MIR 2024 Annual Symposium and Robotics Championship

Instituto Superior Técnico / Centro Náutico do Parque das Nações

<u>Robotics Championship</u>

Lisbon, 11 – 13 June, 2024

The challenge:

During the championship, you will have the opportunity to control a high velocity Autonomous Surface Vehicle. The goal is to design a controller that maximizes the following function:

$$\mathcal{J} = \omega_d D_f^2 - \omega_s \int_0^{t_f} \overline{s}(\tau)^2 d\tau + \omega_c \int_0^{t_f} c(\tau)^2 d\tau; \tag{1}$$

where ω_d, ω_s and ω_c are coefficients that define the weight of each of the components. The function $\overline{s}(\tau)$ is defined as follows,

$$\overline{s}(t) = \begin{cases} s(t) - s_{max} & s(t) > s_{max} \\ 0 & s(t) \leq s_{max} \end{cases}; \tag{2}$$

where $s(t)$ corresponds to the speed of the vehicle at time t, and s_{max} is a constant which defines the maximum allowed speed. The function $c(t)$ is defined as the minimum distance between the vehicle and the straight line path defined by the initial position of the vehicle and the initial yaw of the vehicle at time t. The function is illustrated in Figure 1 to provide greater intuition.

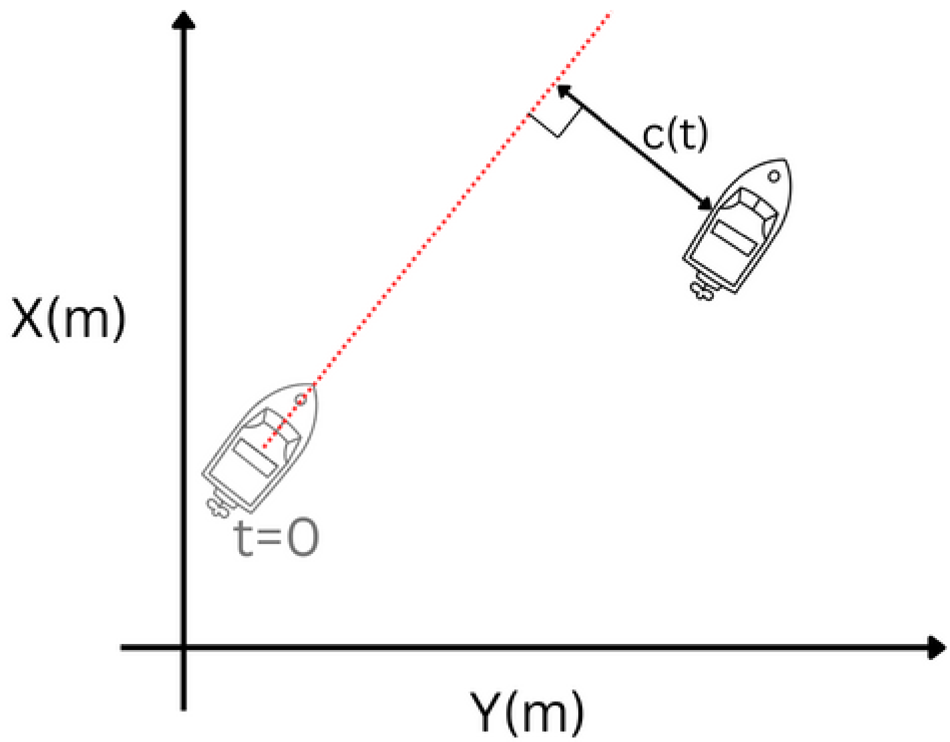

Figure 1: Illustration of the cross track distance.

Lastly, D_f corresponds to the distance between the initial position of the vehicle and the final position. Therefore, the goal is to go as far as possible in a straight line, without deviating from the path and without exceeding the speed limit.

Your controller will run during 20 seconds. The initial position and orientation (at $t = 0$) will be used to define the straight line path to be followed.

The Platform:

The vehicle you will be working with is shown in Figure 2:

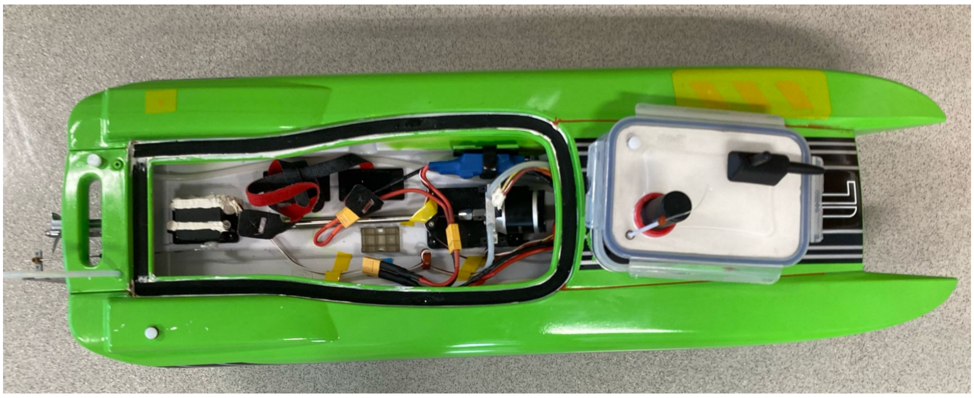

Figure 2: Autonomous surface vehicle used in the challenge.

