Cooperative Robotics

Pascal Quach*[1]

[1]Università di Genova

March 25, 2023

*s5335541@studenti.unige.it

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

1.1 Adding a vehicle position control objective

Q1. Initialize the vehicle far away from the seafloor. An example position could be

$$\mathbf{p} = \begin{bmatrix} 10.5 & 35.5 & -36 & 0 & 0 & \pi/2 \end{bmatrix}^{\mathsf{T}}$$

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

$$\mathbf{vehicleGoalPosition} = \begin{bmatrix} 10.5 & 37.5 & -38 & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}$$

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

(Q1.1) What is the Jacobian relationship for the Vehicle Position control task? How was the task reference computed?

Solution: Let $\dot{\mathbf{x}} = \mathbf{J}\,\dot{\mathbf{y}}$ be the Jacobian relationship between the task variables, and the control variables. We remind ourselves that the control variables vector is defined such that

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{\mathbf{q}} & {}^{v}\mathbf{v_l} & {}^{v}\mathbf{v_a} \end{bmatrix}^{\mathsf{T}},$$

where $\dot{\mathbf{q}} \in \mathbb{R}_{1\times 7}$, ${}^{v}\mathbf{v_l}, {}^{v}\mathbf{v_a} \in \mathbb{R}_{1\times 3}$.

The Vehicle Position control task computes the control variables separately for linear and angular velocities.

Thus, the jacobians for the Vehicle Position control task are

$$\begin{aligned} {}^{w}\mathbf{J}_{\mathbf{v_l}} &= \begin{bmatrix} \mathbf{0}_{3\times 7} & {}^{w}_{v}\mathbf{R} & \mathbf{0}_{3\times 3} \end{bmatrix} \\ {}^{w}\mathbf{J}_{\mathbf{v_a}} &= \begin{bmatrix} \mathbf{0}_{3\times 7} & \mathbf{0}_{3\times 3} & {}^{w}_{v}\mathbf{R} \end{bmatrix}. \end{aligned}$$

The references rates $\dot{\bar{\mathbf{x}}}_{\mathbf{v_l}}$ and $\dot{\bar{\mathbf{x}}}_{\mathbf{v_a}}$ are defined as

$$\begin{aligned} &{}^{w}\dot{\bar{\mathbf{x}}}_{\mathbf{v_l}} = \lambda\left({}^{w}_{v}\mathbf{R}_{\text{goal}} - {}^{w}_{v}\mathbf{R}\right) \\ &{}^{w}\dot{\bar{\mathbf{x}}}_{\mathbf{v_a}} = \lambda\left(\text{VersorLemma}({}^{w}_{v}\mathbf{R}_{\text{goal}}, {}^{w}_{v}\mathbf{R})\right) \\ &\lambda \in \mathbb{R}^{+}, \end{aligned}$$

where VersorLemma is a function computing the misalignement vector between two orientations matrixes by using the unit vector lemma and λ is a positive arbitrary real.

Both references rates are then saturated.

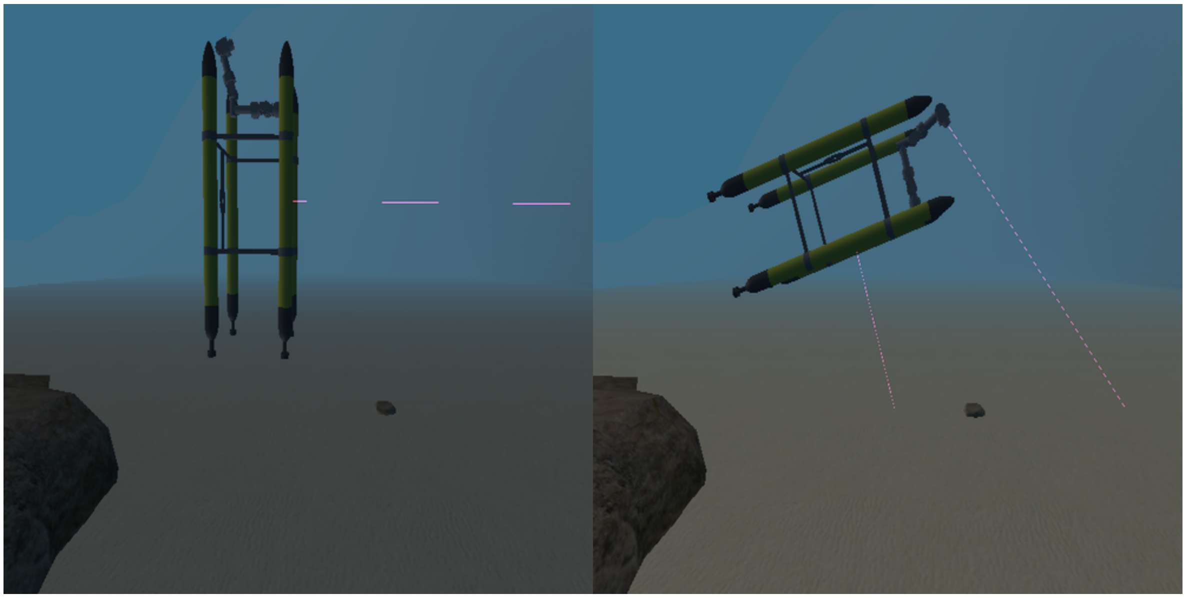
(Q1.2) What is the behaviour if the Horizontal Attitude is enabled or not? Try changing the initial or target orientation in terms of roll and pitch angles. Discuss the behaviour.

