

# RobotX 2022 Technical Design Paper

*National University of Singapore (Bumblebee Autonomous Systems)*

Amadeus Aristo Winarto, Ananya Agarwal, Chew Zhi En Samuel Joshua, Chin Zheng Hao, Gokul Rajiv, Ho Wei Zong Jasper, Justin Foo Guang En, Kaitlyn Ng Ke Yi, Lee Chan Wai, Lee Shi-An Matthew, Lee Tze Han, Li Po Hsien, Lim Sheng Wei, Lu Sicheng Isabella, Manzel Joseph Seet, Marvin Pranajaya, Michael Jervoso, Ng Cheng Yang Titus, Ng Xing Yu, Ng Zhia Yang, Nguyen Minh Nguyen, Niu Xinyuan, Png Qun Shen, Quek Wei, Rani Karthigeyan Rajendrakumar, Seow Alex, Stevanus Williemi, Tan Chern Lin Justin, Tan Chew Miang Edwin, Teoh Xu En, Tran Phuoc Huy Khang, Yam Jin Ee Dmitri, and Zhu Tianqi

**Abstract**—Bumblebee Autonomous Surface Vessel 3.0 is the newest iteration of Team Bumblebee’s Autonomous Surface Vessel (ASV), and features major upgrades to the mechanical, electrical, and software subsystems. The integration of an Unmanned Aerial Vehicle in RobotX 2022 presented several challenges which are discussed in this paper, along with the various upgrades we have added to the ASV during our development process.

## I. DESIGN STRATEGY



Fig. 1: Bumblebee Autonomous Surface Vessel 3.0.

The Bumblebee Autonomous Surface Vessel 3.0 (BBASV 3.0) (Fig.1) was the product of a complete overhaul of our previous BBASV 2.0, addressing various shortcomings while preparing for the changing requirements in RobotX 2022. There were 3 key objectives guiding Team Bumblebee’s development:

Firstly, our development process centred around adding new capabilities, pushing the envelope of what had been previously explored. With the introduction of an Unmanned Aerial Vehicle (UAV) to RobotX 2022, a rethink of our existing design was deemed necessary. This culminated in custom modifications made to a DJI M210, as well as the design and implementation of an autonomous drone take-off and landing system for a comprehensive ASV-UAV integrated deployment capability.

Secondly, the software architecture also received a major upgrade. Since the last RobotX competition in 2018, the software stack has seen many improvements, including a brand-new

Behaviour Tree (BT)-based mission planner, a revamped machine learning pipeline, and a new control system. Integrating the improvements to the software stack with BBASV 3.0 brought many new capabilities, and also proved that our new software architecture design was adaptable to accommodate a new vehicle without major changes.

Lastly, upgrades and redesigns also centred around ensuring that maintenance and repairs could be done swiftly, to ensure less downtime during testing. This included quality of life upgrades, such as making sensors like the radar easier to mount, a more accessible main hull for quick debugging and testing even while it is on the water, and easily visible telemetry readouts to ensure that the state of the Autonomous Surface Vessel (ASV) can be quickly and easily known.

## II. VEHICLE DESIGN

### A. Mechanical Sub-System

#### 1) Sensors & Antenna Mast

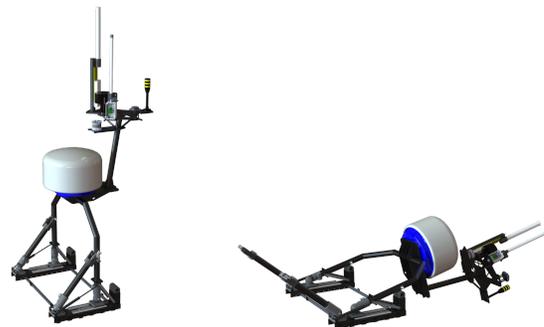


Fig. 2: The redesigned sensors & antenna mast. Left: Raised. Right: Lowered.

The sensors and antenna mast (Fig.2) houses most of the sensors and stands 2.5 m above sea level, providing a good balance between antenna range, mast weight, and ease of assembly. As part of the upgrades to BBASV 3.0, a Wäertsillä RS24 radar was installed, resulting in a threefold increase in weight over the previous Raymarine Quantum radar. The GPS was also upgraded to a dual-antenna model, which necessitated the installation of an extra antenna. Lastly, an anemometer was added to assist in control systems. All these additions added considerable weight to the mast.

Furthermore, we observed slight cracking along the base of the mast during RobotX 2018, due to vibrations induced by the strong waves. Transportation of the ASV also caused vibrations that resulted in the deflection of various components. Taking all these factors into account, we felt that a complete redesign of the mast was necessary to ensure reliability.

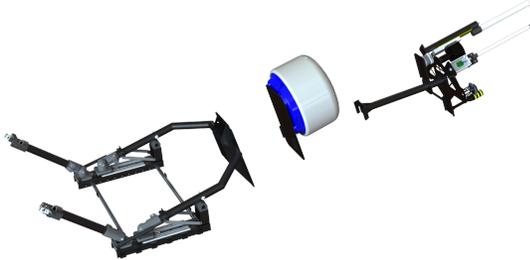


Fig. 3: The 3 interlocking parts of the mast.

