A picture containing logo

Description automatically generated

**Università degli Studi di Genova**

**CORSO DI LAUREA MAGISTRALE**

**Robotics Engineering**

**COOPERATIVE ROBOTICS**

Authors:

Giacomo Lugano

Youssef Attia

Hussein Fouad

Email:

[jek.lugano@yahoo.com](mailto:jek.lugano@yahoo.com)

[Youssef-attia@live.com](mailto:Youssef-attia@live.com)

[hussienfouad90@gmail.com](mailto:hussienfouad90@gmail.com)

Date: 02/02/2023

# Robust Nodule Inspection Mission

1. Tasks unified hierarchy

As shown from the table below the Task Priority Control hierarchy is composed of three actions. Firstly, the vehicle is spawned in:

and navigates to:

keeping into account a minimum altitude to the see bed “assuming it’s flat” of 1 meter and maintaining a proper pitch, roll and yaw presentenced in the so-called vehicle horizontal attitude control task.

Secondly, the landing action starts in which the vehicle lands to the seabed and concurrently “with higher priority” aligned itself to the nodule to be inspected and keeping a desired distance from the nodule.

Lastly, the inspection actions start maintaining the vehicle position in the so-called Vehicle null velocity task and guiding the end-effector to the nodule keeping into account “with higher priority” the arm joint limits and manipulability to avoid singularity. Furthermore, the vehicle null velocity task is crucial in this action to avoid the vehicle motion due to the forces and moments generated by the end effector motion.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Type | Navigate to goal | Landing | Inspection |
| Arm Joints Limits | I |  |  | 1 |
| Arm Manipulability | I |  |  | 2 |
| Vehicle Null Velocity | E |  |  | 3 |
| End Effector Linear Position | E |  |  | 4 |
| End Effector Angular Position | E |  |  | 5 |
| Vehicle minimum altitude | I | 1 |  |  |
| Vehicle horizontal attitude | I | 2 | 1 |  |
| Vehicle heading control | I | 3 |  |  |
| Vehicle position control | E | 4 |  |  |
| Vehicle alignment to nodule | I |  | 2 |  |
| Vehicle distance to nodule | I |  | 3 |  |
| Vehicle altitude control | E |  | 4 |  |

1. Robot’s behavior

As per the discussion of the robot’s task priority control hierarchy in section I.1. This section will discuss the robot behavior proving it is following the proposed hierarchy.

Starting from the vehicle altitude and attitude throughout its mission which is composed of the three actions. It can be seen from the following graph [Fig.1] of the vehicle altitude that as the vehicle is spawned it has relatively high altitude and moving to the goal with the action *Navigate to waypoint* the altitude starts converging to the goal height. However, the *Vehicle minimum altitude* task should prevent the vehicle from going beneath 1 meter of altitude, but it is not the case here as the goal height is higher than task’s threshold. Moreover, as the vehicle have reached its goal and action *Landing* takes over the altitude starts converging to zero and stays there due to the task “Vehicle null velocity” in action *Inspection*.

Similar behavior can be seen in the graph of the vehicle attitude for the misalignment that is used for the *Horizontal attitude* task. However, a small change can be seen between the transition of action *Landing* to action *Inspection*. This is due to the relatively long period of transition between the two actions.

A picture containing chart

Description automatically generated

Figure 1:Vehicle Altitude and Attitude

Moreover, the following graphs in Fig.2 shows how the action *Navigate to waypoint* takes effect. In the beginning and as the simulation starts the vehicle is spawned away from the goal. And, as the action starts is can be seen that the distance vector starts to converge to zero and same goes to the misalignment vector of the vehicle to the goal. As the Vehicle reaches its goal, the action *Landing* two takes over and the vehicle goes further from the original goal. It can be noted here that the change is mainly on the Z-axis as the vehicle is Landing.