Solution: For the given initialization, the Horizontal Attitude task did not activate. We modified the target orientation to $\left(0, -\frac{\pi}{2}, 0\right)$.

Enabling the Horizontal Attitude task with a higher priority than the Vehicle Position task (for angular velocities) severely restricts the manifold of solutions computed. The restriction is illustrated in Figure 1, where the task is disabled, then enabled for the same orientation goal.

Without the Horizontal Attitude task, the vehicle successfully orients itself according to the goal, as demonstrated in Figure 1a.

With the Horizontal Attitude task enabled however, the vehicle seems to be stuck in a compromise between the two tasks where neither one is fully satisfied, as seen in Figure 1b. The reason being, higher priority tasks partially fix the control variables, such that subsequent



(a) Horizontal Attitude disabled

(b) Horizontal Attitude enabled

Figure 1: Vehicle angular velocities (Goal orientation $\left(0, -\frac{\pi}{2}, 0\right)$)

tasks only optimize the remaining arbitrariness and not the control variables themselves. In practice, that leads the lower priority tasks to be considered as a preference as not a requirement.

(Q1.3) Swap the priorities between Horizontal Attitude and the Vehicle Position control task. Discuss the behaviour.

Solution: When exchanging priorities, the Vehicle Position control task is always enabled, as it has the highest priority.

The behaviour is similar to the previous situation, where the Horizontal Attitude task was disabled (see Figure 1a), the Vehicle Position control task is fully satisfied, while the Horizontal Attitude task is never satisfied.

Previously, when both tasks were activated but their priorities reversed, both tasks were partially satisfied. However, in the situation where the Vehicle Position control task has a higher priority, then it supersedes the Horizontal Attitude task.

Probably, the manifold of solutions first computed by the Horizontal Attitude task has considerably more leeway in terms of optimization, which leads the subsequent task to be partially fulfilled. The Vehicle Position control task does not have the same leeway, which leads to its dominance.

1.2 Adding a safety minimum altitude control objective

Q2. Initialize the vehicle at the position:

$$\mathbf{p} = \begin{bmatrix} 48.5 & 11.5 & -33 & 0 & 0 & -\pi/2 \end{bmatrix}^\mathsf{T}$$

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

$$\mathbf{vehicleGoalPosition} = \begin{bmatrix} 50 & -12.5 & -33 & 0 & 0 & -\pi/2 \end{bmatrix}^\mathsf{T}$$

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

(Q2.1) Report the new hierarchy of tasks of the Safe Waypoint Navigation and their priorities. Comment how you choose the priority level for the minimum altitude.

Solution: The hierarchy of tasks for the Safe Waypoint Navigation is described in Table 1, where the minimum altitude task has the highest priority.

During the Safe Waypoint Navigation, the minimum altitude task must be activated whenever necessary, that is when the vehicle's altitude goes over an arbitrary threshold. When activated, its directives must supersede other possibly conflicting tasks, otherwise the vehicle might accidentally collide with the ground, which is undesired.

Table 1: Hierarchy of tasks for Safe Waypoint Navigation - ROBUST

Task	Type	Safe Waypoint Navigation
Minimum altitude	I	1
Horizontal attitude	I	2
Vehicle position	I	3

(Q2.2) What is the Jacobian relationship for the Minimum Attitude control task? Report the formula for the desired task reference generation, and the activation thresholds.

Solution: We consider the actual vehicle altitude a, and the distance vector given by the sensor $\mathbf{d}$. Let ${}^{v}\mathbf{n}$ be the seafloor normal vector projected on the vehicle frame $\langle v \rangle$. The vehicle altitude is considered as

$$a = {}^{v}\mathbf{n} \cdot {}^{v}\mathbf{d},$$

where the vector $\mathbf{d}$ is projected on the vehicle frame.