The redesigned mast is split into 3 separate interlocking parts (Fig.3) to allow quick disassembly for transportation, reducing the risk of damage to the mast before deployment. Furthermore, we designed the mast to be able to be completely lowered (Fig.2) to allow for easier maintenance and storage of the ASV.



Fig. 4: Mast raising/lowering mechanism.

For the new design, an arch support was chosen over the single beam support featured on the previous mast, in order to reduce the stress sustained by the support system. Instead of a welded base, we opted for aluminium swivel clamps (Fig.4) to allow the mast to be easily raised and lowered, without compromising on strength. After raising the mast, 2 support arms lock it in place to ensure stability. While lowering the mast, a linear guide rail (Fig.4) is used to shift its pivot point towards the centre of the ASV. This aligns the centre of mass of the mast and ASV as much as possible, preventing a large weight imbalance. Lastly, a gas spring (Fig.4) is installed on both sides of the arch support to assist in raising the mast.

## 2) UAV Development

The Bumblebee Unmanned Aerial Vehicle 1.0 (BBUAV 1.0) (Fig.5) is Team Bumblebee's first foray into unmanned aerial vehicles, and is based on a DJI M210. The decision to modify an off-the-shelf UAV was made in consideration of the short development time available, as well as the team's lack



Fig. 5: Bumblebee Unmanned Aerial Vehicle 1.0.

of experience with UAVs. The DJI M210 provides a stable platform with a Robot Operating System (ROS) interface, allowing us to quickly jump-start development.

Given the regulations regarding the maximum payload capacity of UAVs at RobotX 2022, a major design consideration for BBUAV 1.0 was weight. Multi jet fusion was used to fabricate the shell and electronics hull, allowing us to achieve complex part geometries that are simply impossible with traditional fabrication techniques. This allowed us to vastly reduce part count and weight, while improving the shell's robustness.

Finally, the drone is also equipped with a custom power distribution board that receives power from an external lithium polymer battery and supplies power to the carrier board, radio board, and internal components such as the cooling fan. Due to the weight and size constraints, most of the electronics are packed into a small space. As a result, heat dissipation from the voltage regulators could greatly increase the internal temperature of the hull, potentially overheating sensitive components such as the cameras and onboard computer. To counteract this, we ensured that the electronic hull had vents for heat dissipation, and rearranged the components to direct the heat away from sensitive components. Finally, we also integrated a fan into the hull to improve circulation and internal hull thermals.

## 3) UAV Flotation System

BBUAV 1.0 is equipped with a flotation system that automatically deploys upon crashing into the water, ensuring that the UAV remains afloat for easier recovery. A 16g canister filled with highly pressurised CO<sub>2</sub> gas inflates the floats to provide buoyancy. However, the high pressure presents challenges in selecting a suitable valve. Hence, a custom-made, servo-driven actuation system was designed around the CO<sub>2</sub> canister's valve head to ensure maximum reliability.

## 4) Drone Take-off & Landing System

The Drone Take-off and Landing System (DTLS) (Fig.6) ensures the safe deployment of the UAV. It consists of 4 linear stages driven by stepper motors to clamp BBUAV 1.0 when it lands on BBASV 3.0, securing it safely to the ASV when it is not in use.

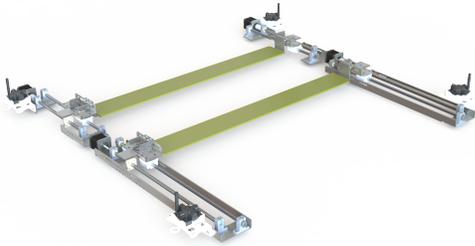


Fig. 6: Drone Take-off and Landing System.

These linear stages are independently driven by 4 stepper motors. Using StallGuard™ technology allows the actuators to achieve sensorless homing. Additionally, the linear stages stop clamping upon encountering resistance over a fixed threshold, allowing a successful drone clamping to be detected without additional sensors.

The clamps of the DTLS are designed to secure and reorient the UAV after landing. Additionally, these pushers are spring-loaded to accommodate misalignment of the linear stages. Oil-impregnated polymer bearings were used to reduce part count and cost, as well as increase maintainability while ensuring smooth motion.

To optimise the stepper motors’ current draw, the DTLS utilises CoolStep™ technology to dynamically set the maximum current threshold based on the load of individual motors. Energy consumption and heat generation are minimised, as the stepper motors now always draw the minimum current for each of the different required loads.

5) *Electronic Hulls*



Fig. 7: Main hull.

