RoboBoat 2022: Technical Design Report
AGH Solar Boat

Robert Koziarski, Bartłomiej Wójcik, Maciej Borowicz, Mieszko Mieruński, Paweł Nowak, Szymon Dobrzyński, Marcin Warchoł, Mikołaj Grzybek, Olivier Lichota

aghsolarboatteam@gmail.com

***Abstract* – Since the year 2020 we've been meaning to compete at the RoboBoat competition. Last year, in 2021, we were unable to do so because of the pandemic and general shutdown of our team's operations. This year, we're finally able to compete with our boat, R.O.B.U.R. We see this competition as a great way of experimenting with new technology and a major engineering task. Since this is our first time entering in this competition, we plan on approaching only a limited number of tasks, with plans on increasing that number in the coming years. The system we've developed so far is heavily focused on vision based tasks which, luckily for us, compose the majority of the competition. Our hardware is extremely competetive, as we have great experience in building large racing boats.**

***Keywords* – Computer Vision, Mobile Robots, ASV**

I. COMPETITION STRATEGY

This year has proven extremely difficult for us, having to design and build not one, but two boats due to agreements the former management made with our University. We've bearly come out of a global pandemic and a complicated season, so we didn't have the time or manpower to aim at every single task at the competition.

Finally, we've decided to focus on just three tasks: *Navigation Channel, Avoid the Crowds* and *Snack Run!*.

With our boat building experience and know-how, we quickly got up to speed with the hull design and manufacturing, as well as completely new electronics. The greatest challenge we faced was building a proper vision based control system, which we had no prior experience with.

Despite that, we've managed to design an architecture that can provide autonomy during those three selected tasks.

Because our team and boat would have to travel halfway around the world, we've decided to keep it simple for the rest of the hardware. We focused on 3D-printed parts and parts we can machine in-house using existing equipment.

II. DESIGN CREATIVITY

A. *Hull design*

As mentioned before, we have quite an extensive experience with racing boat building. Our latest, largest and fastest one is named *Celka* and she was one of the main inspirations when selecting our building materials.

We chose preimpregnated with epoxy resin carbon fiber fabrics with aramid honeycomb for the composite sandwich. It provides extreme strength and stiffness with minimal weight.

The hull's shape was inspired by *catamarans* – boats who's hulls are composed of two smaller ones, connected together.

This design gives us stability and speed on straight courses, whilst retaining the ability to turn almost in one place.

B. *Powertrain*

The main goal of our powertrain design is to keep it simple. It consists only of a 3D printed motor mount, to which a nacelle with

the main vertical shaft is connected, and a propeller fixed to the horizontal shaft.

As for the materials, the nacelle is made out of symmetrical parts milled out of carbon fiber and epoxy resin blocks. The stainless steel shafts are held in their appropriate axis by five bearings. The top of the vertical shaft is attached to the motor through a clutch.

The boat has a differential steering mechanism, with symmetrically placed propellers on both sides of the back of the hull. This helps us with changing course, enabling the vessel to turn in one place.

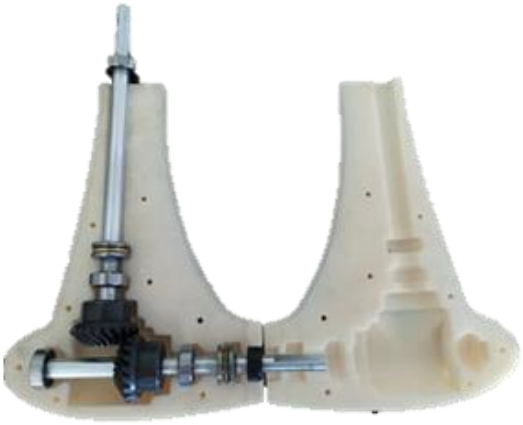

Figure 1. Inside of nacelle

C. *Power supply and electronics*

For the power supply we've decided to go with the *Power Brick+ 24V Battery Pack*, that should last us for hours of operation.