Consequently, if we assume the seafloor is flat, then we can write that

$$\dot{a} = {}^{v}\mathbf{n} \cdot \mathbf{v}_{v/w},$$

where $\mathbf{v}_{v/w}$ is the linear velocity vector of the vehicle frame $\langle v \rangle$ with respect to the world frame $\langle w \rangle$.

Thus, we can write the jacobian for the minimum altitude control task, that is

$${}^{v}\mathbf{J}_{\mathbf{ma}} = \begin{bmatrix} \mathbf{0}_{1\times 7} & {}^{v}\mathbf{n}^{\mathsf{T}} & \mathbf{0}_{1\times 3} \end{bmatrix}.$$

The desired task reference is generated using the following formula

$${}^{v}\dot{\bar{\mathbf{x}}}_{\mathbf{ma}} = \lambda(\mathbf{ma} - \mathbf{a}) + 1 \quad \lambda \in \mathbb{R}^{+}.$$

We introduce a bias of 1, to prevent the reference rate from converging to small values. Indeed, when the vehicle altitude is close to its objective, the reference rate might become insignificant, which might endanger the vehicle's safety, as the task can be fully activated whilst not influencing the control variables to the full extent it ought to.

The activation thresholds are $[\mathbf{ma}; \mathbf{ma} + 0.5]$. If the minimum altitude requirement must be very strict, then a safety buffer can be added to the interval, as to activate the task earlier.

(Q2.3) Try imposing a minimum altitude of 1, 5, 10 m respectively. What is the behaviour? Does the vehicle reach its final goal in all cases?

Solution: For the 1 m minimum altitude task, the vehicle successfully arrives at its goal destination, whilst doing its best to respect the minimum altitude task. When 5 or 10 m is imposed, the minimum altitude task conflicts with the vehicle position task and the latter is superseded in terms of altitude due to the priorities.

If the imposed minimum altitude is 1 m, then the vehicle immediately moves towards its goal, as the minimum altitude task is not activated in the first few seconds.

If the imposed minimum altitude is either 5 or 10 m, then the minimum altitude task is immediately activated, leading to the vehicle increasing its altitude until the minimum altitude has been reached. Other active tasks run in parallel.

(Q2.4) How was the sensor distance processed to obtain the altitude measurement? Does it work in all cases or some underlying assumptions are implicitly made?

Solution: The altitude measurement is obtained by the dot product of the seafloor normal vector and the sensor distance $\mathbf{d}$. This method does not work in all cases, as the sensor could point towards an irregular seafloor. That is, if the vehicle is above a slope and/or tilted, then the process could either return a lower or higher altitude than expected.

2 Exercise 2: Implement a Basic "Landing" Action.

2.1 Adding an altitude control objective

Q3. Initialize the vehicle at the position:

$$\mathbf{p} = \begin{bmatrix} 10.5 & 37.5 & -38 & 0 & -0.06 & 0.5 \end{bmatrix}^\mathsf{T}$$

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

(Q3.1) Report the hierarchy of task used and their priorities to implement the Landing Action. Comment how you choose the priority level for the altitude control task.

> Solution: The hierarchy of tasks for the Landing is described in Table 2, where the horizontal attitude task has the highest priority.
>
> Given the vehicle, the landing is safer to perform if the vehicle is horizontal to the seafloor, when it is assumed to be relatively flat. The horizontal attitude is a prerequisite for the landing action.

Table 2: Hierarchy of tasks for Landing - ROBUST

Task	Type	Landing
Horizontal attitude	I	1
Altitude	E	2

(Q3.2) What is the Jacobian relationship for the Altitude control task? How was the task reference computed?

> Solution: The Jacobian relationship for the altitude control task is identical to the minimum altitude task.
>
> $${}^{v}\mathbf{J}_{\mathbf{alt}} = \begin{bmatrix} \mathbf{0}_{1\times 7} & {}^{v}\mathbf{n}^\mathsf{T} & \mathbf{0}_{1\times 3} \end{bmatrix}.$$
>
> The task reference is equally similar to the minimum altitude task, except the objective altitude is in this instance 0.
>
> $${}^{v}\dot{\bar{\mathbf{x}}}_{\mathbf{alt}} = \lambda(0 - \mathbf{a}) \quad \lambda \in \mathbb{R}^{+}.$$

(Q3.3) How does this task differs from a minimum altitude control task?

> Solution: This task differs from the minimum altitude control task in two points:
>
> - The objective altitude is 0;
> - The altitude task has a lower priority than the horizontal attitude task.

2.2 Adding mission phases and change of action

Q4. Initialize the vehicle at the position:

$$\mathbf{p} = \begin{bmatrix} 8.5 & 38.5 & -36 & 0 & -0.06 & 0.5 \end{bmatrix}^\mathsf{T}$$

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

$$\mathbf{vehicleGoalPosition} = \begin{bmatrix} 10.5 & 37.5 & -38 & 0 & -0.06 & 0.5 \end{bmatrix}^\mathsf{T}$$

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

(Q4.1) Report the unified hierarchy of tasks used and their priorities.