Similar to BBASV 2.0, the main hull of BBASV 3.0 is a modified Pelican case (Fig.7), and contains the onboard computer, as well as the power and sensor subsystems. It is secured with ratchet straps, allowing the hull to be easily opened to facilitate troubleshooting. To accommodate the new drone take-off and landing system, the main hull was shifted from the middle of the payload tray to underneath the

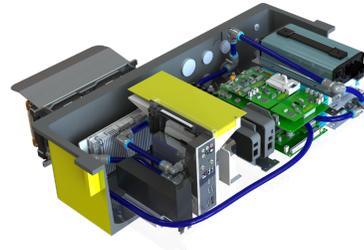


Fig. 8: Main hull liquid cooling system.

sensors & antenna mast. The liquid cooling system (Fig.8) was upgraded to use water blocks to directly cool any heat-generating internal components, significantly reducing their temperatures.



Fig. 9: Secondary navigation and acoustics hull.

A secondary hull (Fig.9) houses our navigation and acoustics boards, isolating them from the significant electrical noise generated by the power subsystem. This hull is mounted underneath the payload tray to save space, and contains all the electronics required to run our actuators, allowing for quick debugging of these systems without needing to open the main hull.

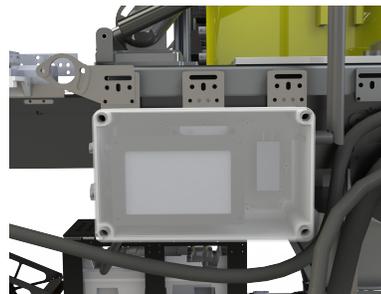


Fig. 10: Telemetry hull.

Finally, a telemetry hull (Fig.10) mounted on the side of BBASV 3.0 houses a screen which displays vital information, such as battery voltage and hardware statuses.

6) *Main Hull Shelter*

The main hull shelter (Fig.11) prevents the main hull from being directly exposed to sunlight, reducing heating due to black-body absorption. Additionally, it serves to provide extra protection against heavy rainfall. This year, the shelter was re-designed to accommodate the new drone take-off and landing

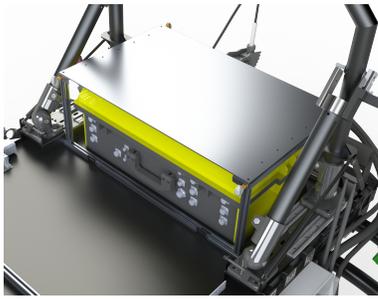


Fig. 11: Main hull shelter.

system, which takes up much of the payload tray and reduces the available space for the shelter.

Furthermore, we also found that the canvas used in the previous iteration would often trap wind and cause significant upward drag on the ASV. Thus, the new design fits the main hull exactly, and is constructed out of aluminium profile bars and composite sheets. It is secured by locking pins, allowing the top sheet to be quickly removed for easy access to the main hull during maintenance.

7) *Ball Shooter*

The BBASV 3.0 ball shooter was redesigned to resolve a key problem faced by the previous iteration, which used a double flywheel design that consisted of counter-rotating wheels to shoot the balls. While mostly reliable, the reliance on friction rendered the system useless in wet weather.

To reduce the reliance on friction, two different ideas were explored. The first design is spring-loaded, while the second design makes use of tension, similar to a crossbow. A TIMOTION MA4 linear actuator was chosen to arm the ball shooter for both designs due to its reliability at high load.

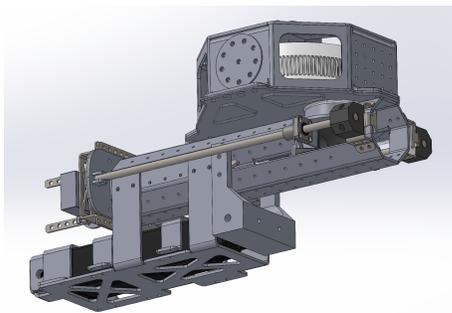


Fig. 12: Assembly of the spring-loaded design.

The spring-loaded design (Fig.12) uses the linear actuator to push a launchpad, compressing a spring until it is locked in position by a custom-made latch. The latch is then released by an IP67 rated servo and the spring accelerates the launchpad to shoot the ball. Guiding rails and linear bearings are used to align the launchpad to ensure shot consistency.

The launchpad (Fig.13) uses sheet metal attached to a 3D-printed part to balance weight and strength. The latch was designed to fit inside the barrel with only its tail exposed

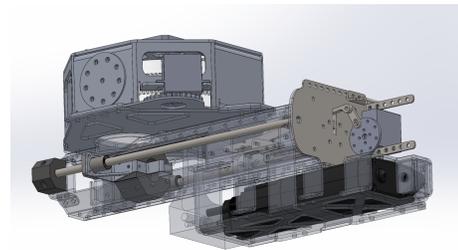


Fig. 13: Launchpad pushed back by the linear actuator and aligned by the guiding rails.

for activation to keep it aligned with the launchpad. The shot strength is manually adjustable by various means, including moving the spring forward and moving the latch back, allowing the shot trajectory to be fine-tuned.

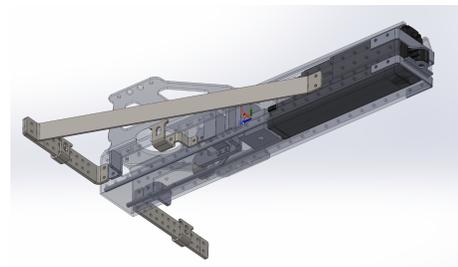


Fig. 14: Assembly of the crossbow design.

