Development and Testing of the TopCat Autonomous Surface Vessel for the Maritime RobotX Challenge 2016

Andrew Webb, Bradley Donnelly, Jesse Stewart, Jonathan Wheare, Michael Kossatz, Scott Hutchinson, Shane Geyer, Tenzin Crouch
Dr Andrew Lammas, Associate Professor Karl Sammut

***Abstract* - This paper gives an overview of the system design and testing of the TopCat Autonomous Surface Vessel developed by the Centre for Maritime Engineering, Control and Imaging at Flinders University. The paper specifically focuses on the approach to solve the tasks of the 2016 Maritime RobotX Challenge but also provides a broader overview of the development and application of TopCat.**

I. INTRODUCTION

Autonomous Surface Vessels (ASVs) are the future of maritime technology. Their unique ability for extended missions in dangerous areas with improved payload to weight ratios gives them the potential to revolutionise marine operations. To further research in this area a team of staff and students from Flinders University are developing their ASV, TopCat to meet the needs of real world research tasks in addition those of the 2016 Maritime RobotX Challenge. Under the auspices of the Centre for Maritime Engineering, Control and Imaging (CMECI), this vehicle is intended as a research platform for maritime autonomy, situational awareness, and environmental monitoring.

An earlier version of the vehicle was entered in the 2014 Maritime RobotX competition [1] where the team gained experience in the development and operation of an ASV.

RobotX 2016 presents a new level of complexity, with tasks that are not only more challenging, but also interrelated, requiring increased sophistication in autonomy and planning. Specific tasks for 2016 include: demonstrate navigation and control, find totems and avoid obstacles, identify symbols and dock, scan the code, underwater shape identification, find the break, detect and deliver and acoustic beacon based transit [2].

Given the increased emphasis on connectivity and autonomy, the Flinders' team approach was to develop a generalised system which could be adapted to suit both the competition, and other research and data gathering tasks. This involved developing planning algorithms capable of navigation in public waterways and looking at control systems for open-water testing.

After the experiences our team had in 2014, our intended design approach was to develop a stable and reliable hardware platform by early 2016. To this end, TopCat was rebuilt and tested on the water throughout 2015. This testing revealed some key flaws with the vessel including a high centre of gravity and, the ad-hoc nature of some system elements. Therefore, in late 2015 design work started on TopCat version two, with the goal of refreshing the hardware platform to provide a robust and stable base for both the 2016 RobotX competition and future applications.

The structure of this paper comprises of a detailed discussion of our design strategy, a description of our vessel, it's software and hardware design, how it aims to solve the RobotX challenges and finally, a discussion of results from the testing we performed.

II. DESIGN STRATEGY

TopCat has always been a vessel designed for more than just the RobotX challenge and as such the vessel design strategies reflect the overall goal of providing a research platform for government and industry. The key design criteria include;

- ease of use;
- safety;
- security;
- practicality

This has led to a number of basic principles during the development of TopCat.

A. *Design for the General Case*

Where possible, systems should be developed to maximize commonality and the re-use of standard designs. This is reflected in the choice of parts ranging from fasteners to microcontroller boards.

B. *Design for the Worst Case*

An optimal vehicle will have only the structure and equipment required to perform specific tasks. As a research and development platform, the design of TopCat must reflect that not all requirements will be known during the design phase. The hardware and software of TopCat has thus been developed to meet the expected maximum loading and power (worst case) scenarios for future mechanical and power requirements.

C. *Design for Sensor Coverage*

TopCat carries a mixture of active and passive sensors.

These all require maximal fields of view while preventing cross-talk and occlusion. Due to the range of sensors installed, some compromises have been necessary, particularly the view of some sensors is occluded to the rear of the vessel, but the forward and side views are uncluttered.

D. Design for Environmental Protection

The electronics of TopCat are either water resistant/proof or enclosed within watertight containers. All connecters have been selected to be IP67 (ingress protection 67) or higher.

E. Design for Software Modularity

The software for TopCat was developed using the Robotics Operating System (ROS) [3], a commonly used system for research robotics. ROS not only allows access to a wide ecosystem of robotic software, it also allows a tool-based approach to development where individual tasks are factored into separate programs called nodes. These nodes are connected by message passing interfaces called topics. Use of this approach simplifies development in that each node can be developed independently, and also allows testing by transmitting real-world or simulated data into the node's topic interfaces. Our design uses a number of custom ROS message types, allowing descriptive interfaces to be created.

F. Design for Future Applications

Autonomous Vehicles in general have several key components: control, trajectory planning, object recognition, object classification and mission planning. Our approach to these challenges focused on designing systems that would work in a real maritime environment not just the RobotX competition course. Our object classification system aims to be capable of detecting navigation markers, and our trajectory planning to be capable of following international maritime rules.

For the specific challenges of the competition we focused on how these systems could be integrated in a broader context. The Maritime RobotX challenges require capabilities that would be applicable to future ASV applications including autonomous navigation, survey, and firefighting. By approaching such problems in a general manner, we have laid support for future research applications.

III. VESSEL DESIGN

A. Lessons from 2014

The initial design of TopCat in 2014 demonstrated a number of shortcomings. These included issues with CPU cooling, steering, navigation, and image capture. The updated vessel has been designed to address these issues.

B. Vessel Overview

TopCat consists of a hardware platform with sensors, computing, and electric propulsion required to create an autonomous vessel. The overall design of TopCat version two can be seen in Figure 1. Details of the hardware will be covered in the individual sections below.

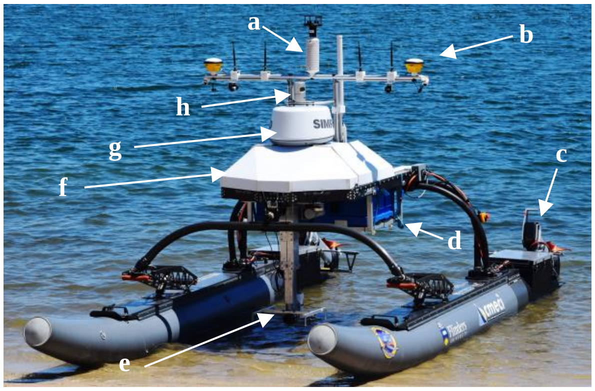

Figure 1. Overview of TopCat's hardware platform

Key:	
a: central mast	e: underwater sensor deployment system
b: sensor and comms array	f: computing and electronics
c: propulsion system	g: marine radar
d: battery system	h: velodyne lidar

C. Sensors

Before it can act, a robotic platform must be able to sense its environment. TopCat uses a number of sensors to estimate its position, and perceive its environment.