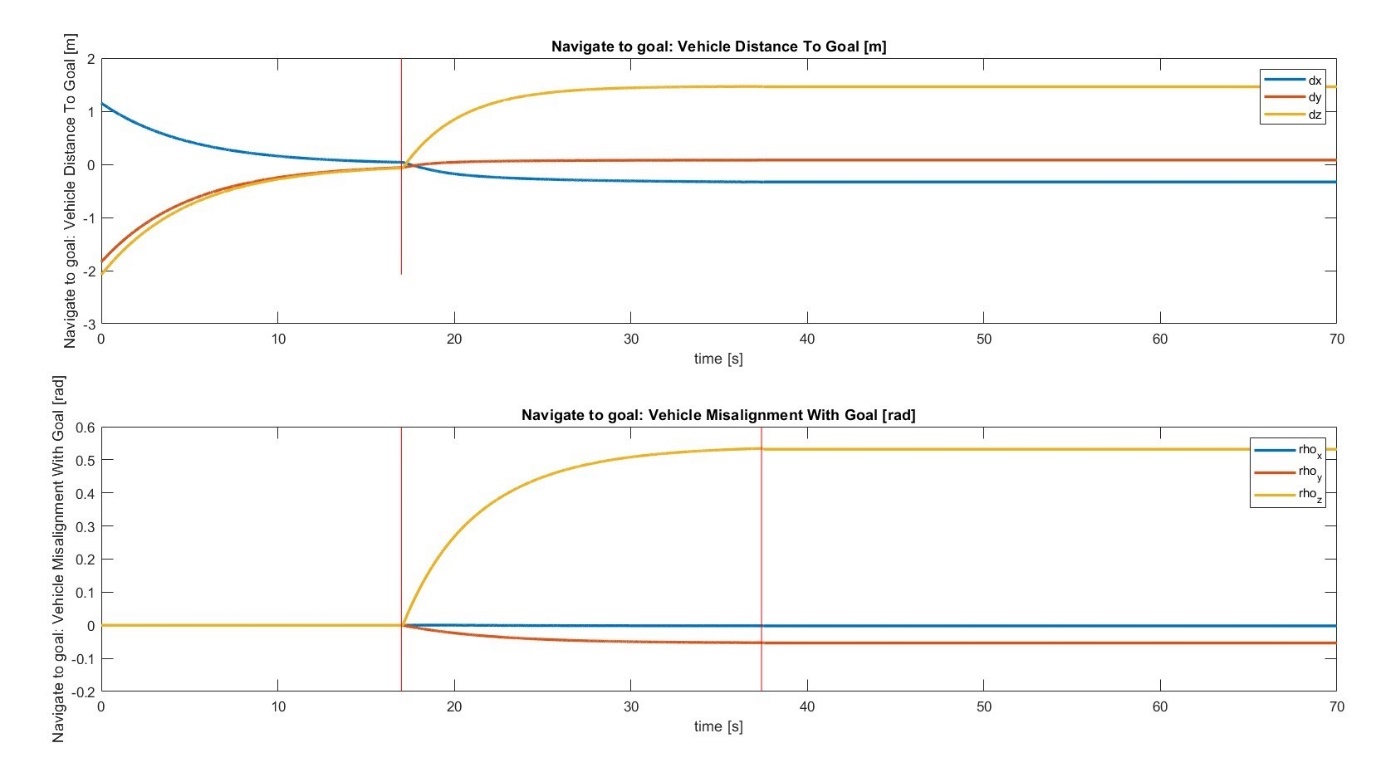


Figure 2: Vehicle distance and misalignment to goal

Moving to the action *Landing* the main goal here is to land making sure that the vehicle is aligned to the nodule and that nodule is in the manipulator’s workspace. Furthermore, the robot’s behavior in this case can be studied from the following graphs in Fig.3 showing the distance between the vehicle and the nodule and also the misalignment vector between the vehicle and the nodule. It can be noted here that after the action *Navigate to waypoint* has ended the vehicle is not aligned to the nodule and as the action *Landing* aligns the vehicle the misalignment vector converges to zero. Increasing the gain makes the vehicle align faster and vice versa. This is the task *Vehicle longitudinal alignment* *to nodule* is further discussed in section 3 and section 4.