The crossbow design (Fig.14) differs by using the linear actuator to move a standard rotary latch to latch onto the kicker. The surgical tubes that connect the arms to a 3D-printed kicker are then tensioned when the linear actuator retracts. The latch is then released to shoot the ball. The linear actuator in this design is aligned with the axis of motion of the kicker to reduce the stress experienced by the the kicker and guiding rods.

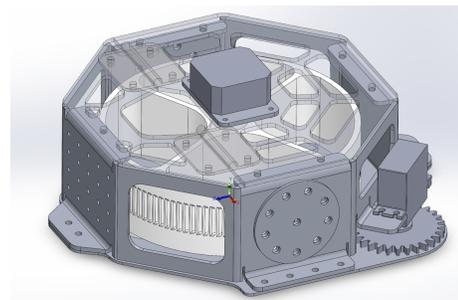


Fig. 15: Turntable-based reloading drum.

To reload the ball for both designs, a turntable-based reloading drum (Fig.15) is attached over the shooting barrel. The stepper-actuated centre divider of the turntable rotates to drop the ball through the hole in the drum into the barrel.

The ball shooter had to be moved below the payload tray to make way for the drone take-off and landing system, resulting in a new mount being made (Fig.16). This mount retained the adjustable shot angle of the previous iteration, having holes spaced 5° apart. The ball shooter also has a modular

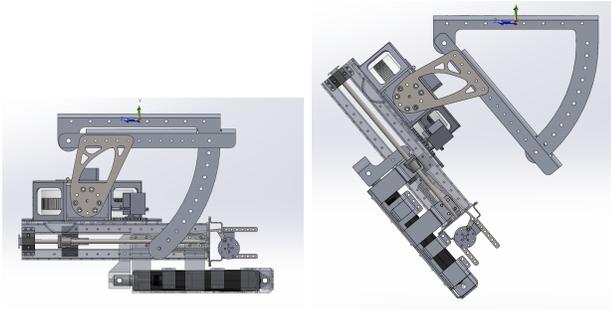


Fig. 16: Ball shooter mount and manual aiming mechanism.

design, with the mount only attaching to the reloading drum, allowing either of the two shooting mechanisms to be swapped out.

**B. Electrical Sub-System**

**1) Electrical Architecture**

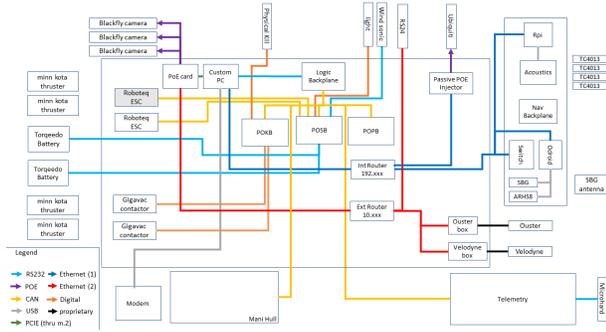


Fig. 17: BBASV 3.0 Electrical Architecture.

The electrical system of BBASV 3.0 (Fig.17) is split into multiple hulls. The main hull contains the electronics which form the backbone of the electrical system, and are categorised into 2 subsystems: power distribution, and sensors and thruster controls. Ethernet and Controller Area Network (CAN) are our primary methods of communication between the high-level and low-level components, as they ease the adding of new peripherals.

**2) Control Architecture**



Fig. 18: Operator Control System.

The Operator Control System (OCS) (Fig.18) communicates to BBASV 3.0 through 2 separate wireless interfaces. The data

link is a high-speed link used mainly for the streaming of sensor data such as the ASV camera feed. The control link is a robust long-range link used to send hardware statuses and provide safety features such as an emergency stop for the ASV. The control link uses N2420 radio modems, which form a dynamic mesh network to relay messages from one modem to another. A 10 dBi directional antenna on the OCS maximises the range of the control link to ensure control is maintained of the ASV at all times.



Fig. 19: OCS Ground Station.

The OCS is also part of the kill system. Activating a latching push button will send the kill command to the ASV via the N2420 radio modem attached to the OCS ground station (Fig.19). To ensure the safe operation of BBASV 3.0, multiple fail-safes are also incorporated into the kill system. If no messages are received from the OCS radio modem, or the sensor board crashes, the thrusters will be killed as a precaution, as the sensor board sends values to the thruster ESCs.

**3) Power Distribution & Power Control**

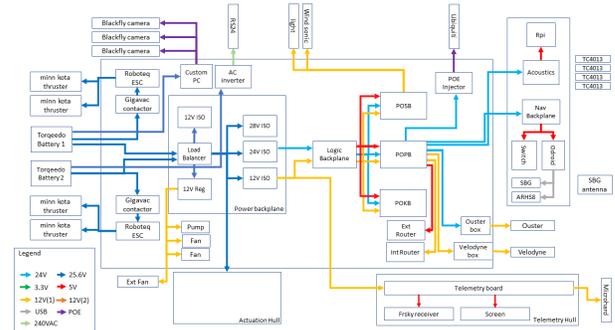


Fig. 20: BBASV 3.0 Power Architecture.