The primary navigation system is a dual-antenna Trimble BX982 GPS system. Based on a pair of Trimble's flagship Maxwell chipsets, this GPS is capable of producing a Real Time Kinematic (RTK) solution between its antennae. This allows not only a precise position estimate, but also can estimate the heading, heading rate, and roll of the vessel. Unlike the magnetometer heading sensor used in 2014, GPS is robust to electromagnetic disturbance.

During survey operations, this system has been used with an RTK base station to improve its accuracy, but during the RobotX competition it will operate without external augmentation following competition requirements for operation without external sensor systems. Further rate information is provided by a Microstrain Attitude Heading Reference System (AHRS).

The location of obstacles is found using a combination of active sensors, a Simrad 4G maritime navigation radar, and a Velodyne HDL-32 lidar. The 4G radar is a very low power system that was obtained for the 2014 competition [1]. This system has shown itself capable of tracking compact objects such as marker buoys as shown in Figure 2.

The Velodyne sensor allows the precise location and structure information of objects to be found. Providing 360-degree coverage, this sensor has been found to be particularly efficient in maritime operations as the laser is absorbed by undisturbed water. The combination of radar and lidar sensors allows obstacles to be detected and tracked at range.

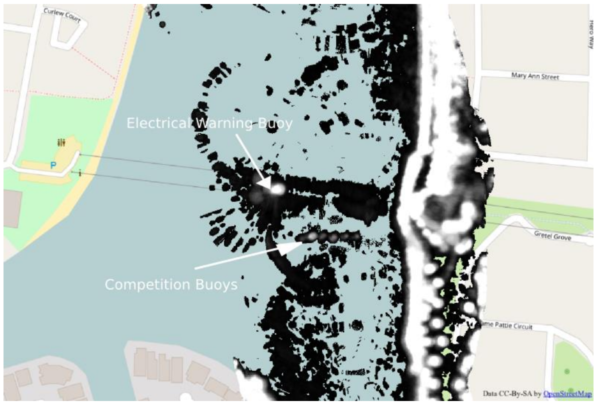

Figure 2. Radar map of West Lakes. Highlighted objects include a power-line warning buoy, and a set of marking buoys. Backing map © OpenStreetMap contributors.

The computer vision system aboard TopCat has undergone significant changes in preparation for the coming competition, both to the hardware used and the approach taken to object classification.

Review of the images taken during the 2014 competition showed that that the combination of wide-angle lens and machine vision camera previously used prevented robust image processing due to difficulties in achieving equalized exposure over the image as shown in Figure 3.

Figure 3. Image of Marina Bay, Singapore, taken with 2014 camera system.

Ultimately this lead to the decision to implement an array of Microsoft LifeCam Studio web cameras. These cameras have narrower fields of view which simplifies the equalization task, however multiple cameras are required to provide the same coverage as the machine vision camera and interchangeable lens. The new cameras were chosen at a low price-point to allow for multiple cameras on board. The smaller field of view also allows a larger down angle on the camera resulting in reduced specular effects as shown in Figure 4. This image shows the same set of buoys that are annotated in Figure 2.

We used a total of five LifeCam Studio web cameras, four above water and one built into a waterproof case to be used for underwater vision tasks.

Figure 4. Lifecam image of buoys in West Lakes northern Boating Lake.

In addition to the underwater camera system, two further acoustic sensors are employed. A mechanically scanning sonar, and a hydrophone array. The mechanical scanning sonar is a TriTech Super Sea Prince. Although it was designed as a navigation sonar for Autonomous Underwater Vehicles (AUVs), this system can also be used for the measurement of bathymetry. In the RobotX competition, this will be used for underwater search tasks.

The hydrophone array uses a quartet of hydrophone sensors for the localization of acoustic beacons. This will be covered further in the hardware section.

The WAM-V platform is designed for operation in littoral and surf zone environments, where a shallow draft minimises potential groundings. Additionally, TopCat's emphasis on clear sensor views has resulted in a tall vessel with correspondingly large windage. This combination of high windage and low draft has meant that the TopCat vessel is susceptible to drift and weathercocking during operations. To counter this, systems for producing navigation state estimates, guidance, and control are required. A Windsonic acoustic wind sensor allows the vessel's control system to measure and compensate for these wind effects.

D. Batteries and Propulsion

The battery systems are identical to the solution used in 2014 [1] and comprises two 3.88kWh LiIon batteries.

TopCat's propulsion is provided by a pair of Torqeedo Cruise 2R electric outboard motors whose tiller angles are controlled via Max Jac linear actuators. These are controlled by the Actuator Control Module (ACM), which consists of a Digilent Cerebot MC7, a 24V DC/DC converter to power each Cerebot's PWM controlled H-bridge motor controller, and an RS485 PMOD for communication with the Torqeedos. The H-bridge system drives the linear actuator bi directionally at a variable speed. The relative position of the actuator is determined by the Cerebot reading and integrating the quadrature encoder output of the actuator. This relative positon is resolved to an absolute position by the use of a magnetic limit switch.

Updated steering, control and communications will be covered in their corresponding sections.

E. Hardware Platform

The base platform for TopCat is the same innovative Wave Adaptive Modular Vessel (WAM-V) [4] provided for the 2014 RobotX competition. However, the post competition sensor upgrades and results of on water testing required upgrading of mounting systems.

The main objectives of the hardware upgrades were to:

- lower the vessel Centre of Gravity (CoG)
- maximise sensor coverage
- prepare the vessel for RobotX 2016 and beyond,
- enable tiller angle control

To lower the vessel CoG while maximising sensor coverage a system was developed to sling the batteries under the main payload tray. Along with a winch, this system significantly lowered the CoG and reduced manual handling.

The lowering of the batteries also allowed a new sensor mounting structure to be developed. In order to maximise sensor coverage a complete CAD model of the vessel and each sensor was built as seen in Figure 5. This allowed the sensor field of view to be visualised and interference minimised.

Due to the WAM-V's modularity and compact shipping ability it was decided to design the hardware upgrades to be flat packable. Retaining strength with this design criteria was challenging but ultimately very beneficial.