Diagram

Description automatically generated with medium confidence

Figure 3: Vehicle distance and misalignment to nodule

Lastly, from *Inspection* action point of view both the Arm distance and misalignment to the nodule should be studied, and it can be seen in Figure 4 that as soon as the *Inspection* action takes place both the distance and the misalignment start to converge to zero and they arrive to zero around t = 60 seconds and this can be considered the end of the mission.

Diagram

Description automatically generated

Figure 4: Arm distance and misalignment to nodule

1. Vehicle alignment to nodule task Jacobian

This objective requires that the vehicle aligns to a nodule in the space, the nodule, but only in the horizontal part, as it can be seen from the figure below:

Graphical user interface, application, Word

Description automatically generated Graphical user interface, application, Word

Description automatically generated

Figure 5: Faulty misalignment evaluation

Figure 6: Proper misalignment evaluation

In other word the following equality objective must be satisfied:

Where is the misalignment vector between the vehicle -unit vector and the projection of the distance vector on the world horizontal plane.

As it can be seen, this task is structured as an equality task and acquire the attribute of operational prerequisite:

= desired misalignment

Therefore, the activation function will be an identity matrix, and since it is desired to control three scalars (the component of the error along x, y, z), of dimension 3x3:

From here the evaluation of the task Jacobian can take place, let’s start by evaluating the distance vector between the vehicle position and the nodule , which can be called .

Then project such vector on the world horizontal plane as:

Where is the vertical unit vector of the world reference frame, the identity matrix.

Thus, the problem would be solved if the -unit vector of the vehicle reference frame align to unit vector , let’s consider the misalignment vector between the two, computed with the reduced versor lemma:

To achieve that the two-unit vector align it is needed to find the relationship between how evolves in time and the system control variables .The derivative of with respect to an observer α is given by:

If the observer α is considered to be on one of the two vectors, for example considering it on the vehicle reference frame <v>, the equation simplifies as:

With the goal of expressing such equation in terms of the system control variables , can be expressed as:

Where is the angular velocity between the -unit vector of the vehicle reference frame and the world reference frame and belong to the same class of the vehicle rotational velocity so it can be concluded that:

Concerning it is known that the tip of the vector moves with a velocity which satisfies:

where can be evaluated as the derivative of as:

Where is equal to zero since all its terms are fixed in the world reference frame, obtaining:

In this study case, the nodule is fixed in space, thus the equation simplifies as:

Where is part of the control variables of the vector .

The inverse cross product problem can be solved by finding which is a vector orthogonal to both and whose length is equal to as:

Inserted in the equation of gives:

and by recalling that:

The final relation between the control’s variables and the derivative of can be obtained as:

Therefore, the final task Jacobian can be obtained as:

To solve the problem, it will need to create a task which is able to bring to zero, among many possible choices, one requires the derivative of to be proportional to the misalignment vector itself:

Where is the desired misalignment and it’s equal to zero, if the true derivative follows the desired value, this would guarantee an asymptotic convergence of **.**

1. Alerting gain & observation

In this section it is discussed how the gain of the task *Vehicle alignment to the nodule* takes effect on the robot’s behavior. For this case three different values of gain were chosen to observe the robot’s behavior difference. It was expected that having a lower gain makes the vehicle align to nodule in longer period and vise versa. However, there three gains value that were chosen are 0.01, 0.2 and 0.8. They were chosen like that as the first value is relatively very low and with this gain its expected that the vehicle should not finish it’s alignment even after touch the sea bed. While the second value is usually the same used for most tasks of the vehicle tasks. Lastly, the gain of 0.8 is relatively high value and it is expected that the vehicle will align with nodule even before the landing occurs.

Furthermore, the different values were tested, and the following graphs [Fig.7] was the result. And it can be seen here that the hypothesis of the experiment that was made can be considered as achieved. As for the first gain the vehicle was never able to finish the action *Landing* on the time of the simulation. While for the gain the vehicle was able to finish the action *Landing* and move to the action *Inspection* around t = 35 seconds. Finally for the gain of 0.8 the vehicle finished the action *Landing* around t = 25 seconds.

