OCEANIDS: Building Next Generation Maritime Autonomous Systems

M. Furlong, R. Marlow, S. McPhail, A. Munafo*, M. Pebody, A. Phillips, D. Roper, G. Salavasidis

National Oceanography Centre, Southampton, UK
* Corresponding Author. Email: andrea.munafo@noc.ac.uk

Synopsis

Oceanids is a 4-year programme for the technological development of innovative Maritime Autonomous System (MAS) platforms and associated sensors that will include next generation robotic AUVs, sensors and networks to undertake ambitious, long-range, long-endurance deployments in extreme and hazardous environments, such as the deep ocean or under-ice environments. This paper describes the design of two new types of robotic AUVs, the Autosub 2000 Under-Ice and the Autosub Long Range 1500 vehicles that are being developed under the programme. Two key components of the AUVs are described, the autonomy framework and the navigational system, which relies on a newly developed terrain-aided navigation (TAN) system. At-sea results of the TAN are also reported as obtained during long duration operational deployments done in 2017.

Keywords: AUV; Autonomy; Behaviour Trees; Terrain-Aided Navigation; Marine systems

1. Introduction: The OCEANIDS Programme

Oceanids is a 4-year programme for the technological development of innovative Maritime Autonomous System (MAS) platforms and associated sensors that will include next generation robotic Autonomous Underwater Vehicles (AUVs), sensors and networks. Among its objectives is the design, build and commission of two new classes of AUVs which together will provide a transformative capability to undertake ambitious, long-range, long-endurance deployments in extreme and hazardous environments, focused on under ice exploration.
This paper describes results from the first year of the programme, providing details on Autosub 2000 Under Ice (Autosub2KUI), a ship launched, high power, AUV platform for under ice exploration, and on Autosub Long Range 1500 (ALR1500), a long-range, long-endurance vehicle.
Moreover, together with a description of the mechanical and electronic design of the vehicles, this paper provides details on the following aspects, critical for the usage of the systems in challenging environments:

- Vehicle autonomy system. In fully known environments with available models, the planning can be done offline. Solutions can be found and evaluated prior to execution. Unfortunately, in most cases, the environment is unknown and the strategy needs to be revised online. For this reason, the vehicles base their autonomy on the newly developed behaviour-tree architecture which is able to adaptively respond to changing conditions and mission requests.
- Vehicle navigation system. To enable long-range and long-endurance missions, a terrain aided navigation (TAN) algorithm has been developed. TAN is an alternative to acoustic-based techniques for bounding the inertial navigation error growth over time. It falls in the class of completely on-board navigation methods, no need for external infrastructure, enabling true autonomy. These properties make it particularly appealing for long range AUV missions. The accuracy of this method depends on the dead reckoning accuracy, the quality of available maps as well as the morphology of the seabed and the sensitivity of sensors (e.g. ADCP) to changes in the AUV pose. Preliminary results based on long duration deployments done during the DynOPO sea trial show the method potential.

The rest of the paper is organised as follows: Section 2 describes the two vehicles. Section 3 goes into the details of the vehicle autonomy system, while Section 4 describes the terrain-aided navigation system. Finally, conclusions are reported in Section 5.

2. The Vehicles

NOC operate the National Marine Equipment Pool to support the UK Oceanographic Community. The vehicles described below will join the existing fleet of underwater gliders, AUVs, ROVs and surface vessels once fully commissioned.