Solution: The unified hierarchy of tasks is illustrated in Table 3. The Table is read left to right, where the Safe Waypoint Navigation is the first action, and the Landing the second action.

Table 3: Hierarchy of tasks for Safe Waypoint Navigation & Landing - ROBUST

Task	Type	Safe Waypoint Navigation	Landing
Minimum altitude	I	1	
Horizontal attitude	I	2	1
Vehicle position	I	3	
Altitude	E		2

(Q4.2) How did you implement the transition from one action to the other?

Solution: In order to transition from one action to another, a trigger condition must be met, which can simply be considered as the completion of the current action's objective. In practice, we compute an error metric of the current parameters against the objective parameters and check during the simulation main loop whether the error is insignificant enough to consider the current action as completed.

In order to smoothly activate (or deactivate) tasks between different actions, we sequence all tasks using a new activation function $a^p\left(\mathbf{p}\right)$, such that the modified activation function is defined as

$$a(\mathbf{x}, \mathbf{p}) = a^i(\mathbf{x}) a^p\left(\mathbf{p}\right),$$

where $\mathbf{p}$ is a vector of variables external to the control variables $\mathbf{x}$, conveniently parameterized by the time elapsed in the current action in order to achieve a smooth transition, e.g. a continuous increasing/decreasing function between 0 and 1.

Once the trigger condition is met for the current action, the sequencing activation functions $a^p\left(\mathbf{p}\right)$ are enabled for the transition to the next action, which starts the smooth transition to the new task priorities.

3 Exercise 3: Improve the "Landing" Action

3.1 Adding an alignment to target control objective

Q5. If we use the landing action, there is no guarantee that we land in from 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:

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

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

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

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 $\langle v \rangle$ to the nodule frame $\langle t \rangle$.

(Q5.1) Report the hierarchy of tasks used and their priorities in each action. Comment the behaviour.

Solution: The hierarchy of tasks used is described in Table 4.

The Alignement to Target task is also a prerequisite for the landing task, along with the horizontal attitude. However, its priority is lower than the Horizontal attitude task as the vehicle's attitude is a matter of safety.

During the Landing phase, the vehicle first aligns itself in terms of position, then in orientation. When orienting itself, the descent begins at the same time. This behaviour matches the task priorities, as linear velocities are first generated for the Alignment to Target, then for the Altitude.

Table 4: Hierarchy of tasks for Safe Waypoint Navigation & Improved Landing

Task	Type	Safe Waypoint Navigation	Improved Landing
Minimum altitude	I	1	
Horizontal attitude	I	2	1
Alignement to Target	I		2
Vehicle position	I	3	
Altitude	E		3

(Q5.2) What is the Jacobian relationship for the Alignment to Target control task? How was the task reference computed?

Solution:

Let $\mathbf{O_v}$ and $\mathbf{O_t}$ be the respective origins of the vehicle frame $\langle v \rangle$ and target frame $\langle t \rangle$. Let $\mathbf{i_v}$ be the unit vector of the x-axis of the vehicle, and θ the positive angle between ${}^w\mathbf{i_v}$ and the projected distance vector ${}^w\mathbf{d} = {}^w\mathbf{O_t} - {}^w\mathbf{O_v}$ on the inertial horizontal plane. Figure 2 illustrates this task.

In order to align the vehicle to the target, we define the misalignement vector ${}^w\boldsymbol{\rho} = \theta\, {}^w\mathbf{n_{at}}$, where ${}^w\mathbf{n_{at}}$ is a vector along the axis of rotation such that ${}^w\mathbf{n_{at}} = \frac{1}{\sin\theta}\, {}^w\mathbf{i_v} \wedge \left(\frac{{}^w\mathbf{d}}{\|{}^w\mathbf{d}\|}\right)$.

