Concepts

Following the notations from the main notes. The configuration of the system is the vector of its DOF parameters.

$$c \triangleq \begin{bmatrix} q \\ \eta \end{bmatrix} \quad (1)$$

where q is the configuration of the arm.

$$\mathbf{q} \triangleq \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix} \quad (2)$$

Configuration of the vehicle base is given as below:

$$\eta \triangleq \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \in \mathbb{R}^6 \quad (3)$$

$$\eta_1 \triangleq \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \eta_2 \triangleq \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (4)$$

$$\dot{y} = \begin{bmatrix} \dot{q} \\ \nu \end{bmatrix} \quad \nu = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} \quad \begin{matrix} \nu_1 = {}^v v \\ \nu_2 = {}^v w \end{matrix} \quad (5)$$

The relationship between the derivative of Euler(RPY) angles and vehicle-base's angular velocity is given the below equation.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan(\theta) \sin(\phi) & \tan(\theta) \cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \boldsymbol{\nu}_2 \quad (6)$$

- Thus in the code the Jacobian expected is between ${}^w \dot{\mathbf{x}}$, in the world frame w and ${}^b \dot{\mathbf{y}}$, in the body frame b .

$${}^w \dot{\mathbf{x}} = \mathbf{J}^b \dot{\mathbf{y}} \quad (7)$$

where the components of velocity of the vehicle frame are expressed in the body frame.

- The basic functionality listed below has already been implemented.
 - "ha": Horizontal Altitude
 - "mu": Manipulability
 - "t": Tool Position Control

1 Exercise 1: Implement a “Safe Waypoint Navigation” Action.

1.1 Adding a vehicle position control objective

Initialize the vehicle far away from the seafloor. An example position could be

$$[10.5 \quad 35.5 \quad -36 \quad 0 \quad 0 \quad \pi/2]^\top$$

Give a target position that is also sufficiently away from the seafloor, e.g.,

$$[10.5 \quad 37.5 \quad -38 \quad 0 \quad 0 \quad 0]^\top$$

Goal: Implement a vehicle position control task, and test that the vehicle reaches the required position and orientation.

1.1.1 Q1: Report the hierarchy of task used and their priorities. What is the Jacobian relationship for the Vehicle Position control task? How was the task reference computed?

- Hierarchy of Task used and their priorities

The tasks used for the first questions are listed below. The priority of the task is the same as the enumeration, where the number increase as the priority decreases.

1. "ha": Horizontal Altitude
2. "posc": Vehicle Position Control
3. "mu": Manipulability
4. "t": Tool Position Control

For this task Horizontal Altitude is given the maximum priority, to avoid situations where the robot is upside down. Since in this project the code for the desired behaviour is being incrementally built, the position-control was the only task added at this step to the base code.

Position Control does not require Manipulability and Tool Position-Control, but they were kept to understand their effect on the overall task. Thus they could be switched off if the robot is only required to perform Position-Control.

- Jacobian Relationship for Vehicle Position Control

$$\begin{bmatrix} {}^w\dot{v}_{3 \times 1} \\ {}^w\omega_{3 \times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 7} & {}^w\mathbf{R}_v & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 7} & \mathbf{0}_{3 \times 3} & {}^w\mathbf{R}_v \end{bmatrix} \begin{bmatrix} \dot{q}_{7 \times 1} \\ {}^b v_{3 \times 1} \\ {}^b \omega_{3 \times 1} \end{bmatrix} \quad (8)$$

- Task Reference Computation

The reference velocity \dot{x}_{ref} is computed using the Cartesian Error between the goal pose(vehicle-base) and current vehicle-base pose in the world frame, using the "CartError" function, and scaling it with appropriate factor λ . For this task $\lambda = 0.2$