The whole electronic control system consists of two custom made PCBs. We've been also using a purchased *TinyBMS* with our older power supply, but that setup was only for develpoment.

First of the two is the *Battery Management Board*, which allows us to power the rest of the electronics on and off. Both of the required safety switches – the onboard one and the remote one – are connected to this board. It was designed and assembled by our electronics team in-house. We chose the STM32446RE for the CPU, as it has enough resources to support all of our software. The remote kill switch has been implemented using LoRa protocol and works completely independently with the manual one.

The other one is the *Control Board*. It's also a custom board, with the same processing unit and the main motor and radio access software. A *Crossfire* antenna connects to it

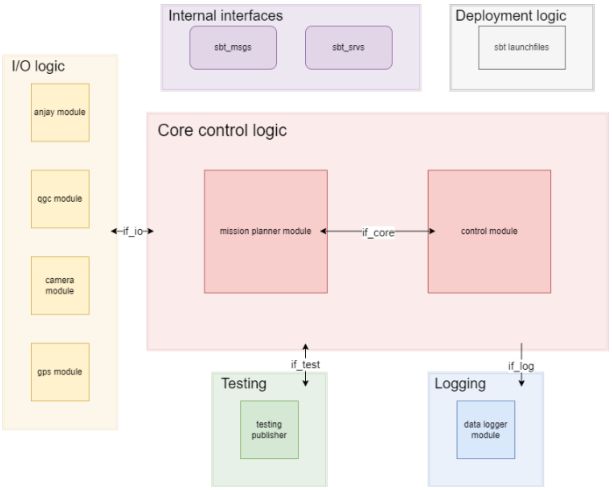

Figure 2. Software overview schematic

via UART, giving us remote control over the motors with a simple radio controller. We've also implemented three operation modes – *off*, *manual*, *auto* – enabling us to stop the propellers in case of emergency.

D. *Software*

The high-level control system has been implemented using ROS (Robot Operating System). It provides us with a complete framework for inter-process communication, which allowed us to modularize our system, which helps during development.

The system architecture is based on independent ROS packages, each with it's own responsibility. This allows us to test each one independently, version our software with ease and expand when in need of new features, all cutting down development time. Because we were working as a small team of five, there was need for task fragmentation.

We build all of our packages directly from source, with minimal outside dependencies. This increases the amount of work we have to put in, but allowed us to have full control over each and every aspect of the system. For storage and version control, we used the GitLab DevOps platform, which really helped us keep up with all the changes the team was making.

The main control logic is contained within the *Control Module* and *Mission Planner Module* packages. *Mission Planner Module* takes all detection data and calculates waypoints for the boat to succesively achieve. *Control Module* calculates direct motor steering based on waypoints received from the *Mission Planner Module*. This control value is then sent via UART to the *Control Board* that connects to the ESCs which drive the motors.

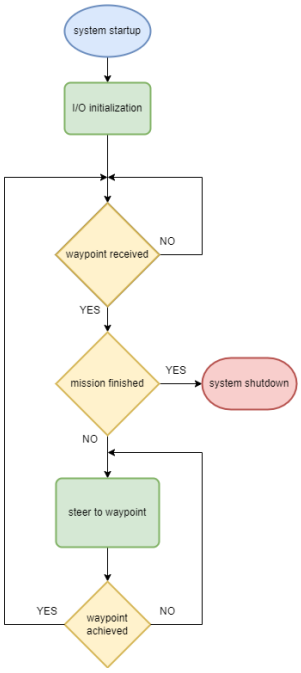

Figure 3. Control algorythm schematic

The *ZED2* camera that provides image data for object detection and classification is driven by the *Camera Module*. We utilize the producer's API to control whether we're grabbing images or not. This optimizes the amount of data that's circulating in the system. The module publishes detection messages containing the detected object's posistion relative to the boat and a timestamp. The *Mission Planner Module* then interprets this information to calculate the next waypoint.