In order to align the vehicle, we define our task with the objective of minimising $\boldsymbol{\rho}$ using its derivative w.r.t. θ. The partial derivative is

$$\frac{\partial\, {}^w\boldsymbol{\rho}}{\partial \theta} = {}^w\mathbf{n_{at}}\dot{\theta}.$$

Let us express the above expression in terms of $\dot{\mathbf{y}}$. First, for $\dot{\theta}$

$$\begin{aligned}\dot{\theta} &= {}^w\mathbf{n_{at}}\, \boldsymbol{\omega}_{t/x} \\ &= {}^w\mathbf{n_{at}} \left(\boldsymbol{\omega}_{t/w} - \boldsymbol{\omega}_{x/w}\right)\end{aligned}.$$

${}^w\mathbf{i_v}$ is a unit vector of the x-axis of the vehicle frame $\langle v \rangle$, thus we can write $\boldsymbol{\omega}_{v/w} = \boldsymbol{\omega}_{x/w}$. Next, for $\boldsymbol{\omega}_{t/w}$

$$\begin{aligned}\boldsymbol{\omega}_{t/w} &= \frac{1}{\|{}^w\mathbf{d}\|^2}\left({}^w\mathbf{d} \wedge \mathbf{v}_{d/w}\right) \\ &= -\frac{1}{\|{}^w\mathbf{d}\|^2}\left({}^w\mathbf{d} \wedge \mathbf{v}_{v/w}\right)\end{aligned},$$

where $\mathbf{v}_{d/w} = \mathbf{v}_{t/w} - \mathbf{v}_{v/w}$ by definition. We assume the target is immobile, thus $\mathbf{v}_{t/w} = 0$. Finally, the Jacobian relationship is

$$\dot{\theta} = {}^w\mathbf{n_{at}}^{\mathsf{T}} \begin{bmatrix}\mathbf{0}_{3\times 7} & -\frac{1}{\|{}^w\mathbf{d}\|^2}\left[{}^w\mathbf{d}\wedge\right] & -\mathbf{I}_{3\times 3}\end{bmatrix} \dot{\mathbf{y}} = {}^w\mathbf{J_{at}}\, \dot{\mathbf{y}}.$$

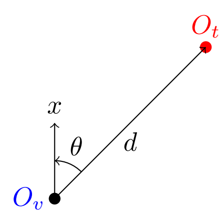


Figure 2: Target alignement on the inertial horizontal plane

Solution: We compute the misalignement vector $\boldsymbol{\rho}$ using the Reduced Versor Lemma. The task reference is computed w.r.t. ${}^w\boldsymbol{\rho}_{\mathbf{at}}$, as when the vectors are aligned, $\|{}^w\boldsymbol{\rho}_{\mathbf{at}}\|$ tends toward 0. We have

$${}^w\dot{\bar{\mathbf{x}}}_{\mathbf{at}} = \lambda\left(0 - \|{}^w\boldsymbol{\rho}_{\mathbf{at}}\|\right) \quad \lambda \in \mathbb{R}^+.$$

(Q5.3) Try changing the gain of the alignment task. Try at least three different values, where one is very small. What is the observed behaviour? Could you devise a solution that is gain-independent guaranteeing that the landing is accomplished aligned to the target?

Solution: The smaller the gain, the slower the vehicle orients itself to the target. However, regardless of the gain, the vehicles changes its position in order to align to the target, albeit slower the smaller the gain. For a very small gain, the vehicle does not seem to orient itself at all.

In order to have a gain-independent behaviour, we can add a bias term to the task reference.

$${}^w\dot{\bar{\mathbf{x}}}_{\mathbf{at}} = \lambda\left(0 - \|{}^w\boldsymbol{\rho}_{\mathbf{at}}\|\right) - \mu \quad \lambda, \mu \in \mathbb{R}^+.$$

(Q5.4) 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. If, after landing, the nodule is not in the manipulator's workspace, what was missing in the previous actions? How would you fix it?

Solution: After landing, the end-effector moves towards the target, however the distance to the nodule is insufficient to reach it with the end-effector. The end-effector stays elongated, but does not touch the nodule.

If after landing, the target is not within the manipulator's workspace, the TKIP is missing a task involving the distance to the target.

To fix the Landing action, we would add a new inequality task called Distance to Target, which makes use of the distance vector ${}^{w}\mathbf{d}$ to reach the objective distance value. The new hierarchy of tasks is described in Table 5, where the Distance to Target task priority is below both Alignement to Target and Horizontal attitude.

Table 5: Hierarchy of tasks for Safe Waypoint Navigation & Improved Landing v2 - ROBUST

Task	Type	Safe Waypoint Navigation	Improved Landing v2
Minimum altitude	I	1	
Horizontal attitude	I	2	1
Alignement to Target	I		2
Distance to Target	I		3
Vehicle position	I	3	
Altitude	E		4

4 Exercise 4: Implementing a Fixed-base Manipulation Action

4.1 Adding non-reactive tasks

Q6. 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.

(Q6.1) Report the hierarchy of tasks used and their priorities in each action. At which priority level did you add the constraint task?

> Solution: The hierarchy of tasks used is described in Table 6.
> The vehicle null velocity task is added as the top-priority task for the Manipulation action, in order for subsequent tasks to account for the constraint.