BBASV 3.0 has a multitude of electronic components that require 5 V, 12 V, and 24 V inputs (Fig.20), powered by 2 Torqeedo Power 26-104 batteries. To enable battery hot-swapping capabilities, the electronics in the main hull receive power from the load-balanced output of the 2 batteries. Moreover, to minimise electrical noise and enhance circuit protection, all our regulators are isolated and have built-in safety features such as over-voltage/current protection. From

our extensive testing, we found that the power regulators dissipate large amounts of heat, which pose a problem to the internal temperature of the main hull. As such, the heat sink of the regulators are water cooled.

Furthermore, we are using a Würtsilä RS24 radar in BBASV 3.0. The radar requires AC power, necessitating a DC to AC inverter. In the beginning, we faced power issues where the radar occasionally shuts down, and we eventually narrowed down the problem to the inverter overheating. Hence, we swapped to a more efficient True Sine Wave inverter that is able to withstand temperatures up to 70 °C.

To enable power control of the various components on BBASV 3.0, the same load switches introduced in BBASV 2.0 were used. This prevents both negative voltage spikes when the input voltage is stepped, as well as inrush current when the control signal is toggled. However, the fuse used was found to be inaccessible and hard to replace. Fuse holders were used instead of soldered-in fuses so that they could be replaced easily while still retaining overcurrent protection.

Furthermore, power control was extended to the onboard computer and radar. For these 2 components, a power ‘button’ is controlled by shorting two terminals. Thus, an electromechanical relay is formed to toggle the power state of these components.

#### 4) Ball Shooter

Initially, the design of the ball shooter called for 3 steps for firing and reloading, and we chose the Trinamic SilentSlapStick drivers for their compact size, high current, and extensive feature set. However, testing revealed that the steppers did not provide the necessary force required for the ball shooter, due to 2 main reasons: while the theoretical force provided by the drivers was enough to push the spring, the stepper drivers tended to heat up extremely quickly when set to near their current limit. Furthermore, unforeseen additional stresses added extra tension for the steppers to overcome, which resulted in the steppers being unable to fully compress the spring.

Hence, the design was changed to a linear actuator based system, driven by a Polulu 24v13 Motor Driver that can support up to 13 A. Due to the inductive loads, all iterations of the ball shooter PCB adopt an isolated design where the power input and output signals are isolated from the microcontroller and CAN communication channels.

#### 5) Connectors

In BBASV 2.0, two types of connectors were used. Soriau UTS Series connectors were used for power and low data rate signals, while Glenair Mighty Mouse connectors were used for Ethernet and POE signals, which require higher data rates.

For BBASV 3.0, we decided to cut costs by changing all our Glenair Mighty Mouse connectors to Soriau UTS Series connectors, as they were up to 10 times less expensive. We chose the solder cup version of the connectors, and paired them



Fig. 21: Broken Soriau UTS Series connector.

with thin 26–28 AWG wires. However, this turned out to be a mistake, as they were extremely prone to breaking (Fig.21), exacerbated by the frequency at which we were removing components from the main hull during testing. For example, removing the main hull electronics tray required us to bend the wires at extreme angles, putting strain on the wires and causing them to break at the solder point.

We attempted to switch back to the Glenair Mighty Mouse connectors, but found that the lead time for procuring new connectors was unacceptably long. Hence, we made the decision to make 200% spare connectors so that we are able to quickly swap the connectors if any connector breaks, reducing downtime. We have started sourcing for alternative connectors for our next iteration of the BBASV, and hope to be able to improve this situation in the future.

### C. Software Sub-System

#### 1) Software Architecture

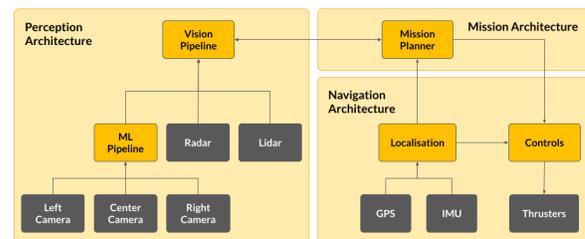


Fig. 22: BBASV 3.0 Software Architecture.

The BBASV 3.0 software architecture (Fig.22) is largely similar to that of the Bumblebee Autonomous Underwater Vehicle 4.0 by design, as our goal is to maximise code reuse, lowering the burden of maintaining 2 separate software stacks. Notably, major subsystems such as the mission planner, machine learning pipeline, and control system are shared between both vehicles.