Additional cameras, that help with obstacle avoidance are supported via additional *Camera Modules* that differ in implementation, depending on underlying hardware.

To be able to navigate on large bodies of water, we utilize GPS (GLONASS to be exact). The *GPS Module* receives NMEA messages via UART, processes and publishes them to an appropriate topic.

We log selected messages in two ways. The *Data Logger Module* can either write raw messages with a timestamp into a file or, if internet access is available, send them to a remote database. Our data is stored on our private instance of InfluxDB – a time series database, with excellent visualisation capabilities.

To deploy and run our system remotely, we use the combination of *roslaunch* and Linux services.

The architecture presented above can be extended further, integrating different sensors and more advanced control logic.

III. EXPERIMENTAL RESULTS

A. CFD Simulations

We performed initial testing of our hardware components using ANSYS Fluent CFD software. Our construction team benchmarked different hull and propeller designs. During the design process, a comparative analysis of the hull geometry was performed. Using calculations based on Computational Fluid Dynamic (CFD), the drag force of the hull was determined during movement in the water. The hull geometry was designed as surfaces, converted to a solid model and then meshed.

The parameter calculation model, apart rom diffrent hull geometries, took into account variable velocity values – from 0.5 [m/s] to 3.0 [m/s], with a step of 0.5 [m/s]). On the basis of computational fluid dynamics, the values of the hull's drag force were obtained, and graphs were made. Figure 4 shows the graph of the drag force of the boat hull as a water in the function of velocity.

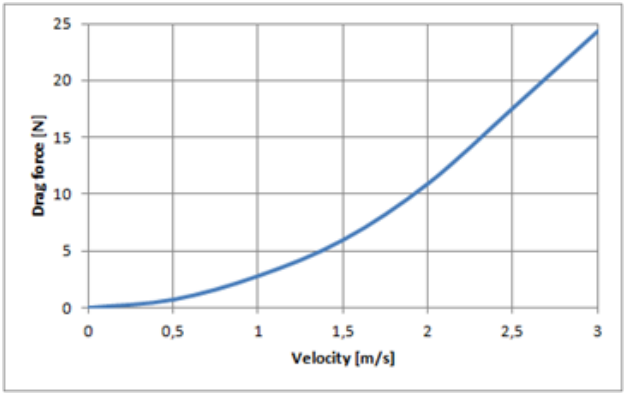

Figure 4. Graph of the drag force as a function of velocity

The wave speed for the tested simulation was 3 m / s. The purpose of this selection of parameters was to test the flow velocity of the

boat under normal operating conditions. The use of the K-Omega SST computational model allowed for an appropriate combination of the boundary layer with free-flow elements (combining the advantages of k-epsilon and k-omega). The layer of the two-phase part was additionally thickened - it was aimed at the best reproduction of the waves and reflections around the hull geometry. The obtained data indicate that the model was designed correctly - no negative turbulence can be seen.

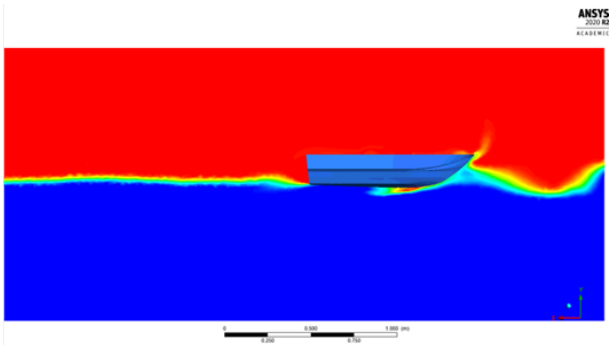

Figure 5. The velocity of water flow in catamaran hull

The proper construction of hull geometry is one of the crucial aspects from the standpoint of hydrodynamic resistance and consequently - boat performance and energy consumption.

Based on the obtained results, a two-phase simulation was also performed (see Figure 6). Its aim was to show the behavior of the catamaran in a two-phase environment, that is, both parts in water and parts in air. Such a simulation allowed for an optimal selection of the shape of both the hull and the deck of the boat.