In conclusion, it can be reported that altering the gain has a directly proportional effect on the time needed for achieving the task and therefore the action. Keep in mind that this test is done on the simulation which has a max time of 70 seconds, this is relatively far from the real-life implementation and the gains must be tuned according to the vehicle specifications.

Diagram

Description automatically generated

Figure 7: Different gains for Vehicle alignment to nodule task

On the other hand, this implementation can not be granted 100% to finish the action of *Landing* on time. A proper solution for this problem is to divide the action of *Landing* into two separate actions in which the vehicle first aligns to the nodule as one action and once done it moves to the second action which is *Landing.* A modification of the proposed hierarchy in section I.1 can be seen in the following table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Type | Navigate to goal | Alignment | Landing | Inspection |
| Arm Joints Limits | I |  |  |  | 1 |
| Arm Manipulability | I |  |  |  | 2 |
| Vehicle Null Velocity | E |  |  |  | 3 |
| End Effector Linear Position | E |  |  |  | 4 |
| End Effector Angular Position | E |  |  |  | 5 |
| Vehicle minimum altitude | I | 1 |  |  |  |
| Vehicle horizontal attitude | I | 2 | 1 | 1 |  |
| Vehicle heading control | I | 3 |  |  |  |
| Vehicle position control | E | 4 |  |  |  |
| Vehicle alignment to nodule | I |  | 2 | 2 |  |
| Vehicle distance to nodule | I |  |  | 3 |  |
| Vehicle altitude control | E |  |  | 4 |  |

Having such a hierarchy ensures that the vehicle will never land without aligning to the nodules prior to the *Landing* action. This is also crucial in real life situation as it was observed in the test of gain 0.01 that the vehicle laned touching the nodule which can cause leakage in an AUV (Autonomous Underwater Vehicle) if there were no obstacle avoidance task with higher priority. Please find the proposed hierarchy solution as version 2 of the MATLAB scripts.



Figure 8: Vehicle landing while low gain of alignment to nodule task, hitting the rock.

1. Landing and Inspection actions

After the vehicle has landed and the *Inspection* action takes over, the magnitude of the distance between the Arm and the nodules was less than 1 meter as seen before from figure 4 which is enough to do the *Inspection* action without having the task of the *Arm Joint Limits* stopping the action defining task from being archived.