#### 2) Mission Planner

For RoboSub 2022, we switched to a Behaviour Tree (BT)-based mission planner (Fig.23) for its flexibility and maintenance ease. Based on the *BehaviorTree.CPP* package, it has many advantages over a traditional Finite State Machine (FSM) planner, like being able to quickly interpret the ASV’s behaviour visually. The linear structure and well-defined node

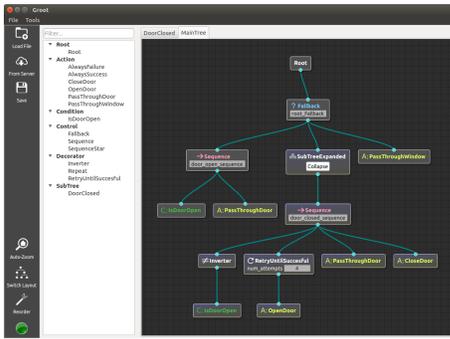


Fig. 23: Mission Planner GUI.

transitions also make it easier to configure robot behaviours simply by repositioning nodes, without accounting for innumerable state transitions. This lets our planner accommodate more situations, whereas an FSM would require exponentially more states and transitions to compete.

BT planners also give us a high level of composability and abstraction: both nodes and subtrees can be reused, allowing us to define increasingly complex robot behaviours easily. This was especially valuable when porting our mission planner to BBASV 3.0, as many nodes could simply be reused with minor modifications. Furthermore, our planner allows for tree structure and parameter configuration at runtime without recompilation, saving precious time in a competition setting.

3) Machine Learning Pipeline

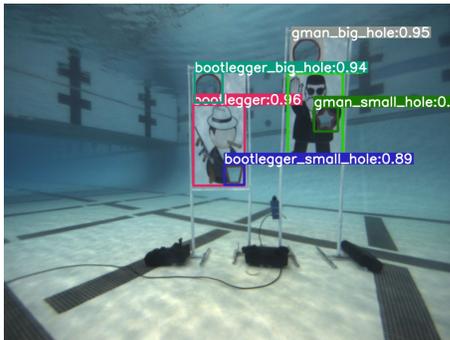


Fig. 24: Machine learning-based object detection during RoboSub 2022.

We completely revamped our machine learning pipeline (Fig.24) this year, making it faster and more modular, and have ported it to BBASV 3.0. Instead of a mixture of Python 2 and Cython, we now use Python 3 for easier maintainability without sacrificing speed. The new modular design also allows additional models to be added in the future. We also refactored our existing YOLOv5-based architecture to allow the use of TensorRT, letting us leverage device-specific optimisations, maximising our NVIDIA RTX A2000’s potential and gaining a 2x FPS increase.

One problem that we faced during testing was that despite accurately determining the bounding boxes of the buoys, our machine learning model had problems classifying the buoys correctly (Fig.25). We hypothesised several factors: firstly, we



Fig. 25: Red buoy wrongly identified as a white buoy (left) and black buoy (right).

may simply not have collected enough training data, given our limited ASV testing time during the school term. Secondly, like many other models, YOLOv5 augments its images before training, includes HSV augmentation, possibly reducing the accuracy of colour detection. Lastly, differences in lighting conditions may result in differently-coloured buoys giving the same pixel-level values, causing the model to become colour-agnostic when detecting buoys.

To overcome this problem, we use traditional computer vision techniques. Using the bounding boxes given by the machine learning model, we use colour difference algorithms such as CIEDE2000 to extract the dominant colour in the bounding box, allowing us to accurately determine the buoy’s colour.

4) Control System

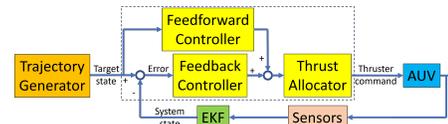


Fig. 26: Control architecture.

We implemented a trajectory generator (Fig.26) to create smooth continuous paths for BBASV 3.0 to follow. The trajectories are calculated using linear segments with polynomic blend, with limits on the maximum velocity, acceleration, and jerk of the vehicle. This improves performance for distant setpoint goals by avoiding controller saturation.

The control system makes use of control law partitioning, including a full state feedback controller which enables positional and velocity tracking, and a feed-forward controller to compensate for non-linear terms in the vehicle’s motion dynamics.

Our thrust allocator uses quadratic programming to optimise each thruster’s command based on the required forces, and maintains control along each axis of motion even during thruster saturation.

5) Inter-vehicle Communications

To allow communication between the ASV and UAV, off-the-shelf radio modules from Microhard, operating at 2.4 GHz, were used. Its simple communication protocol — serial UART — allowed us to drastically reduce testing and development time; development PCBs were produced which allowed control and testing via a USB connection.

For reliable communication, a TCP-esque protocol was implemented on top of the rudimentary (unreliable) raw radio interface. This allows the rest of the software architecture to assume that any messages will always be delivered, eliminating any additional complexity resulting from inter-vehicle communications.

Since both the UAV and ASV run ROS, communication between the two vehicles happens primarily through normal ROS topics. Since the underlying reliable radio protocol supports transmitting arbitrary binary data, we simply repurposed the existing (de)serialisation functions in ROS to send messages over the radio. On each vehicle, we create nodes that translate between radio messages and ROS topics, allowing for seamless interoperability — it is functionally indistinguishable from the topic publishers being on the same vehicle.

