Exercise 1

Goal: Implement a vehicle position control task, and test that the vehicle reaches the required position. Add the proper Jacobian variable, its activation function, and the desired reference rate.

To test that the task is working properly

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

eta = [10.5 35.5 -36 0 0 pi/2] (x,y,z,roll,pitch,yaw)

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

w_vehicle_goal_position = [10.5 37.5 -38]

Exercise 2

Goal: Implement a horizontal attitude control task, and test that the vehicle reaches the required orientation. Add the proper Jacobian variable, its activation function, and the desired reference rate.

To test that the task is working properly

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

eta = [10.5 35.5 -36 -pi/3 pi/3 pi/2] (x,y,z,roll,pitch,yaw)

and then require that the norm of the misalignment between the vehicle's z vector and the absolute z-axis is less than 0.2

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

[10.5 35.5 -36 0 0 pi/2] (x,y,z,roll,pitch,yaw)

Goal: Implement a vehicle altitude control task, adding the proper Jacobian variable, its activation function, and the desired reference rate.

Test the functioning of the position/attitude vehicle control tasks together with the minimum altitude in two situations:

1) one where the desired position is too close to seafloor, for example

```
[12.2025   37.3748  -39.8860]
```

2) one where the desired position is instead far away from the seafloor, for example

[10.5 37.5 -38]

Exercise 4

Goal: implement two actions, extending the ActionManager to manage transition between actions. Modify the class to associate a name to the action in the addAction, and to select the action with the same name using setCurrentAction. Add an API to add the unifying task list (computed manually). When the computeICAT is called, automatically compute the activation functions to manage the transition between the current and previous action (hint: the function ismember **can be very useful). To this aim, the ActionManager should keep track of the time since the new action was scheduled (modify as necessary).**

To test, implement the following actions:

- *Action "safe navigation" includes minimum altitude, horizontal attitude, and vehicle position and attitude control*
- *Action "landing" includes altitude control (to zero), horizontal attitude, and vehicle position.*

Start by moving from an initial point is close to the seafloor, e.g.

```
[48.5 11.5 -33]
```

And move the vehicle towards a position such as

```
[50 -12.5 -33]
```

Then land on the seafloor. Include the appropriate transition between the two actions. Show that the minimum altitude gets activated during the safe navigation (eventually change the threshold values).