 http://doi.org/10.24868/issn.2631-8741.2018.003

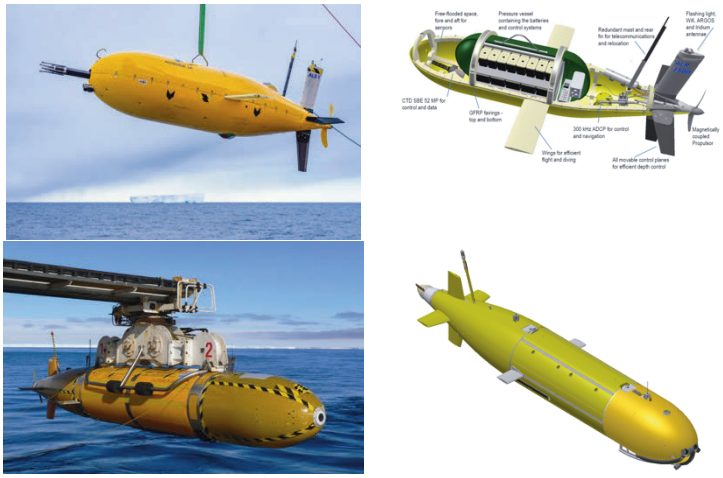

Figure 1: Autosub Long Range 6000 (top left), Autosub Long Range 1500 (top right), Autosub3 (bottom left), Autosub2KUI (bottom right)

2.1. Autosub Long Range 1500

ALR1500 (Roper et al., 2017) is an ultra-long endurance variant of Autosub Long Range (Furlong et al., 2012), also known as BoatyMcBoatface. To enable cross ocean basin operation, the depth rating of the vehicle has been compromised to 1500m to enable sufficient on board energy to enable multi-month operation and up to 6000km range with a low power sensor suite.

2.2. Autosub 2000 Under-Ice

Autosub2KUI will replace the high power under ice survey capability of Autosub3 (McPhail et al., 2009). Providing a complementary capability to ALR1500, Autosub2KUI is focussed on short duration missions with high power acoustic sensors. Table 1 below provides a comparison between the two platforms.

	ALR1500	**Autosub2KUI**
Length	3.5m	5.5m
Mass (dry)	800kg	1800KG
Range	6000km	250km
Speed Range	0.5 to 0.8 m/s	0.8 to 1.6m/s
Endurance	Up to 6 months	Up to 72 hours
Navigational Accuracy (with bottom lock)	<1% distance travelled	<0.1% distance travelled
Typical Payload	Upwards and downwards ADCP, turbulence probe, CTD	CTD, upwards and downwards ADCP, side scan sonar, sub-bottom profiler, multibeam ecosounder

Table 1: Indicative main particulars of the two platforms

3. The Vehicle Autonomy System

3.1. System Design

One of the key components of the Oceanids programme is represented by the development of a new On-Board Control System (OCS) for the Autosub platforms. The objective is to create a unified autonomy and control architecture for the NOC fleet to improve the usage of resources, equipment and people, allowing for easier and fast development, including integration of new payloads and higher degree of autonomy/situational awareness, and to reduce costs and time for vehicle preparation.

In this regards, the OCS is an infrastructure to support the engineering of advanced control and navigation systems. As common in state-of-the-art robotic software development (Ferrati et al., 2016), the objectives for the development of the vehicle OCS are:

http://doi.org/10.24868/issn.2631-8741.2018.003

- avoid code duplicates and enhance code reuse;
- provide a suite of API to developers, including C++ classes and utilities, to ease and speed up production of significant code by hiding code complexity;
- leverage on simulators for fast testing and debugging.

In this way it possible to focus on low level interfaces, middleware management, and network and performance optimization.
The main software architecture is based on a layered component-based architecture (Brugali & Shakhimardanov, 2010) that focuses on modularity to support code reuse and rapid development.
The OCS is based on a distributed network of applications or nodes built in such a way that each different process is an independent element of the infrastructure (Quigley et al., 2009). In this way developers can have more freedom when developing their own modules. Moreover, when possible, the OCS relies on existing components (e.g. algorithms, drivers) available within existing code (see, for example, (McPhail et al., 2009), (Pebody, 2008)).
From an implementation standpoint, the OCS has been divided in two main layers:

- the ROS middleware has been chosen to provide the main inter-process communication links. ROS allows to seamlessly interface to simulators or to the real robot (either remotely through the network or on the same machine using inter process communication).
- the NOCS layer (NOCS On-Board Control System Layer) has been developed to provide an abstraction layer for the robot hardware and to provide a unique set of interfaces. This is implemented through a set of classes and libraries for navigation and control purposes and for the lower level operations (e.g. driver interface). The NOCS layer is responsible for the algorithmic part of the software and interfaces to the ROS layer to communicate with the rest of the system (see Figure 1).