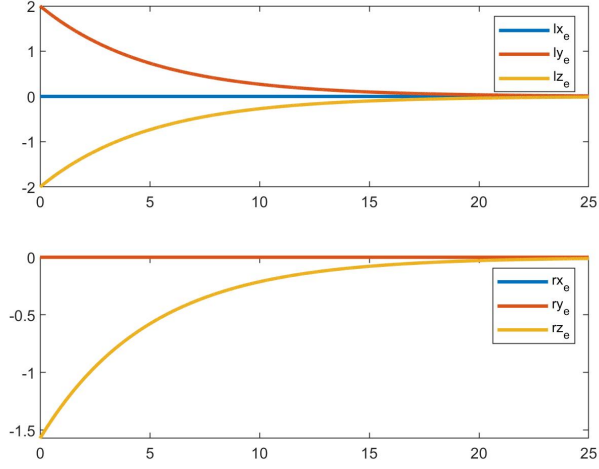
$${}^w\dot{x}_{ref} = \lambda \begin{bmatrix} \delta r \\ \delta \theta \end{bmatrix}^w \quad (9)$$

1.1.2 Q2: What is the behaviour if the Horizontal Attitude is enabled or not? Try changing the initial or target orientation in terms of roll and pitch angles. Discuss the behaviour.

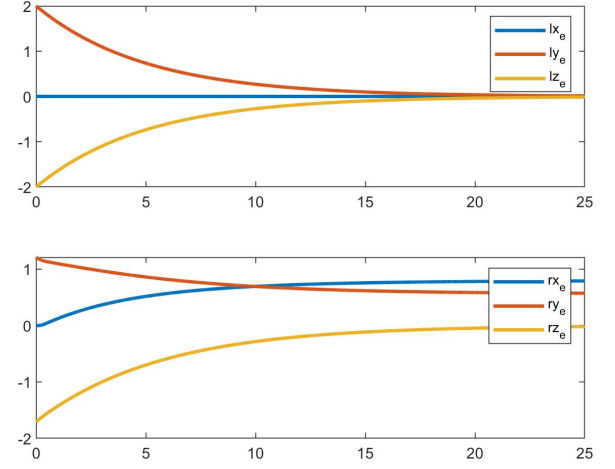
To understand the difference between behaviour, we conducted 4 runs of the simulation using different parameters as stated below.

- Default priority (dp): ["ha", "posc", "mu", "t"]

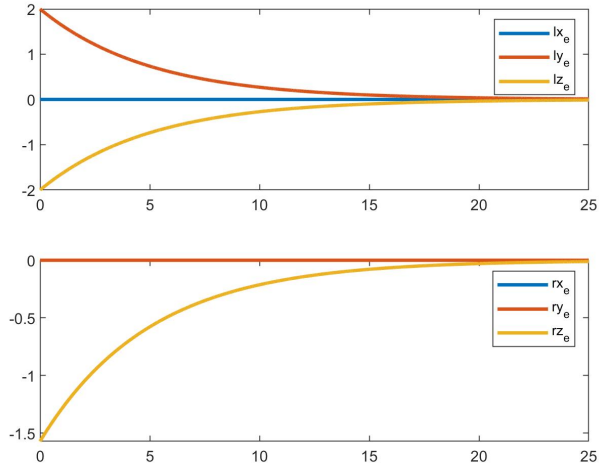
- Horizontal priority off (haOff): ["posc", "mu", "t"]
- Target default (tdef): [10.5, 37.5, -38, 0, 0, 0]
- Target new (tnew): [10.5, 37.5, -38, degtorad(45), degtorad(45), 0]



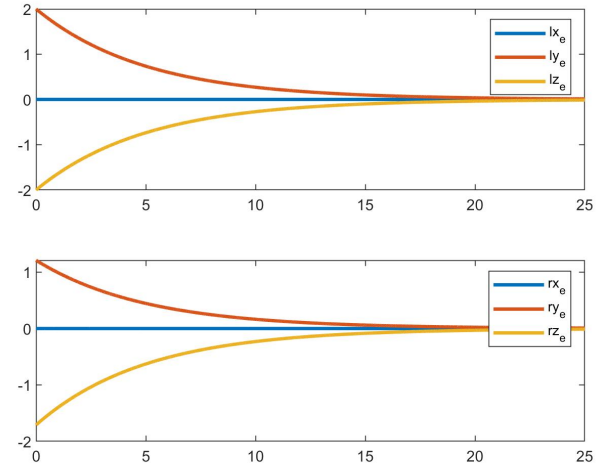
(a) (dp,tdef)



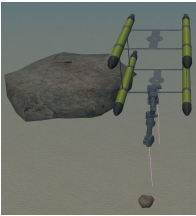
(b) (dp,tnew)