A particular design choice that we made was to separate the two vehicles' ROS masters, rather than running both machines on a single ROS instance and directly sending messages over the radio channel. The main reason for this choice was reliability; in the event that the radio channel is disrupted (due to range, interference, etc.), it is crucial that the UAV's software stack continues to function independently of the ASV, and it can still perform tasks like landing. Another consideration was the lower latency involved with a custom protocol, rather than directly using TCP.

### 6) UAV Localisation

To allow the UAV to perform localisation over the ASV while deployed, two separate systems were used. Firstly, ultrasonic beacons from Marvelmind Robotics were installed; 4 receivers are placed on the corners of the ASV, with two ultrasonic wave emitters on the drone. With this setup, the drone's beacons can triangulate their position with respect to the 4 ASV beacons, allowing for a pose estimate (relative to the ASV) at a range of up to 20 metres.

For finer-grained localisation — important for landing — we employ Aruco markers. In order to perform landing, the ASV first sends its GPS position via radio to the UAV, which then proceeds to fly toward that position. Once within range, we use the ultrasonic beacons to position the UAV over the ASV, with the objective of getting the Aruco markers into view of the camera. At that point, the UAV utilises the pose estimations given by the markers to land on the landing platform.

## III. EXPERIMENTAL RESULTS

### 1) In-water testing of BBASV 3.0

To facilitate testing of BBASV 3.0, our team secured a wet berth at the Republic of Singapore Yacht Club (RSYC), one of our supporting organisations. Due to administrative problems, we were only able to begin testing in mid-September, pushing us to conduct testing at the RSYC marina 3 to 4 times a week, despite it being the middle of a school term.



Fig. 27: Testing BBASV 3.0 at RSYC.

### 2) Troubleshooting Radar Power Issues

To operate our Wäertsillä RS24 radar, power is delivered from the battery through a load balancer, followed by an isolated DC-DC converter, and finally an inverter. During initial electrical testing, it was found that the radar could only be powered by a DC power supply through the inverter. When the same inverter was connected to the battery through the converter, it would continuously power cycle, causing the radar to be unable to start up.

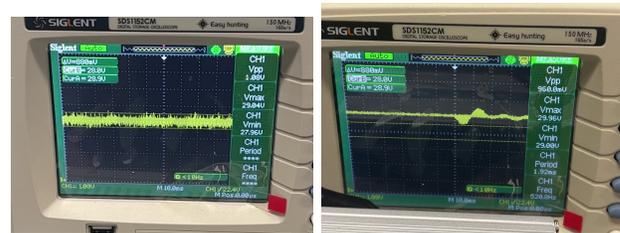


Fig. 28: Oscilloscope data for the isolated DC-DC converter (left) and inverter (right).

To narrow down the problem, we tested the converter and inverter separately. We probed the converter with an oscilloscope and found that the maximum voltage provided was around 29 V (Fig.28). Power cycling was not observed when providing the inverter with a 29 V DC input. However, upon probing the inverter, we observed a large voltage ripple (Fig.28).



Fig. 29: High inrush current through the inverter.

Suspecting the problem to be excessive inrush current to the inverter, we probed the inverter when connected to the battery through the isolated DC-DC converter (Fig.29), and confirmed our suspicions. As a temporary fix, we chose to power the

radar directly from the battery, although we are exploring the use of a soft starter to limit the inrush current.

### 3) System Identification

To obtain the hydrodynamic properties of BBASV 3.0, experiments were conducted in the RSYC marina. The ASV was commanded to move at specific velocities for a distance of 50 m, and the vehicle velocity and applied thruster commands were recorded. Previously, a thruster profile was taken for the Minn Kota RT80 thrusters in the swimming pool, and the mapping from thruster command to force was used to estimate the applied force on the vehicle during the runs.

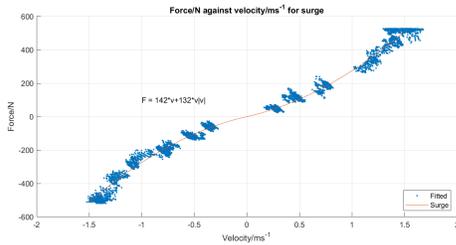


Fig. 30: Force against velocity graph for BBASV 3.0.

The collected data was fitted into the linear and non-linear drag functions (Fig.30) as described by Fossen [1]. Additionally, the transient velocity profile during the acceleration phase of each run was also used to estimate the total mass and inertia of the system, including the added mass and inertia from the water.

The results of the system identification were used as a reference in the feed-forward term of the controller, and for the tuning of the model-based controller. This allowed us to simplify the process of tuning the controller, while achieving better tracking to the reference trajectory.

### 4) Simulation-aided Mechanical Design

To improve the mechanical design of BBASV 3.0, simulation tools on SolidWorks were employed to optimise various aspects of the design.