The new system, seen in Figure 5, has a main sensor bar with the dual GPS antennae, four 900 MHz telemetry antennae, the visual indicator and, the wind sensor. Below this bar sits the Velodyne lidar and the Simrad radar on separate levels. In these positions the lidar has maximum view of on water targets, as does the radar. The wind sensor has clear access to the undisturbed wind from all directions and the GPS and critical communications have minimal interference from other sensors or structures. Finally, for the underwater sensing tasks, the sensor deployment arm from 2014 was moved forward on the payload tray to accommodate the batteries.

Figure 5. Complete CAD model of the vessel and sensors

To mount the electronics and computing systems a bolted rail system was used. This securely mounts the CPU and the electronics housing. The rigid mounting structure provides increased safety and security as well as allowing for a coroplast cowling to increase heat shielding and weather protection.

1) Tiller Angle Control

To allow tiller angle control it was necessary to design a mount and linkage for the Max Jac linear actuators. A simple mount was developed to place the linear actuator parallel to the axis of rotation. Then a linkage was attached to enable the existing Torqeedo steering rod to be connected to the linear actuator. Although the linkage added several degrees of freedom to the system, it enables the motors to be lifted out of the water without disconnecting the steering.

Due to the additional weight of the steering system, it was necessary to add buoyancy to the engine modules. However due to the available shipping space it was necessary to develop a modular, removable section. These extension modules were cut out of high density foam and covered in carbon fibre.

2) RobotX 2016 Additions

In addition to the required mounting upgrades it was necessary to develop a ball launching mechanism for the detect and deliver task.

Due to strict firearms restrictions in Australia the only option available without a firearms license was mechanical launch.

Our design was modelled on a ball launcher placed on a rotating turret such as those used for sporting applications. However, after construction the overall size and weight of this system required that it be placed at the rear of the vessel and thus limited targeting and stabilisation options. Therefore, alternatives will be considered prior to the competition.

F. Communication

TopCat uses three communication buses, Ethernet, Universal Serial Bus (USB), and Controller Area Network (CAN) bus. Communication with the battery and engine modules is across CAN while sensors are connected via a combination of Ethernet and USB. The host computer connects directly to the USB and Ethernet buses while CAN data is interpreted via a core microcontroller.

Communication channels from vessel to shore were selected based on legal and operational requirements. For teleoperation and critical communications multiple 900MHz radio modems were selected. Australia and the USA both permit use within the 900MHz Industrial Scientific Medical (ISM) band though Australia's ISM frequency allocation is smaller than that of the USA. Therefore, it was possible to configure the radio modems to be compliant with both USA and Australian regulations. These radios were tested with a spectrum analyser and found to be within the expected frequency range.

For data transfer and high bandwidth communications a point to point 5GhZ WiFi link was selected. This system uses two Ubiquiti Rocket M5 modems with an Omni and Sector antenna. The system can be configured for use around the world and is set up to meet both USA and Australian frequency and power specifications.

G. On Board Electronics

Design of the electrical system of TopCat V2 began with a review of the electrical system developed for the 2014 version of the vessel. Many parts of this system, particularly the power distribution and supply elements, were found to be acceptable.

The primary area identified for improvement was the e-stop, teleoperation, and CAN communication systems. These were based on microcontrollers that had a limited amount of available memory. Review of the available design showed that the Digilent Max32 PIC based microcontroller had not only the memory to support the required functionality, but also supported two CAN networks.

A shield Printed Circuit Board (PCB) was developed for the Max32 that supported interfaces for communication with the engine modules via CAN, base station via 900MHz radio, and controlled relays for the e-stop system.

H. Computer System

Due to the hardware failure our computer suffered during the 2014 RobotX competition we required a new computing solution. Our key design criteria for the new system included required processing power, ability to interface with on-board electronics, and effectiveness of cooling. The solution we developed was a custom-built metal CPU case with a custom heatsink to transfer heat to the outer case which could operate as a huge heat sink. The design carefully isolated the CPU electrically while ensuring heat conduction occurred and mechanical loading was minimised. In this box an I7 mini ITX board with 16GB of RAM and a 250GB SSD were added. Additional space was left for the later addition of a second smaller CPU or extra storage drives.

I. E-stop and motor isolation system

As the competition rules require a method for removing power from the motors remotely and independently of the teleoperation system, a second board was also populated to be used as a parallel kill system. This system controls normally-open relays in the engine modules. Power is only available to the motors and steering system while these relays are energised. A pair of mechanical e-stop switches are wired in series with the relays. Actuation of either of these switches will also cause the relays to open.

An independent power supply circuit connected to a set of dry-cell batteries allows the Core and Kill boards to operate while main power is unavailable.

J. Acoustic Beacon Localisation

In the 2016 RobotX competition, for the robot to enter the course, the correct entrance gate must be identified by detecting an acoustic beacon. To solve this, the location of the acoustic beacon is determined using a modified version of the acoustic beacon localization platform developed in 2014 [5]. This platform consists of 4 Teledyne Reson hydrophones, arranged in a cruciform, connected to a pre-amplifier/filter board, which prepares the signal so it is compatible with the analogue to digital converter (ADC) on a Digilent Nexys4 Field Programmable Gate Array (FPGA) board. The FPGA is currently fitted with hardware to perform cross-correlation on the incoming signals, control the pre-amp board and send data to the main system.

From the cross-correlation of the signals, Time Difference of Arrival (TDOA) can be obtained. These TDOAs are used by a particle filter to abstract out the relationship between the TDOAs and the acoustic beacon location whilst also being resistant to echoes and reverberations. More detail on the particle filter will be covered in the software section.

K. Visual Indicator System

A new requirement for RobotX 2016 is that the vessel must have a visual indication system capable of being seen at 150m range [6]. Initial experiments with a number of indicator lights showed that reliable viewing in sunlight at this distance was a challenging problem. To produce a sufficient level of illumination, a strip of Adafruit Neopixel Light Emitting Diodes (LEDs) was chosen. The illumination system and its power supply draws approximately 24-30Watts of power when operating.

To ensure stable output, the system is driven by a dedicated microcontroller connected to the kill system.

IV. SOFTWARE DESIGN

The software design for TopCat follows a three-layer model, with the top level consisting of mission planning software, the mid-layer consisting of navigation and guidance, and the lowest level containing control firmware. The top and mid-level systems are implemented as ROS nodes, while the lowest level firmware mostly uses the Arduino framework [7].

A. Software Design - Mission Planning

The Mission Planner node is responsible for determining TopCat’s mission and the scheduling of all associated tasks. In the case of RobotX 2016 this requires a system capable of selecting which task to execute and passing or storing task information. This system is especially important in the semi-finals and finals where completing certain tasks can provide information about other tasks.