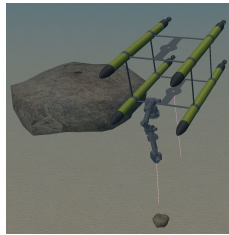
(c) (haOff,tdef)



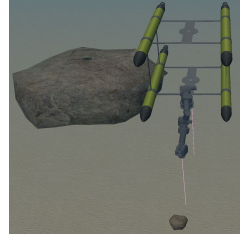
(d) (haOff,tnew)



(a) (dp,tdef)



(b) (dp,tnew)



(c) (haOff,tdef)



(d) (haOff,tnew)

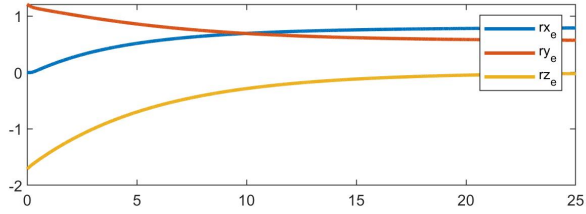
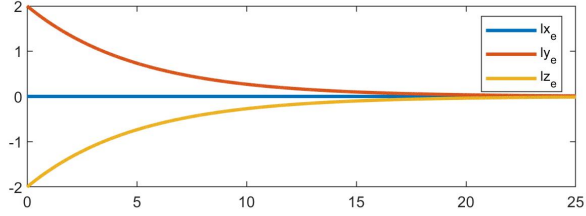
As we see that in the first case where the default priority is used the errors do not converge for the

tnew position, as the position-control and horizontal altitude have conflicting goals. Since the "ha" has higher priority, the target is not achieved for the second case.

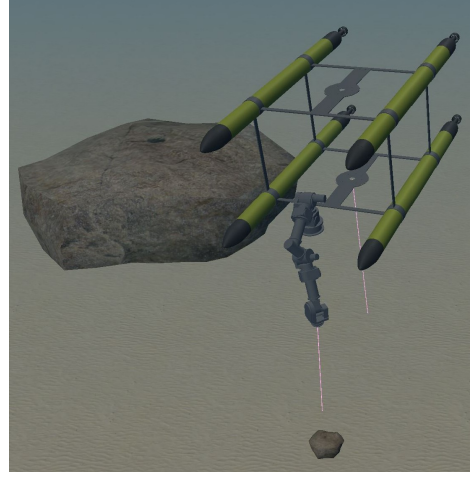
Were as the (c) and (d) part "ha" is off, thus the errors for vehicle-base converge to zero, but the robot is flipped in part (d).

1.1.3 Q3: Swap the priorities between Horizontal Attitude and the Vehicle Position control task. Discuss the behaviour.

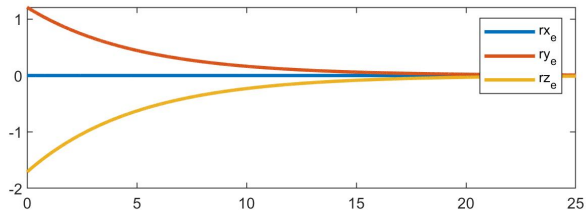
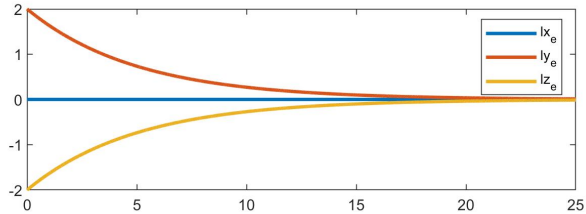
Since for target position tdef there is no substantial conflict of goals between "ha" and "posc", here we are omitting the graphs and instead focusing on results for tnew. The switched priority is (haSwt): ["posc", "ha", "mu", "t"]



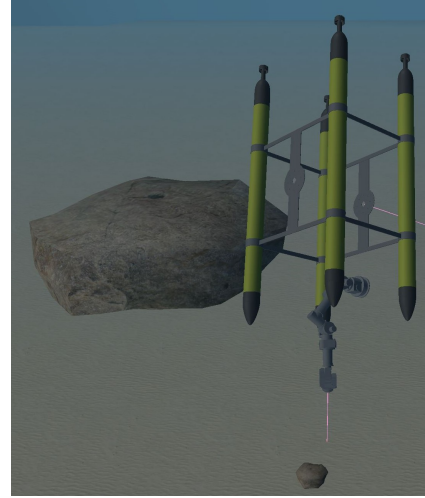
(a) (dp,tnew)



(b) (dp,tnew)



(c) (haSwt,tnew)



(d) (haSwt,tnew)

As expected in the case of default priority the "ha" takes precedence and hence the vehicle is not able to achieve a zero error for "posc". Whereas in the switched propriety case the errors for position goes to zero, but the vehicle is flipped.