Table 6: Hierarchy of tasks for Safe Waypoint Navigation & Improved Landing & Manipulation - ROBUST

Task	Type	Safe Waypoint Navigation	Improved Landing	Manipulation
Minimum altitude	I	1		
Horizontal attitude	I	2	1	
Alignement to Target	I		2	
Vehicle null velocity	E			1
Vehicle position	I	3		
Altitude	E		3	
End-effector position	I			2

(Q6.2) What is the jacobian relationship for the vehicle null velocity task? How was the task reference computed?

> Solution: We consider simply the Jacobian ${}^{v}\mathbf{J}_{\mathbf{vnv}}$ as
>
> $${}^{v}\mathbf{J}_{\mathbf{vnv}} = \begin{bmatrix} \mathbf{0}_{6\times7} & \mathbf{I}_{6\times6} \end{bmatrix},$$
>
> which corresponds to the end-effector joints (zeros block) and vehicles velocities (ones block).
> The task reference corresponds to the objective vehicle velocities, that is
>
> $${}^{v}\dot{\bar{\mathbf{x}}}_{\mathbf{vnv}} = \begin{bmatrix} \mathbf{0}_{6\times1} \end{bmatrix}.$$

4.2 Adding a joint limit task

Q7. 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.

(Q7.1) Report the hierarchy of tasks used and their priorities in each action. At which priority level did you add the joint limits task?

> Solution: The hierarchy of tasks used is described in Table 7 The Joint limits safety task is added below the vehicle null velocity constraint task, and before the end-effector position action-defining task.

Table 7: Hierarchy of tasks for Safe Waypoint Navigation & Improved Landing & Manipulation v2 - ROBUST

Task	Type	Safe Waypoint Navigation	Improved Landing	Manipulation v2
Vehicle null velocity	E			1
Minimum altitude	I	1		
Horizontal attitude	I	2	1	
Joint limits	I			2
Alignement to Target	I		2	
Vehicle position	I	3		
Altitude	E		3	
End-effector position	E			3

(Q7.2) What is the Jacobian relationship for the Joint Limits task? How was the task reference computed?

Solution: We consider simply the Jacobian $\mathbf{J}_{\mathbf{jl}}$ as

$$\mathbf{J}_{\mathbf{jl}} = \begin{bmatrix} \mathbf{I}_{7\times 7} & \mathbf{0}_{6\times 6} \end{bmatrix}.$$

We want the desired task reference to be a value away from the limits $\mathbf{uvms.jlmin}$ and $\mathbf{uvms.jlmax}$. We choose $\dot{\bar{\mathbf{x}}}_{\mathbf{jl}}$ to reach for the mean value, a center position,

$$\dot{\bar{\mathbf{x}}}_{\mathbf{jl}} = \lambda \left(\frac{\mathbf{uvms.jlmin} + \mathbf{uvms.jlmax}}{2} - \mathbf{q} \right) \quad \lambda \in \mathbb{R}^+,$$

where $\mathbf{q}$ are the current joint values.

5 Exercise 5: Floating Manipulation

Use the DexROV simulation for this exercise.

5.1 Adding mission phases

Q8. 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.

(Q8.1) Report the unified hierarchy of tasks used and their priorities. Which task is active in which phase/action?

> Solution: The unified hierarchy of tasks used is described in Table 8.
> For the Safe Waypoint Navigation, only two tasks are active, i.e. Horizontal Attitude and Vehicle position. For the Floating Manipulation, four tasks are active, i.e. Horizontal Attitude, Joint limits, Dexterity and End-effector position.

Table 8: Hierarchy of tasks for Safe Waypoint Navigation & Floating Manipulation - DexROV

Task	Type	Safe Waypoint Navigation	Floating Manipulation
Horizontal attitude	I	1	1
Joint limits	I		2
Dexterity	I		3
Vehicle position	I	2	
End-effector position	E		4

(Q8.2) What is the difference if, from the very beginning, you use the action of floating manipulation (i.e. just a single action)?

> Solution: If the mission starts from the Floating Manipulation, instead of the Safe Waypoint Navigation, then the arm moves during the translation from the initial position to the objective position. The arm movements influence the vehicle from the beginning, which means some safety tasks might not be fully satified, e.g. Horizontal Attitude.

5.2 Adding an optimization control objective

Q9. 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

$$\mathbf{q_{pas}} = \begin{bmatrix} -0.0031 & 1.2586 & 0.0128 & -1.2460 & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}$$

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

(Q9.1) Report the hierarchy of tasks used and their priorities in each action. At which priority level did you add the optimization task?

> Solution: The unified hierarchy of tasks used is described in Table 9.
> The optimization task is added to the bottom priority level.