To facilitate a generalised, modular system which can be used beyond RobotX 2016 the mission planning system was broken into two segments, the top-level mission planner and individual task planners. The mission planner simply connects task planners and selects desired tasks while the task planners are responsible for execution of specific tasks.

The top-level mission planner was developed as a ROS node using the publisher/subscriber model. The task planners publish criteria states and the mission planner listens to these messages, giving permission to execute only when a task planner has enough information. The interconnection of mission and task planners is shown in Figure 6 below.

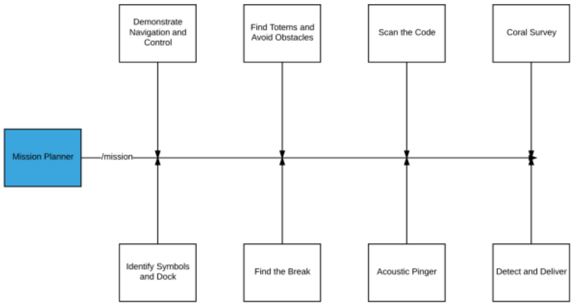

Figure 6. Mission and task planner interconnections. The task planners (white) connect to the mission planner via the '/mission' topic.

The main decision making that the mission planner must perform is selecting which task to perform next. It must only allow one task to execute at a time and requires the ability to cancel tasks if an error occurs. This communication style is already implemented in ROS as the actionlib library. Using this library, the mission planner gives the task planners permission by sending a *'goal'*. The task planner executes the task and the mission planner waits for the completion confirmation. Figure 7 demonstrates this state transition:

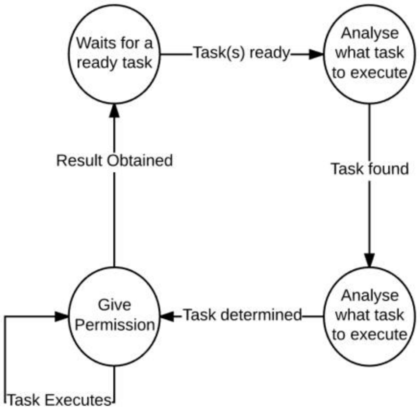

Figure 7 - Mission Planner States

However, this solution is only efficient, if there is only one task ready to execute at any given time. To overcome the issue of multiple ready tasks a greedy model solution was implemented. Each task planner calculates a 'cost of execution' based on the time required to execute, the quantity of information known about the task and the value associated with the specific task. This cost value is sent to the mission planner, allowing it to select the cheapest task in the event that there is more than one task to choose from.

B. Software Design - Object Recognition and Classification

For an autonomous surface vessel to successfully navigate its environment, it must have access to a live map of surrounding features and a robust estimation of its location amongst them. Existing maps of the world's waterways can provide useful static data, but cannot reliably account for the unpredictable nature of dynamic surface features. Safe operation of the ASV requires a constantly updated map of the environment to be constructed using sensor measurements and vessel navigation data.

The most important factor to consider when trying to solve the mapping problem is the operating environment of the robot. The ASV will navigate waterways that typically consist of sparsely distributed and highly dynamic surface-features. Objects that would generally be considered static such as buoys are still subject to drift with current and will require a robust data association process for consistent mapping. The identification of surface-features is also crucial in marine environments as foreign vessel class and various types of navigation markers dictate the appropriate manoeuvres for compliance with international maritime regulations.

Developed as part of our ongoing research, the object localization system is also designed to consider the possibility of operating in a GPS denied environment. As such, it will not only produce an estimate of an object's position, but also the vessel's. This is referred to as a Simultaneous Localisation and Mapping system (SLAM).

The FastSLAM algorithm [8] was chosen as the most suitable approach to solving the SLAM problem for these operating conditions. FastSLAM maintains a feature-based map where identified objects are described by type and location. This is a memory-efficient representation for a sparsely-populated environment and offers great capacity for maintenance of object classifications. The FastSLAM algorithm also offers a more robust solution to the data association problem than other SLAM algorithms such as Extended Kalman Filter (EKF) SLAM [9].

The algorithm is implemented using the Point Cloud Library (PCL) [10]. The Object Detection node forms the first stage of the mapping system and aims to segment objects of interest from point cloud measurement data provided by a lidar sensor.

Given a new point cloud measurement, the Object Detection node must first remove unwanted data returned from features beyond the shoreline. This issue is addressed by defining a polygon that encloses the ASV's operating field, and using a point-in-polygon algorithm [11] to remove any points that fall beyond it. An additional source of unwanted data includes points returned from disturbances in the water's surface. Analysing average point cloud intensity values reveals that water surface returns have a significantly lower average intensity value than points returned from physical features. These points are thus filtered out based on a minimum intensity threshold.

After removing the unwanted data, a point cloud clustering algorithm [12] is used to separate features of interest based on distance thresholding. Bounding box representations of the segmented clusters are constructed and published as a list of landmark-measurements to the SLAM node and used to update the current map. This process is shown in Figure 8.

Experimental classification of complex 3D shapes, such as the docking platform is performed using Iterative Closest

Point (ICP) algorithm [13] to compare point cloud clusters to local template models. The algorithm inspects the point cloud cluster in search of unique key-point features and iteratively attempts to match them to the template. The final result is a best-fit score and a transformation matrix that describes the estimated orientation of the observed feature.

Given that the environment is expected to be sparse, the map is published as a list of objects. Each object contains not only a unique identifier and position, but is also tagged with zero or more identifying parameters such as type, colour or shape.

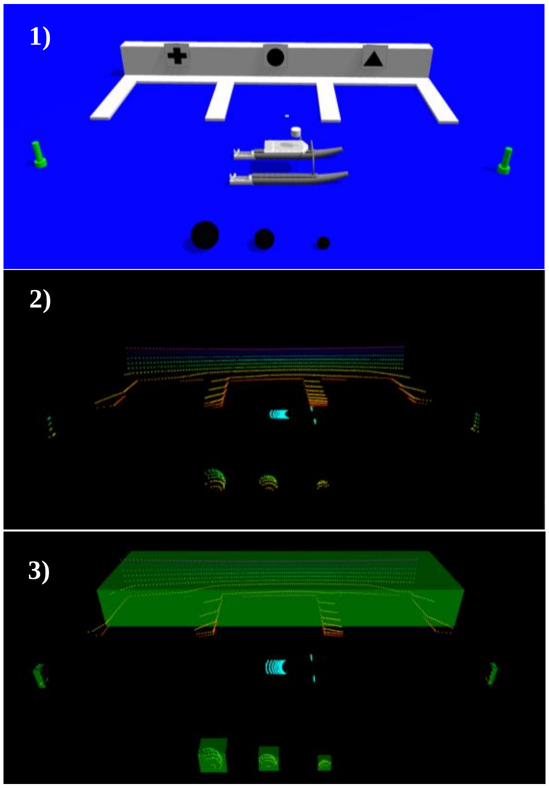

Figure 8. Process of measuring, segmentation and clustering point cloud data. 1) shows the vessel detecting near a dock and buoys in the simulation environment, 2) shows the resultant lidar data prior to segmentation and clustering, 3) shows the results of clustering.