The available actuators are a rudder and a propeller. Your controller will provide a value between 0 and 1 to the propeller, where 1 corresponds to max power and 0 corresponds to no power. Similarly, you will provide a value between -1 and 1 to the rudder, which will map to a rudder angle. An image of the actuators can be seen in Figure 3

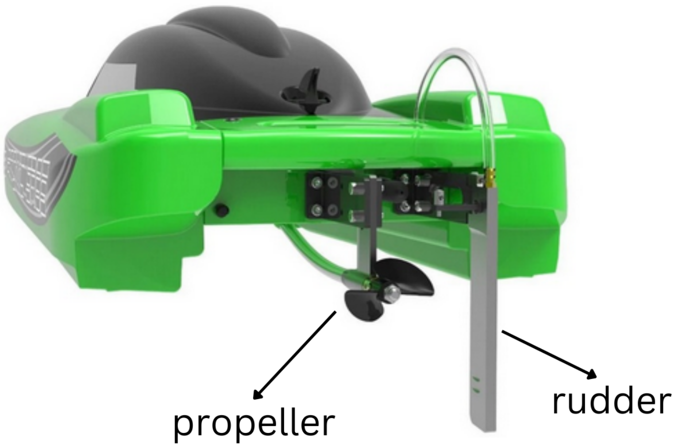

Figure 3: Actuators present on the vehicle.

The vehicle communicates with the ground station using both a telemetry radio for bidirectional data transfer and an RC controller and receiver for fail-safe reasons (in case of loss of telemetry/control, the RC controller can be used to disarm the vehicle). The ASV is equipped with a flight controller (the name comes from the fact that these micro controllers are usually employed in aerial applications, onboard quad rotors, fixed-wing airplanes, ...). The flight controller is used as a base due to the ease of sensor integration, safety features and state estimation algorithms. A Raspberry PI is connected to the flight controller and will receive the state, run your controller and send the inputs to the actuators.

The setup of both the vehicle and the ground station is presented in Figure 4.

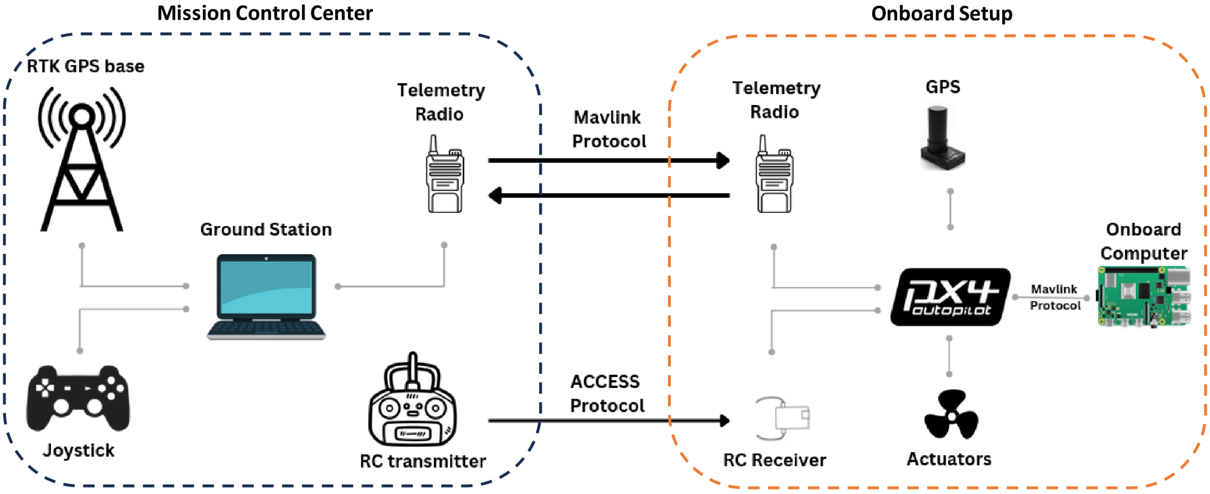

Figure 4: Experimental setup.

From the ground station, you will start the mission by entering Offboard mode (a mode where the flight controller allows the Raspberry PI to send commands to the actuator) and Arming the vehicle (this corresponds to allowing the motors to spin). After the mission starts, your controller will take over and will run during 20 seconds. After the 20 seconds, you will get your score.