Table 9: Hierarchy of tasks for Safe Waypoint Navigation & Floating Manipulation v2 - DexROV

Task	Type	Safe Waypoint Navigation	Floating Manipulation v2
Horizontal attitude	I	1	1
Joint limits	I		2
Dexterity	I		3
Vehicle position	I	2	
End-effector position	E		4
Preferred arm shape	I		5

(Q9.2) What is the Jacobian relationship for the Joint Preferred Shape task? How was the task reference computed?

Solution: The Jacobian $\mathbf{J}_{\mathbf{pas}}$ is identical to the Joint limits' Jacobian, i.e.

$$\mathbf{J}_{\mathbf{pas}} = \begin{bmatrix} \mathbf{I}_{7\times 7} & \mathbf{0}_{6\times 6} \end{bmatrix}.$$

The desired references values correpond to the preferred arm shape,

$$\dot{\bar{\mathbf{x}}}_{\mathbf{pas}} = \lambda \left(\mathbf{q}_{\mathbf{pas}} - \mathbf{q} \right) \quad \lambda \in \mathbb{R}^{+},$$

(Q9.3) What is the difference between having or not having this objective?

Solution: Given the preferred arm shape, the difference shows not only in the final position and orientation of the vehicle, but also in the dexterity of the arm. Comparing the manipulability with and without the preferred arm shape optimization task, it is obvious that the task improves the manipulability of the arm over time, which reduces the risk of reaching a kinematic singularity.

On Figure 3, we can see that the manipulability measure reaches much lower values during the Floating Manipulation task around 30s without the optimization task.

Equally, the optimization task affects the paths chosen by the TPIK algorithm, leading the vehicle to move differently with the task active than without in order to keep the preferred arm shape.

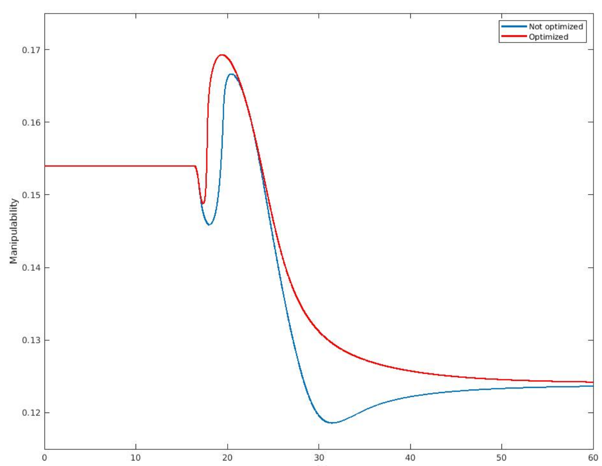


Figure 3: Manipulability measure with and without preferred arm shape task

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

6.1 Adding the parallel arm-vehicle coordination scheme

Q10. Let us now see how the two different subsystems (arm and vehicle) can be properly coordinate. Introduce in the simulation a sinusoidal linear velocity disturbance acting on the vehicle (along a constant inertial x-y direction of your choice), 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.

(Q10.1) Which tasks did you introduce to implement the parallel coordination scheme?

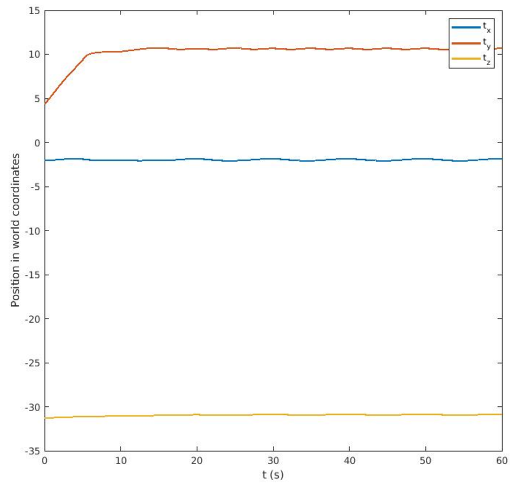
> Solution: The TPIK is duplicated so that the second TPIK optimizes the arm velocities for the current reference vehicle velocities, from the first TPIK.
>
> In order to implement the parallel coordination scheme, the vehicle constrained velocity task is introduced at the top priority level of the second TPIK.

(Q10.2) Show the plot of the position of the end-effector, showing that it is constant. Show also a plot of the velocities of the vehicle and of the arm.