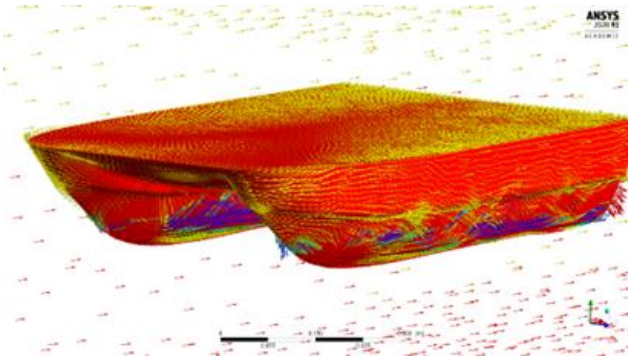

Figure 6. A two-phase simulation of the hull

A simulation of a small wave collision with the hull was also carried out. Figure 5 shows the catamaran cross section, showing the moment of wave impact. It may be observed that wawes will not affect the boat significantly, which is important from the point of view of safe and energy-efficient operation on the water.

B. Gazebo simulation

To test our software, we've included simulation software in our project. Gazebo was the obvious choice, as it has excellent ROS integrations, allowing to run the exact same code that will be deployed on the boat.

The main difficulty was to implement sensors such as GPS and Imu that can mimic real ones. We've managed to get decent results meaning our control logic could be tested and worked fine in the simulated environment.

As a baseline project, we used the VRX simulation [6], which we extented by writing packages linking our topic namespace with the simulation's.

IV. ACKNOWLEDGEMENTS

This project was supported by Miquido, Stereolabs, RoboNation, MAT and AVSystem. AGH Solar Boat team receives financial support from AGH University of Science and Technology for which we give thanks.

V. REFERENCES

[1] ROS Documentation
http://wiki.ros.org/Documentation,
[2] Mavlink Developer Guide
https://mavlink.io/en/,
[3] cppreference
https://en.cppreference.com/w/,
[4] InfluxDB Python API Documentation
https://docs.influxdata.com/influxdb/cloud/api-guide/client-libraries/python/,
[5] Debian Manpages
https://manpages.debian.org/,
[6] VRX Gazebo simulation
https://github.com/osrf/vrx

VI. APPENDIX A: COMPONENT SPECIFICATIONS

Component	Vendor	Model/Type	Specs	Custom/ Purchased	Cost	Year of Purchase
ASV Hull Form/Platform	-	-	made of carbon fiber composite	custom		2021
Waterproof Connectors	KSS Wiring	-	-	purchased	25$	2021
Propulsion						
Battery	Power Tech systems	Power Brick+ 24V Lithium-Ion battery pack	24V, 32Ah	purchased	800$	2022
DC/DC Converrter	Daygreen	24/12	24V to 12V converter	purchased	30$	2021
BMS	EnergusPs	TinyBMS		purchased	250$	2021
Motor Controls	MGM	TMM 14063-3		purchased	680$	2021
SBC	UP board	UP Squared		purchased	350$	2021
Image processing unit	Nvidia	Nvidia AGX Xavier		purchased	480$	2021
Controlling PCB v1	-	-		Custom	50$	2021
Controlling PCB v2	-	-		Custom	50$	2021
Teleoperation	Hoperf	RFM95		purchased	30$	2021
GPS	Velleman	VMA430		purchased	35$	2021
Intertial Measurement Unit (IMU)	Stereo Labs	ZED 2	Part of ZED 2 camera	purchased	0$	2021
Doppler Velocity Logger (DVL)						
Camera(s)	Stereo Labs	ZED 2		purchased	450$	2021
Hydrophones						
Algorithms				Custom		
Vision	Miquido	Neural network	Buoya detection	custom	0$	2021
Localization and Mapping				Custom		
Autonomy				Custom		
Open-Source Software				custom		