C. Software Design - Computer Vision

The computer vision system has been designed to implement a series of specific functions, each tuned for an individual task. These specific functions are activated through the different task planners as they are required. The specific functions currently implemented include object classification, docking and target sign identification, underwater imaging and light buoy sequence extraction.

The image based object classification has been designed to work in conjunction with the object detection system. Rather than performing a search of the visual field for the expected objects, the image classification system is cued using a bounding box to the expected position of the object. This bounding box can then be transformed to the camera frame and used to isolate a region of interest (ROI) within the camera frame where the object of interest is located. This approach allows more invasive thresholding processes to be applied to the camera frame to extract the objects, and reduces the computational requirements for each frame. The shape and colour identification functions common throughout the system can then applied to the extracted object.

The shape classification functionality of the system currently implemented has maintained the approach used for the 2014 competition [14]. Once the object or shape of interest is extracted from the image frame the blob area, enclosing circle area, and convex hull area are deduced. The relationship between these three attributes are then used to identify its shape.

With the increased emphasis placed on colour for the 2016 RobotX competition a robust method of colour identification is critical. First the centre of the object or shape is identified following the thresholding and extraction phase. A sampling window is then applied at the centre of an object, averaging colour within the sampling window. Once the average colour has been found it is then compared to a set of predefined colours to determine the difference between the observed and expected colours. This difference value is then used to determine a confidence factor in the colour which can be integrated with the confidence factor in the higher-level classification system.

The docking and target sign extraction for the docking task exclusively uses image based classification, accomplished through segmentation in the Hue Saturation and Variance (HSV) colour space. Once the shape is extracted from the image frame, the shape and colour identification approached mentioned previously are used to classify the shape.

Classification of the underwater symbols used in the coral survey task is done through the application of an equalizing filter biased towards white on a grayscale image, maximizing the contrast between the sea floor and the shape.

Finally, for the scan the code task a colour sequence extraction function was implemented, capable of observing the colour sequence displayed on the light buoy. The light buoy code extraction program is activated through the task planner for the scan the code task, scanning the code until the code is successfully read.

Classification information from all elements of the computer vision system is submitted to the object tracking system with confidence values. Our decision to move towards a likelihood based image processing system was led by the challenges associated with image processing in different conditions as well as the experience we have with Bayesian filtering techniques developed from our centre's work on navigation state estimation [15].

D. Software Design - Acoustic Beacon Localisation

The course area is shallow, with an expected maximum depth of less than three metres. Acoustic beacons in this environment are expected to product a significant amount of multipath reflections, reverberations and echoes. Therefore, simply driving towards the strongest signal is likely to produce incorrect results. In order to overcome this issue and fuse multiple sensor readings a particle filter will be used.

A particle filter provides two main benefits. It is capable of incorporating an arbitrary number of measurements and, depending on configuration, is capable resolving ambiguity in the measurements. The former is important as the received measurements are likely to contain significant noise levels and some spurious measurements. Therefore, to achieve an accurate and stable solution many measurements must be combined. The second benefit is particularly useful in this context as ambiguity can come from multiple sources. In normal cases ambiguity comes from limitations in the measurements i.e. any individual measurement does not contain enough information to find a unique solution however in the context of the competition the main source of ambiguity will be multipath effects.

Generically, a particle filter is a state estimator that infers the estimate, updating a set of weighted "particles" in lieu of processing the entire state space [16]. The particle filter consists of two phases, prediction and correction. The prediction phase "moves" the particles in the set according to the knowledge of the behaviour of the system being estimated. In the correction phase weights of the particles are updated according to the correlation between the particle state and the current measurement.

The particle filter implemented in this context is a bootstrap filter [17]. Since the targets that are being estimated are stationary, the prediction stage consists of adding uncertainty to the particles without any gross motion. The correction phase updates the weight of the particles via a causal measurement model. This model is the critical feature that enables the filter to maintain and when possible resolve an ambiguous solution. A causal measurement model uses a predictive principle by asking: "if the value of the state of each particle is known what would the measurement be?" By comparing the predicted measurement from a given particle with the actual measurement, the correlation between the particle and the measurement can be calculated.

E. Software Design – Path Planning

Ground vehicles commonly use grid based systems for the storage of spatial data, with path planning performed using search algorithms such as A*. In a maritime environment, landmarks such as buoys can be treated as point sources. Thus, a landmark based system for the storage of spatial relationships is attractive.

In such a landmark based system, the task of path planning remains necessary. A technique that has been used successfully in the generation of topological maps is the use of the Voronoi diagram [18], the locus of points that lies equidistant between pairs of nearest points. Use of the Voronoi diagram to generate paths thus has the advantage in that it maximises clearance between obstacles and the path. However, in the maritime environment, obstacles are sparse and have specific functions, including marking channels and exclusionary areas. For this purpose, the Delaunay triangulation, the dual of the Voronoi diagram, has a number of advantages. Delaunay has been used to generate paths for underwater vehicles [19], but the most interesting property of the triangulation in this case is the ability to build Constrained Delaunay triangulations. In this form of triangulation, pairs of vertices can be constrained to produce an edge. Using the information from the object tracking system, edges can be built based on the object's properties.

Once the triangulation is complete, a graph is built of the environment. This graph is searched using the shortest path function provided by the scipy library [20]. The resultant sequence is then used to build the dual Voronoi path.

F. Software Design – Guidance, Navigation and Control

Due to the requirement for increased manoeuvrability and resultant addition of tiller steering it was necessary to significantly modify the vessel control system. A research project was undertaken to investigate the current state of the art in ship navigation and control. This is well summarised in the Handbook of Marine Craft Hydrodynamics and Control by Fossen [21]. Combined with the modularity of the existing ROS architecture this research led to the ambitious development of a multi component guidance, navigation and control (GNC) system as shown in Figure 9.