To make changes to controller gains or test different controllers, you will need to manually (using the RC controller) bring the ASV closer so that you can connect by SSH to the onboard Raspberry PI and perform the changes you want to.

Model of the ASV:

A 3 DOF model was identified for the ASV. The available states are the surge, sway and yaw rate. The surge (u) is defined as the projection of the velocity onto the x axis of the body frame (x_B). Similarly, the sway (v) is defined as the projection of the velocity vector onto the y axis of the body frame (y_B). The yaw rate (r) corresponds to the rate of change of the yaw angle (ψ). The inertial frame is fixed to a point on earth and follows the NED convention. The body frame is fixed to the center of mass of the vehicle. These are illustrated in Figure 5.

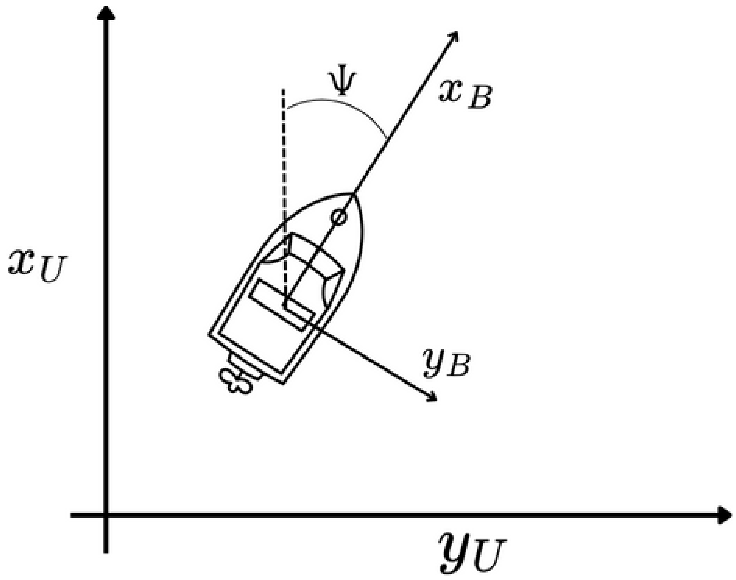

Figure 5: Coordinate frames.

Considering the above mentioned states, the following model of the motion of the vehicle was identified:

$$
\begin{aligned}
\dot{u} &= a_1 vr + a_2 u + a_3 u^2 + E_T \cdot \mathsf{u}_T + a_7 u|r| + a_8 u^2|r| \\
\dot{v} &= b_1 ru + b_2 v + b_3 v|v| + b_4 u \cdot sin(\delta_{rud}) + b_5 u^2 \cdot sin(\delta_{rud}) \\
\dot{r} &= c_1 vu + c_2 r + c_3 r|r| + c_4 u^2 \cdot sin(\delta_{rud}) + c_5 \cdot sin(\delta_{rud}) \cdot \mathsf{u}_T + c_6 ur + c_7 u^2
\end{aligned}
\tag{3}
$$

where u_T corresponds to the motor input (between 0 and 1), the rudder angle in degrees is given by:

$$\delta_{rud} = d_1 \mathsf{u}_R \tag{4}$$

where u_R is the rudder input (between -1 and 1).Finally, the thrust efficiency (E_T) corresponds to:

$$E_T = a_4 \left[1 - exp\left(\frac{-a_5 u^2 + a_6}{|r| + \epsilon} \right) \right] \tag{5}$$

Parameters of the model:

The following tables have the parameters identified for the model presented in the previous section.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
1.1965	-0.6218	-0.0216	16.4	0.0976	0.5056	-0.1154	-0.0025

Table 1: Identified parameters: surge evolution.

b_1	b_2	b_3	b_4	b_5
-0.1885	-4.4450	-0.1937	-0.3681	-0.1363

Table 2: Identified parameters: yaw rate evolution.

c_1	c_2	c_3	c_4	c_5	c_6	c_7
2.1225	-0.8592	-0.0963	2.2910	-5.5000	-1.9001	0.0123

Table 3: Identified parameters: yaw rate evolution.

d_1
60.000

Table 4: Identified parameters: mapping from rudder input to angle.

Important Notes:

- Although a model has been identified, it is not perfect, so your controller should be robust.
- Think about simplifications/assumptions that can simplify the model and allow simpler controllers to be designed.
- The model utilized for the simulation provided is **NOT** identical to the one depicted above.
- Again, **DO NOT** assume a perfect model, and try to simplify the one given taking into account the task you will be addressing.
- You controller should be simple, 20 lines of code should be more than enough.
- The model given has 3 DOF, however you only have 2 actuators.
- If possible design the controller such that if some changes occur to the identified parameters, the necessary changes are easy to do.
- Structure your code so that if you need to change gains, ... it can be easily done. This is important for the day of the challenge.

GOOD LUCK !!!