1.1.4 Q4: What is the behaviour if the Tool Position control task is active and what if it is disabled? Which of the settings should be used for a Safe Waypoint Navigation action?

The behaviour of the robot with respect to achieving required pose is the same. The activation for, joint velocity remains 0 and there is no change in configuration of joints. Although there is no impact of Tool position control on vehicle position control in our case, it should be disabled for safe waypoint navigation to avoid possible collision and any other unanticipated behaviour.

1.2 Adding a safety minimum altitude control objective

Initialize the vehicle at the position:

$$[48.5 \quad 11.5 \quad -33 \quad 0 \quad 0 \quad -\pi/2]^\top$$

Choose as target point for the vehicle position the following one:

$$[50 \quad -12.5 \quad -33 \quad 0 \quad 0 \quad -\pi/2]^\top$$

Goal: Implement a task to control the altitude from the seafloor. Check that at all times the minimum distance from the seafloor is guaranteed.

1.2.1 Q1: Report the hierarchy of task used and their priorities. Comment how you choose the priority level for the minimum altitude.

The hierarchy used is given below. Since the Minimum Altitude is a safety feature so that the vehicle does not crash into the sea floor, thus it has been given the highest priority.

1. "mac": Minimum Altitude Control
2. "ha": Horizontal Altitude
3. "pose": Vehicle Position Control
4. "mu": Manipulability
5. "t": Tool Position Control

1.2.2 Q2: What is the Jacobian relationship for the Minimum Altitude control task? How was the task reference computed?

The Jacobian "Jmac" is given as below:

$$\begin{bmatrix} {}^w\dot{v}_{3 \times 1} \\ {}^w\omega_{3 \times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 7} & {}^w\mathbf{R}_v & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 7} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \dot{q}_{7 \times 1} \\ {}^b v_{3 \times 1} \\ {}^b \omega_{3 \times 1} \end{bmatrix} \quad (10)$$

The reference velocity is given by

$$[0 \quad 0 \quad v_z \quad 0 \quad 0 \quad 0]^\top$$

$$v_z = \lambda [\text{minimum.threshold} - \text{floor.distance}] \quad (11)$$

where sensor distance is the distance from sea floor given by the sensors on the vehicle.

1.2.3 Q3: Try imposing a minimum altitude of 1, 5, 10 m respectively. What is the behaviour?

1.2.4 Q4: How was the sensor distance processed?

The distance given by the sensor of the vehicle is in the z direction of the body frame. As the body frame's z axis may not be parallel to the world frame's z axis, thus the displacement vector (with only z component) was projected to the world frame. For our calculation we used only the z component of the projected vector as it would be the actual distance between the vehicle and the sea floor.

2 Exercise 2: Implement a Basic “Landing” Action.

2.1 Adding an altitude control objective

Initialize the vehicle at the position:

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

Goal: add a control task to regulate the altitude to zero.

2.1.1 Q1: Report the hierarchy of task used and their priorities. Comment how you choose the priority level for the altitude control task.

2.1.2 Q2: What is the Jacobian relationship for the Altitude control task? How was the task reference computed?

2.1.3 Q3: how does this task differs from a minimum altitude control task?

2.2 Adding mission phases and change of action

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:

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

When the position has been reached, land on the seafloor using the basic “landing” action.

2.2.1 Q1: Report the unified hierarchy of tasks used and their priorities.

2.2.2 Q2: How did you implement the transition from one action to the other?

3 Exercise 3: Improve the “Landing” Action

3.1 Adding an alignment to target control objective

If we use the landing action, there is no guarantee that we land in front of the nodule/rock. We need to add additional constraints to make the vehicle face the nodule. The position of the rock is contained in the variable `rock_center`.