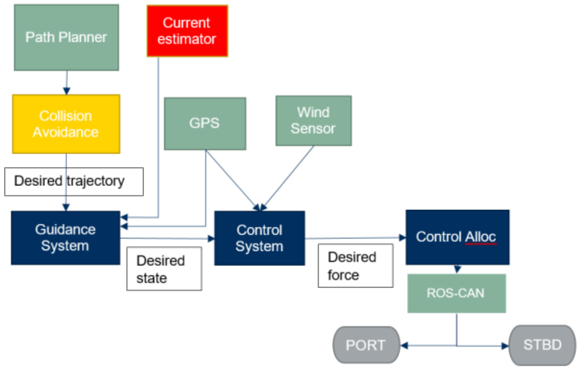

Figure 9. Block diagram of guidance, navigation and control system where blue designates software nodes in ROS, green denotes existing sensors and software, yellow and red denote future work.

In this system, a path planning node will pass a desired trajectory to the guidance controller. The guidance node will then calculate a desired state to ensure the vessel moves along the given trajectory. The main control system takes the desired state and calculates the desired forces on the vessel. Finally, the control allocation system will convert the desired forces to actuator commands. While seemingly complex to implement the major advantage of this modular system is that individual elements can be easily modified and new algorithms tested while maintaining comparable data for research.

Given the plans to use TopCat for research beyond the RobotX competition the GNC system design looked at path following of curved trajectories using Serret-Frenet frames such as in [22]. This system was tested in Matlab however it was not implemented for RobotX due to the limited benefits it would provide given the path planner currently provides straight line trajectories. In its place a simple lookahead steering algorithm [21] was modified from the existing control system.

The control node itself uses a model based proportional, integral, derivative (PID) controller based on the system in [21]. Using model based controllers allows for an initial PID controller to be tested with no model (a model set to zero) and when a model can be generated from test data it can be easily added to improve the control system. This model takes into account the vessel hydrodynamics and local environmental forces.

The final element of the control system is the control allocation node. This was especially important given that the previous work into control allocation for 2014 [23] could provide impossible force allocations due to matrix singularities and therefore needed a complete revisit.

The problem of control allocation for over actuated vessels can be described as selecting the best set of values to achieve the desired trajectory and meet given criteria such as minimum power usage. This is in essence a constrained optimisation problem and although computationally challenging it is not impossible due to the small number of states involved.

The research in [24] provides a way to implement the optimisation in practice by using reasonable assumptions about the vessels motion to simplify the problem to a convex quadratic programming problem. This system was implemented in Matlab and later ROS using the Computer Graphic Algorithms Library (CGAL) [25] to solve the optimisation defined in [21] and [24].

Initial testing showed the complete control system communicates correctly however further testing is required to ensure that the output values are in the expected range.

G. *Software Design – Low Level Firmware*

1) *Safety System*

Any robot of non-trivial scale needs careful attention to ensure that it only operates when safe. Since much of the software and hardware on TopCat is under development, the human operators must be able to halt the vessel at any time if it begins operating outside the expected parameters.

With this in mind, a derivative of the 2014 tele-operation system was developed [26]. Referred to as the estop system, this uses the MAVLINK [27] protocol to send heartbeat and control packets across a set of 900MHz ISM band radio modems between the base station and vessel. If the vessel is manually halted, or the heartbeat connection times out, the vessel will transition to a LINKLOSS state and send zero throttle commands to the engines.

A parallel system uses another set of 900MHz radios to communicate with a second board. This KILL system controls a set of relays that are capable of isolating primary power to the motor and steering systems. In the STOP or LINKLOSS states these relays are opened.

Motion of the vessel requires both ESTOP and KILL systems to be in the driving state.

The 2016 RobotX rules indicate that the vessel's visual indicator must show when the motors are isolated from power, but it must also show when the vessel is in AUTO mode. As such the kill board is also connected to the CAN bus. During mode change by the e-stop between manual and auto, a message is sent to the kill board informing it of this change. This allows the visual indicator to alter to reflect this fact.

Table I
SAFETY SYSTEM VISUAL INDICATOR STATE DIAGRAM
THIS TABLE OUTLINES THE DETAILS THE SAFETY SYSTEM STATES AND CORRESPONDING VISUAL INDICATOR.

		Core board State			
Kill Board State		LINKLOSS	STOP	MANUAL	AUTO
	LINKLOSS	Red	Red	Red	Red
	STOP	Red	Red	Red	Red
	MANUAL	Yellow	Yellow	Yellow	
	AUTO				Blue

2) *Actuator Control Module (ACM) Firmware*

The ACM receives commands from the core board via a CAN communication bus running at 250kbits/s utilising the extended address size. The CAN protocol was chosen as it is a robust fault tolerant serial interface utilised in similar environments such as road vehicles. Due to the ad-hoc nature of the CAN bus nodes on the vessel additional fault tolerance characteristics have been added. These include recovery from errors such as packet collision and buffer overflows.

For the purposes of safety, the firmware initialises the actuator control systems in a disabled state on boot. The only way to make the system active is to have the core board send an enable message to the ACM. The core board can also cease motion at any time by sending a disable message.

Once enabled, the steering will not respond until the actuator position has been calibrated by putting the ACM in calibration mode and moving the actuator until the limit switch has been reached at which point the actuator stops. When releasing the control and calibration mode the steering turns to 0 degrees. This calibrated state is persistent until cleared by the core board or powered off.

At this point the system ready to use but it must receive a stream of control messages from the core board to operate. If the ACM does not receive a control message within 1 second of the last message, all actuator motion ceases.

The Torqeedo Cruise 2R's are controlled by translating the motor speed provided by the core board via the CAN interface to Torqeedo's proprietary serial interface. This serial interface is configured to maintain constant communication with the Cruise 2Rs, if this communication is lost and cannot be recovered, the ACM informs the core board of a STOP condition.

H. *Experimental Results*

Earlier versions of the TopCat vessel have been extensively tested at West Lakes, an artificial lake connected to the ocean. The latest revision of the vessel was designed based on these tests however manufacturing delays have prevented full testing of the updated vessel.

1) *Navigation and Control Performance*

Since TopCat is being developed for environmental monitoring tasks, one of the areas of interest is the ability to make reliable and repeatable measurements over specified areas. Much of the on-water test time has concentrated on testing the navigation state estimation, modelling of the vessel performance, and testing of guidance and control algorithms.