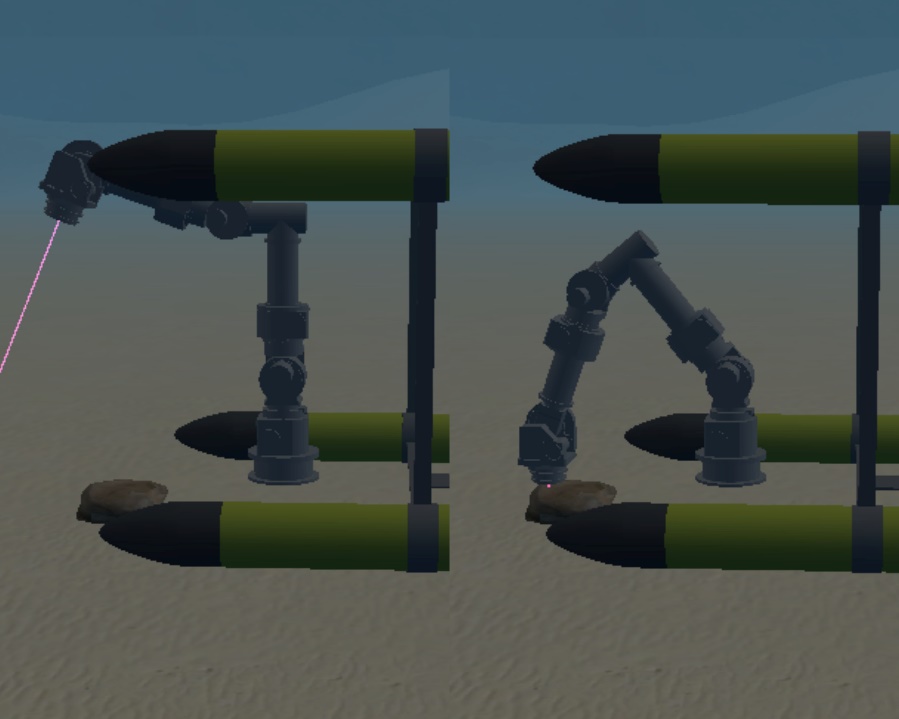


Figure 9: Vehicle between Landing and Inspection actions

1. Proof that nodule is in manipulator’s workspace

To guarantee that after the landing is finished the nodule is in the manipulator’s workspace. The task Jacobian of the task *Vehicle distance to nodule* should be computed with respect to the arm base frame and not the vehicle frame. As for this vehicle the distance between the base of the arm is not relatively far from the vehicle frame. However, the following solution was proposed to ensure a generic architecture.

The objective requires that the distance between the manipulator base and the nodule projected on the horizontal plane goes under a desired threshold in such a way that the nodule will be in the workspace of the arm. In other word the following equality objective must be satisfied:

Where t is the desired scalar position error between the manipulator base and the nodule position projected on the horizontal plane.

As seen this task is structured as an inequality task and acquire the attribute of operational prerequisite task.

= desired distance

So, the activation function will be an increasing bell-shaped function, and since control takes place over three scalars (the component of the error along x, y, z), of dimension 3x3:

Graphical user interface, application, Word

Description automatically generated

Figure 10: Increasing bell shape activation function.

Moving to the evaluation of the task Jacobian, the variable that must be controlled is the projection of the distance vector between the arm base and the nodule, which is equal to:

In which the distance in terms of also the vehicle reference frame origin can be expressed as:

Where is a constant vector expressing the position of the arm base with respect to the origin of the vehicle reference frame .

Now the evaluation of the projection of such a vector on the horizontal plane, obtaining the vector is:

To control that variable, it is needed to compute its derivative and express it in terms of the control variables embedded in the vector .

Where both and are equal to zero since the nodule is fixed in space during time as the distance between the origin of the vehicle reference frame and the arm base , obtaining:

Obtaining the task Jacobian:

Then the task reference, which is the controlling law that drive the control variable, can be obtained by modifying a bit the above relation as:

In which is the desired distance between the base and the nodule as described before.

# Franka Panda Bimanual Manipulation

1. Tasks unified hierarchy

As shown in the table below the Task Priority Control hierarchy is composed of four actions but before proceeding with the explanation of the control hierarchy it must be mentioned the calculations that were made for computing the transformation between the world and the base of each robot knowing that the left arm base coincides with the world frame. While the right arm base is rotated around the z-axis of the world frame with 180 degrees and is translated along it’s x-axis with 1.05 meters.

Moving on to the control hierarchy, the *Reaching goal* action which make the two arms move to the common target to be grasped, this action is composed of two tasks, which are the *Joints limit* “having the higher priority “ and reaching goal to be grasped.

Secondly, the grasping action starts once the robots reach the goal, this action consists of the kinematic constraint imposed by the object, the joints limit task and reaching goal\_1 task which moves the object to the wanted new goal known that the tasks are written with respect to the priority level.

Finally, the finishing action which contains one task responsible for stopping the whole movement of the end effectors for both arms.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Type | Reaching goal 1 | Reaching goal 2 | Grasping | Finish |
| Kinematic Constrain | E |  |  | 1 |  |
| Joint limit | I | 1 | 1 | 2 |  |
| Finish Task | E |  |  |  | 1 |
| Reaching goal | E | 2 | 2 |  |  |
| Reaching goal 1 | E |  |  | 3 |  |

1. Joint Limits task Jacobian and task reference
2. Kinematic Constrain task Jacobian and task reference