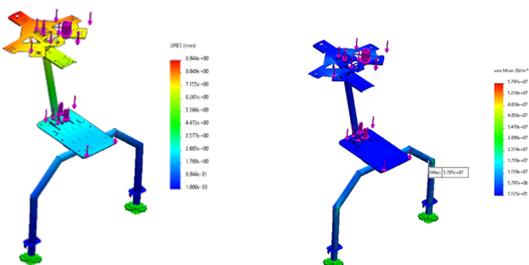


Fig. 31: Displacement (left) and stress analysis (right) of the mast.

From our experience in RobotX 2018, we noted the importance of structural strength for all systems onboard the ASV. To cut down on the wasted cost from excessive iterations of a design, Finite Element Analysis (Fig.31) was performed on the various components of BBASV 3.0, such as the mast and the battery cage. This allowed mechanical problems to be identifier earlier

in the design phase, giving rise to better designs with higher safety factors.

Computational Fluid Dynamics was used to optimise aerodynamic flow over the payload fairing. This allows for the payload fairing to provide maximum shade for the main hull for ideal thermal performance, while reducing the drag induced on the ASV.

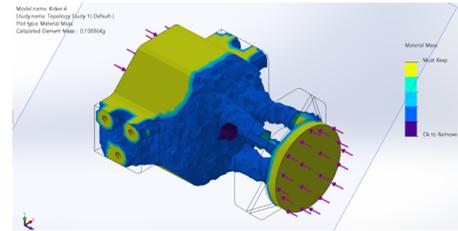


Fig. 32: Topology study on the ball shooter.

Lastly, a topology study (Fig.32) was done on various parts of BBASV 3.0 to improve the weight-to-strength ratio of the parts. Based on the topology study results, designs were changed to remove unnecessary parts to cut down on the weight, while retaining the overall geometry and strength of the part.

### ACKNOWLEDGEMENTS

Team Bumblebee’s development and achievements would not be possible without the help from various organisations and people. The team would like to express their deepest gratitude to the sponsors (Refer to Appendix B), including the Title Sponsors — National University of Singapore (NUS), and Platinum Sponsors — DSO National Laboratories, and Future Systems and Technology Directorate (FSTD). In addition, the team would also like to thank Sport Singapore and the Republic of Singapore Yacht Club for their continuous support.

### REFERENCES

[1] T. I. Fossen, Handbook of marine craft hydrodynamics and motion control. Hoboken N.J.: Wiley, 2021.

APPENDIX A  
SITUATIONAL AWARENESS

To better understand what BBASV 3.0 is doing, several methods are employed. Firstly, the BT-based mission planner (Fig.23) comes with a GUI to allow us to quickly track the current state of the ASV. It also lets us easily check the different outcomes of each state, as well as view mission planner logs to quickly identify any problems.

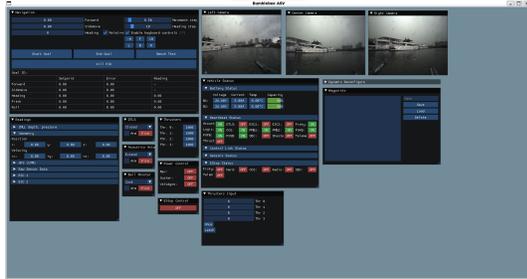


Fig. 33: ImGui-based control panel.

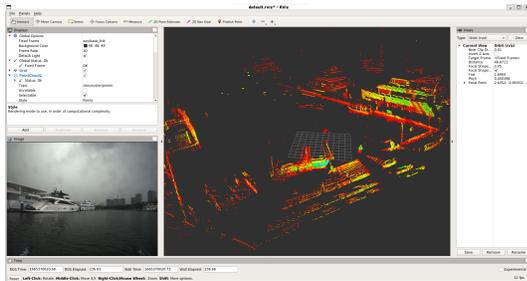


Fig. 34: Using RViz to visualise data.

Secondly, we make use of 2 different UIs to gain a quick overview of BBASV 3.0's status and sensor data. We have an ImGui-based control panel (Fig.33) which provides us a quick overview of hardware statuses, camera feeds, and locomotion goals. This allows us to quickly identify hardware faults and keep an eye on important information such as the battery voltage level. We also make use of RViz (Fig.34) to understand the sensor data recorded by the ASV. This helps us to verify that the ASV's perception of the environment matches up with reality, allowing us to deduce whether problems are caused by faulty logic in the mission planner, or caused by questionable sensor data.

APPENDIX B  
SPONSORS

A. Title Sponsors

NUS (College of Design and Engineering, Innovation & Design Programme, and School of Computing) — For their cash support, equipment procurement, and academic support of our project.

B. Platinum Sponsors

Future Systems Technology Directorate (FSTD) — For cash support.

DSO National Laboratories — For cash support and technical guidance.

C. Gold Sponsors

Fugro, Festo, Cititech Industrial Engineering, Kentronics Engineering, Würth Elektronik, AAEON Technology, SLM Solutions, SBG Systems, Avetics and MacArtney.

D. Silver Sponsors

Bossard, SolidWorks, MathWorks, Southco, Samtec, and Spar-ton.

E. Bronze Sponsors

Edmund Optics and Blue Trail Engineering.

F. Supporting Organisations

Republic of Singapore Yacht Club and Sports Singa-pore.