This on-water testing demonstrated that vessel control under differential drive was precise at low velocity, but the effectiveness of steering with differential drive dropped rapidly with increasing velocity. This drove the development of the updated steering system, and new guidance and control.

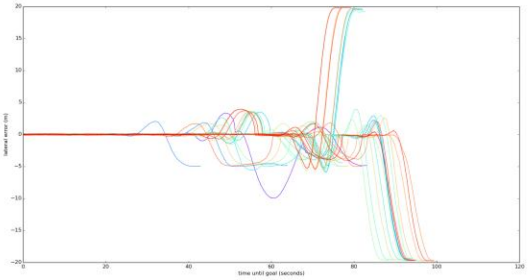

Figure 10 - TopCat lateral error under control of Model Predictive Controller. Image source [28]

During 2015, a guidance system based on a Model Predictive Controller [29] was developed [28]. This was shown to control the vessel with small lateral error as shown in Figure 10.

2) *Simulator*

While field testing is vital to ensuring the reliability of a platform such as TopCat, in-water time is too precious for software debugging. Hardware in the loop testing can allow the isolation of driver faults, but another method is required if software integration is to be possible without wasting on water time.

To resolved this issue our Gazebo [30] based simulator, initially developed in 2014, has been updated to reflect the vessel's new sensor fit. This includes a custom-made simulation plugin for the dual-antenna GPS. For 2016, the simulation has also been updated with new course elements as shown in Figure 11.

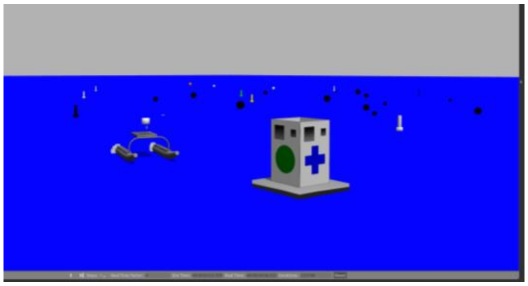

Figure 11 - TopCat on simulated course in Gazebo

The simulator has been useful in allowing debugging of both control and mission planning software before the vessel is transported to its operating area. While the simulator cannot model the performance of the vessel dynamics exactly, as shown in Figure 12, the vessel dynamics are similar.

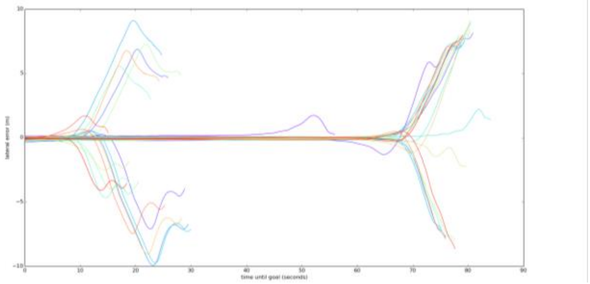

Figure 12 Simulated TopCat lateral error under control of Model Predictive Controller. Image source [28]

3) *Water Testing with New Version of TopCat*

While a full test of the vessel on a physical competition was not possible due to time and budget constraints, some elements of the competition course were replicated (see Figure 13).

These items were set up in the test area and the vessel manually driven around them in order to gather data for testing and simulation.

Figure 13. Test equipment built for data gathering and testing.

I. Acknowledgements

The authors would like to acknowledge the support and encouragement received from our generous sponsors: DST Group, SAAB Australia, Fragment Films, UPG, Flinders University Student Association, the Field Robotics Club, Torqeedo, Teledyne Reson, Observator Instruments and Thomson Linear Motion.

We would also like to acknowledge the City of Charles Sturt council for the permission to use West Lakes for vessel testing and Scouts SA – Rovers for the loan of equipment for testing

Finally, we would like to acknowledge all of the students and staff who have volunteered their time to assist with the development, promotion and testing of the vessel.

V. REFERENCES

[1] K. Sammut, J. Wheare, A. Lammas, R. Bowyer, M. Anderson, T. Arbon, B. Donnelly, R. Peake, T. Crouch, R. Pivetta, J. Renfrey, T. Wooldridge, S. Stevens, A. Webb, A. Forrest, J. Keane, H. Hubbert, and R. Kent, "Maritime RobotX journal paper, Flinders University / Australian Maritime College – Team Topcat," 2014.

[2] M. RobotX, "2016 Maritime RobotX Challenge Preliminary Task Descriptions," 2016.

[3] M. Quigley, B. Gerkeyy, K. Conleyy, J. Faust, T. Foote, J. Leibsz, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," 2009.

[4] Marine Advanced Research, "WAM-V." [Online]. Available: http://www.wam-v.com/.

[5] B. Donnely, "Design of an acoustic underwater pinger locator system," Flinders University Honours Thesis, 2014.

[6] Maritime RobotX, "Preliminary Visual Indicator Specifications," 2016. [Online]. Available: http://robotx.org/images/files/RobotX-2016-Visual-Feedback-Document-2016-06-22.pdf.

[7] Arduino LLC, "Arduino." [Online]. Available: https://www.arduino.cc/.

[8] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Aaai/iaai*, 2002, pp. 593–598.

[9] P. Cheeseman, R. Smith, and M. Self, "A stochastic map for uncertain spatial relationships," in *4th International Symposium on Robotic Research*, 1987, pp. 467–474.

[10] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," *ICRA Commun.*, 2011.

[11] T. Wright, *Fundamental 2D Game Programming with Java*. Belse Education, 2014.

[12] R. B. Rusu, "Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments," *KI - K{ü}nstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

[13] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, "Registration with the Point Cloud Library PCL A Modular Framework for Aligning 3D Point Clouds," pp. 1–13.

[14] T. Arbon, "Maritime Robot X: Image processing," Flinders University Honours Thesis, 2014.

[15] A. Lammas, K. Sammut, and F. He, "6-DoF Navigation Systems for Autonomous Underwater Vehicles," in *Mobile Robot Navigation*, Intech Press, 2010.

[16] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo methods in practice*, Springer, 2001, pp. 3–14.

[17] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proceedings F-Radar and Signal Processing*, 1993, vol. 140, no. 2, pp. 107–113.

[18] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artif. Intell.*, vol. 99, no. 1, pp. 21–71, 1998.

[19] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane, "Path planning for autonomous underwater vehicles," *IEEE Trans. Robot.*, vol. 23, no. 2, pp. 331–341, 2007.