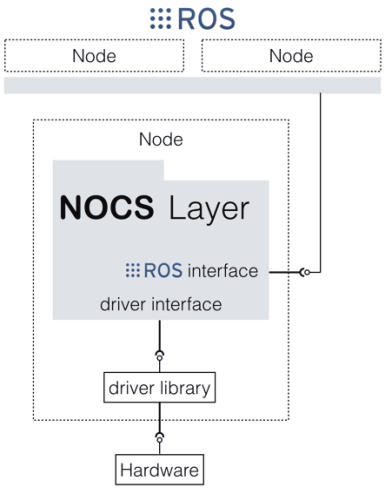

Figure 1. Layered structure of the OCS. The system relies on a set of standard interfaces and API to hide code complexity and increase reuse.

To further decouple algorithmic complexity, the OCS follows the Front-seat/Back-seat design paradigm (Eickstedt & Sideleau, 2010). This has two major components: a higher level, behaviour-based autonomy system, and an interface to a classical dynamic controller. The latter is responsible for real-time control of the vehicle given the decisions of the autonomy system (e.g. vehicle heading, speed, and depth). The front-seat/back-seat driver architecture increases software portability since the autonomy system is completely decoupled from the lower level details of the vehicle.

3.2. System Architecture

The OCS system is based on a hybrid-hierarchical model as shown in Figure 2 and it adopts a three-level control hierarchy (Teck & Chitre, 2012):

- Supervisory level: in charge of making high-level mission decisions, monitoring the vehicle status and communicating with the pilot. The control at this level is deliberative, i.e. each module produces an output based on its internal states and inputs coming from its sensors.
- Mission level: translate higher level tasks (mission goals) and demands into commands (e.g. rudder angles, propeller power, etc.) for the Vehicle level.
- Vehicle level: responsible for performing low-level vehicle control. It interacts reactively with the vehicle sensors and actuators. At the lowest level each actuator of the AUV is controlled by a PID position loop

http://doi.org/10.24868/issn.2631-8741.2018.003

in a distributed embedded electronic system with one board per actuator. In this way the vehicle central pc (MVC) does not have to provide the continuous control of the actuators that instead is delegated to actuator specific PIC controllers. The communication from the MVC to each board is performed using the RS485 communication protocol. The communication with the sensors is based on serial communication, either TTL or RS232.

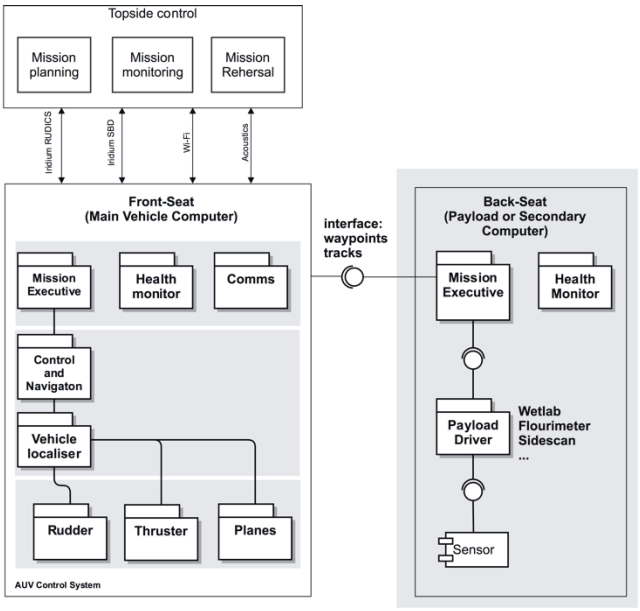

Figure 2. System architecture modularized according to the front-seat (left) and back-seat (right). The front-seat part is further organized according to a three-level control hierarchy: supervisory level (top), mission level (middle) and vehicle level (bottom). The picture also shows the connection with a remote top-side computer.