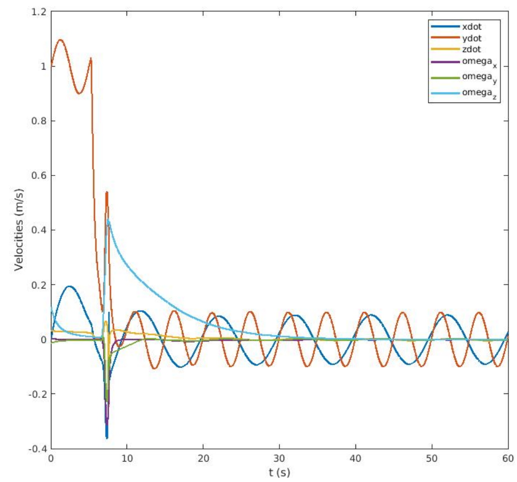
> Solution: The plots are shown in Figure 4.

(Q10.3) What happens if the sinusoidal disturbance becomes too big? Try increasing the saturation value of the end-effector task if it is too low.

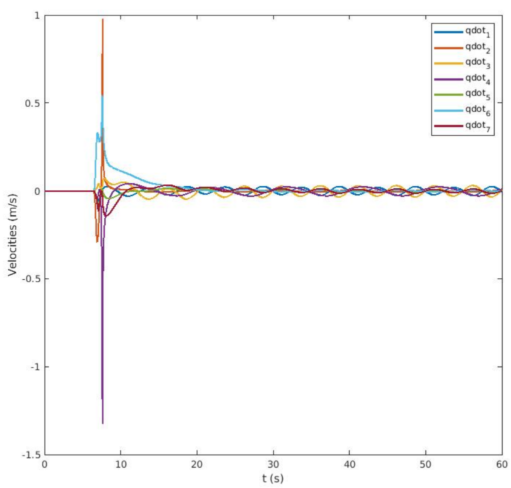
> Solution: If the sinusoidal disturbance is too big, the end-effector position becomes variable, the arm tries to reach for the objective position when the vehicle comes near it.
>
> Despite increasing the saturation value, if the disturbance is too big, the TPIK cannot stabilise the end-effector position along the axis of the disturbance.
>
> The plots in question Q10.2 have been updated in Figure 5 to show the consequences of a big disturbance.



(a) End-effector position in world frame $\langle w \rangle$

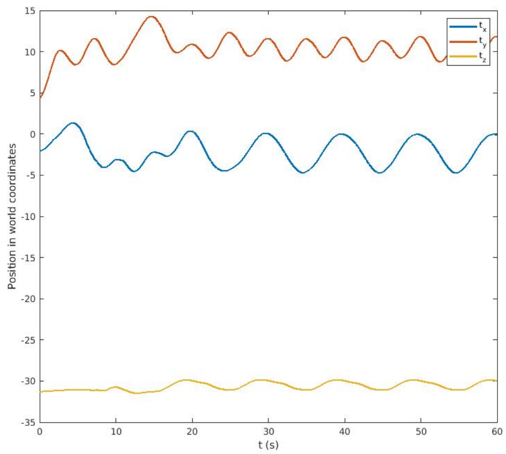


(b) Vehicle velocities

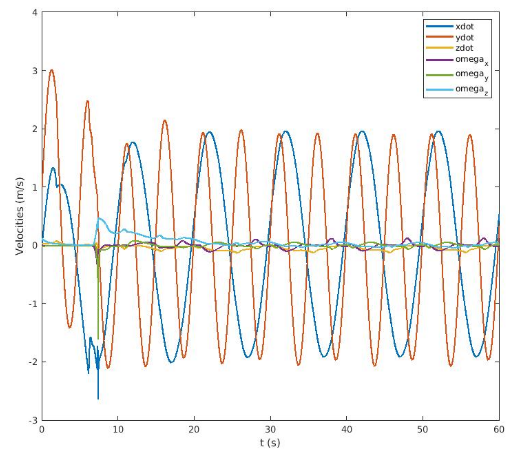


(c) Arm velocities

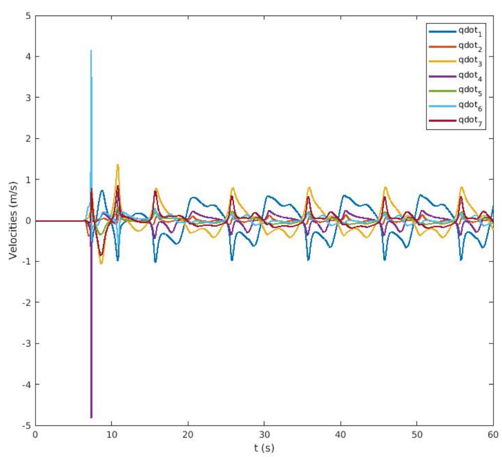
Figure 4: End-effector position, vehicles and arm velocities with linear velocity disturbance along a xy direction



(a) End-effector position in world frame $\langle w \rangle$



(b) Vehicle velocities



(c) Arm velocities

Figure 5: End-effector position, vehicles and arm velocities with bigger linear velocity disturbance along a xy direction