[20] E. Jones, E. Oliphant, P. Peterson, and E. Al., "SciPy: Open source scientific tools for Python.," 2001. [Online]. Available: http://www.scipy.org/.

[21] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. 2011.

[22] M. Bibuli, M. Caccia, and L. Lapierre, "Path-following algorithms and experiments for an autonomous surface vehicle," *IFAC Proc. Vol.*, vol. 7, no. PART 1, pp. 81–86, 2007.

[23] M. Anderson, "Model-Based Control and Control Allocation System for a Wave Adaptive Modular Vessel," Flinders University Honours Thesis, 2014.

[24] T. a Johansen, T. I. Fossen, and S. P. Berge, "Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming," *Control Syst. Technol. IEEE Trans.*, vol. 12, no. 1, pp. 211–216, 2004.

[25] L. Fisher, B. Gartner, S. Schonherr, and F. Wessendorp, "CGAL 4.9 - Linear and Quadratic Programming Solver." [Online]. Available: http://doc.cgal.org/latest/QP_solver/index.html.

[26] R. Peake, "Maritime Robot X: GIS integration with ROS and Improved E-stop and tele operation," Flinders University Honours Thesis, 2014.

[27] qgroundcontrol, "MAVLink Micro Air Vehicle Communication Protocol," 2014. [Online]. Available: http://qgroundcontrol.org/mavlink/start.

[28] J. Wheare, "Mission planning for field robots using symbolic planning and topology," Flinders University Ph.D Thesis (submitted for examination December 2016), 2016.

[29] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[30] N. Koenig, "Design and use paradigms for gazebo, an open-source multi-robot simulator2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)," vol. 3, p. 2149, 2004.

[31] S. M. Zadeh, D. M. . Powers, and K. Sammut, "An autonomous reactive architecture for efficient AUV mission time management in realistic dynamic ocean environment," *Rob. Auton. Syst.*, vol. 87, pp. 81–103, 2017.

VI. APPENDIX—SITUATIONAL AWARENESS

Autonomous vehicles are composed of a large number of individual computation entities. The interactions between these entities make it difficult to predict the exact behaviour of the system as a whole because of the large number of possible non-deterministic ways in which the system can behave. As the application domains for autonomous vehicles grow more challenging, the need for self-adaptation to unforeseen (at planning time) situations encountered in dynamic environments, becomes increasingly necessary. The system is vulnerable to measurement uncertainties, weather, environmental hazards, and equipment failures. As a consequence, as autonomous vehicles develop to higher levels of autonomy they become more intimately linked with the notion of emergent behaviour. We are currently working on and considering a number of approaches to ensure that the vehicle's behaviour performs as intended and does not deviate from the margins of its trusted behaviour. There are three parts to our work.

A front-end user interface is an obvious requirement for most remotely operated robots. In our case, it meets several needs for the vehicle, delivered in the form of multiple widgets displayed on the operator's console, with the primary end goal of the software being to give the operator some situational awareness for the vessel. The first concern addressed for the end user is what the vessel is currently doing. This is achieved by displaying the heading and speed of TopCat itself as well as live depth animation and ability to show pitch and roll. Also required for the operator is a way to monitor the status of the on-board equipment. This is provided in a tabbed status panel and covers all of the information required for the user to be confident in knowing which state each element is in, at any given time. There is a large amount of data being gathered by on-board sensors such as Radar, Lidar, Sonar etc. and this is visualized, again for the operator to know what the vessel can "see" currently and any objects nearby that may be of interest in the research applications. Finally, the interface provides prominent warnings that alert the operator to any faults or errors that present during operation. These are then mirrored in the status panel to doubly provide feedback and alerting that the warning requires attention to dismiss or resolve. Overall the developed interface software suits specific needs addressed by the team and was developed in conjunction with the rest of the project. It provides continuous vigilance for the remotely operated vehicle.

To provide a higher level of trusted autonomy, we have been developing a cooperative hybrid architecture to enhance a vehicle's capacity in decision-making and situational awareness. The cooperative hybrid architecture [31] comprises a high-level task-organizer/mission-planner and a low-level on-line path planner, and is designed to provide comprehensive control of mission time management while performing tasks in the optimal sequence within the available time, while ensuring safe deployment at all stages of the mission. In this respect, the top-level module of the architecture is designed to assign the prioritized tasks in a way that the selected edges in a graph-like terrain lead the vehicle toward its final destination in predefined restricted time. Meanwhile, the lower level module handles safe deployment along the selected edges in the presence of environmental disturbances, in which the generated path is reactively corrected and refined to cope with unforeseen changes of the terrain. To handle the complexity of NP-hard graph routing and task allocation problem, the top-level planner continually utilizes the Ant Colony Optimization algorithm to dynamically find an optimum order of tasks for the overall mission, while for the lower level planner, the Firefly Algorithm is used to carry out collision-free path planning between pairs of waypoints. This work has been developed in the context of autonomous underwater vehicles but will be adapted for use on the autonomous surface vessel.

A third part and future direction for our work is to explore more qualitative ways to validate the behaviour of the autonomous vehicle. So far testing has been conducted simply as a as a cycle of system-level test-fail-patch-test procedures. A more methodological approach will be required before the autonomous vehicle can be safely deployed with the operator completely out-of-the-loop. While it is clearly impractical to physically test out every possible scenario that a vehicle may encounter in order to determine its behaviour, it may be possible to constrain the scenarios to a representative number of sets, that include static/dynamic obstacles, wind and current disturbances, weather conditions, and actuator/sensors states and to implement these in a realistic simulator. Monte Carlo trials with random combinations can then be used to test out a large range of possible variations of these scenarios. Although such testing may work well at the algorithm level it has limitations in being able to represent sensor behaviour. For instance, lidar and radar simulations are based on probabilistic models of sensor physics and noise which are sensitive to small changes in environmental conditions. Likewise, vision systems have trouble disambiguating colour variations due to shadows and glare, and experience difficulties with water reflections. These factors restrict the scope of this form of testing and validation.

It may perhaps be suitable to turn to some form of unsupervised machine learning where the autonomous system is able to learn from previous mistakes and tune its parameters to achieve better performance on its next mission. The testing could be performed using large collections of video, lidar and radar data to train the system so that it may learn to recognise when one or more of its sensor modalities is not reliable and to instead operate in a failover mode forming decisions based solely on those modalities that it believes are correct.

The combination of the above approaches may help to improve confidence in the autonomous vehicle's ability to undertake its mission safely and reliably.