Initialize the vehicle at the position:

$$[8.5 \quad 38.5 \quad -36 \quad 0 \quad -0.06 \quad 0.5]^\top$$

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

$$[10.5 \quad 37.5 \quad -38 \quad 0 \quad -0.06 \quad 0.5]^\top$$

Then land, aligning to the nodule.

Goal: Add an alignment task between the longitudinal axis of the vehicle (x axis) and the nodule target. 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.

- 3.1.1 Q1: Report the unified hierarchy of tasks used and their priorities. Comment the behaviour.**
- 3.1.2 Q2: What is the Jacobian relationship for the Alignment to Target control task? How was the task reference computed?**
- 3.1.3 Q3: Try changing the gain of the alignment task. Try at least three different values, where one is very small. What is the observed behaviour?**
- 3.1.4 Q4: 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.**

4 Exercise 4: Implementing a Fixed-base Manipulation Action

4.1 Adding non-reactive tasks

To manipulate as a fixed based manipulator, we need to constraint the vehicle to not move, otherwise the tool frame position task will make the vehicle move.

Goal: Add a constraint task that fixes the vehicle velocity to zero. Land on the seafloor. Try reaching the rock position with the end-effector, and observe that the vehicle does not move.

4.1.1 Q1: Report the unified hierarchy of tasks used and their priorities. At which priority level did you add the constraint task?

4.1.2 Q2: What is the Jacobian relationship for the Vehicle Null Velocity task? How was the task reference computed?

4.1.3 Q3: Suppose the vehicle floating, i.e. not landed on the seafloor. What would happen, if due to currents, the vehicle moves?

4.2 Adding a joint limit task

Let us now constrain the arm with the actual joint limits. The vector variables `uvms.jlmin` and `uvms.jlmax` contain the maximum and minimum values respectively.

Goal: Add a joint limits avoidance task. Land on the seafloor. Try reaching the rock position with the end-effector, and observe that the vehicle does not move and that all the joints are within their limits.

4.2.1 Q1: Report the unified hierarchy of tasks used and their priorities. At which priority level did you add the constraint task?

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

5 Exercise 5: Floating Manipulation

5.1 Adding an optimization control objective

Use the DexROV simulation for this exercise.

The goal is to try to optimize the joint positions, if possible, to keep the first four joints in a "preferred shape", represented by the following vector

$$[-0.0031 \quad 1.2586 \quad 0.0128 \quad -1.2460]^\top$$

Goal: Add an optimization objective to keep the first four joints of the manipulator in the preferred shape. Observe the behaviour with and without the task

5.1.1 Q1: Report the unified hierarchy of tasks used and their priorities. At which priority level did you add the optimization task?

5.1.2 Q2: What is the Jacobian relationship for the Joint Preferred Shape task? How was the task reference computed?

5.1.3 Q3: What is the difference between having or not having this objective?

5.2 Adding mission phases

Let us now structure the mission in more than one phase. In the first phase, exploit the previous exercises, and implement a safe waypoint navigation. Move the vehicle to a location close to the current defined end-effector goal position, just slightly above it. Then, trigger a change of action and perform floating manipulation.

Goal: introduce mission phases in the floating manipulation scenario. Observe the difference.

5.2.1 Q1: Report the unified hierarchy of tasks used and their priorities. Which task is active in which phase/action?

5.2.2 Q2: What is the difference with the previous simulation (still in exercise 5), where only one action was used?

6 Exercise 6: Floating Manipulation with Arm-Vehicle Coordination Scheme

6.1 Adding the parallel arm-vehicle coordination scheme

Let us now see how the two different subsystems (arm and vehicle) can be properly coordinate. Introduce in the simulation a sinusoidal velocity disturbance acting on the vehicle, and assume the actual vehicle velocity measurable. To do so, add a constant (in the inertial frame) velocity vector to the reference vehicle velocity before integrating it in the simulator.

Goal: modify the control part to implement the parallel arm-vehicle coordination scheme. Observe that, even with a disturbance acting on the vehicle, the end-effector can stay in the required constant position.

6.1.1 Q1: Which tasks did you introduce to implement the parallel coordination scheme?

6.1.2 Q2: What happens if the sinusoidal disturbance becomes too big?