This system design offers many benefits:

- It decouples the deliberative part of the system from the low level reactive vehicle control, and hence allows different level of real-time requirements.
- Each module inside each level has its private data and implements its own algorithms depending on the assigned responsibilities and goals.
- All the modules are self-contained and have a uniform software interface to facilitate inter-module communication via a message-passing mechanism (i.e. ROS).
- Provides flexibility in terms of software implementation: rather than modify existing software components, new modules can be built and loaded when necessary.

The responsibilities of the OCS modules within the architecture shown in Figure 2 are briefly described below:

Mission Executive: starts, coordinates, oversees and controls the execution of missions. Listens to the safety notifications from the System Health Monitor and aborts the mission if any abnormality is observed.
The mission executive receives mission tasks in the form of mission points (waypoints), and translates the waypoints into lower level vehicle control (bearing, speed, depth and altitude control set-points). This is done according to the mode of waypoint execution (Waypoint following, Path following) (McPhail, Furlong, & Pebody, 2010), (McPhail, 1993), (Ferri, Munafo, & LePage, 2018). The mission executive plans a collision-free path and it re-plans the waypoints if obstacles are detected in the path (Pebody, 2008).
The mission executive is implemented using Behavior Trees and it will be further described in Section 3.3.

Communication: is the AUV's external communication node. Updates the Topside PC and the vehicle pilot with the latest mission and AUV status periodically. Decodes the mission commands received and passes the information to the relevant nodes. It encodes data to be transmitted from the vehicle. The communication module is also responsible for handling underwater communications as well as acoustic localization services (Munafo & Ferri, 2017).

http://doi.org/10.24868/issn.2631-8741.2018.003

System Health Monitor: detects any abnormality reported by local node health monitors (one per node), and monitors vehicle navigational status (maximum pitch, roll, depth and minimum altitude) and system resources. Ensures that the vehicle navigates only within the geo-fenced area defined by the Topside (i.e. User/Pilot).

Local Node Health Monitor: All the sensor and actuator drivers in the vehicle implement a health reporting mechanism. This level collects the information and analyzes the severity when the device is found unhealthy. It notifies the System Health Monitor if the severity is high.

3.3. Behaviour trees

The mission executive layer of the OCS architecture is implemented using Behaviour Trees (BTs), which are a way to structure the switching between different tasks (Colledanchise & Ogren, 2018). One of the main advantages of BTs with respect to the more traditional Finite State Machines (FSM) is represented by their modularity.
Formally speaking, a BT is a directed tree where the internal nodes are called *control flow nodes* and leaf nodes are called *execution nodes*. A BT starts its execution from the root node, which has the special role of allowing the execution of every other nodes in the tree with a specific frequency. To do so, the root generates *ticks signals* that are then sent recursively to its children. Only ticked nodes are executed, and then they immediately return *Running* to the parent, if it is still executing, *Success* if the goal has been achieved or *Failure* otherwise. Control flow nodes are divided in Sequence, Fallback, Parallel, and Decorator nodes. Execution nodes into Actions and Conditions. Without entering into details, a Sequence node executes its children according to a pre-defined priority (e.g. left to right) until it finds a child that returns Success or Failure. At that point it stops and returns the same value to its parent. The Fallback node executes its children until it finds one that returns Running or Success, and then returns this value to its parent. Note that this means that it only returns Failure when all its children fail. The parallel node runs its children in parallel and returning to its parent the value obtained according to some majority threshold. A decorator is used to manipulate the output of its only child (e.g. invert). An Action node executes a command and returns *Success* if the action is correctly completed or *Failure* otherwise. A Condition node is used to check the logical value of a proposition (a Condition node never returns *Running*).

More information on behaviour trees can be found, among others, in (Colledanchise, Parasuraman, & Ogren, 2015), (Colledanchise & Ogren, 2018), (Colledanchise & Ögren, 2017).

Figure 3 shows a (simplified) BT used for the ALR1500 vehicle. The mission is realised as a sequence of a diving phase (where the vehicle moves away from the support ship and start its descent to the initial target depth), followed by a surveying step. The switching from one phase to the next is obtained using a condition node (not shown in the picture) that monitors the reception of a ‘Go command’ from the command and control. The Surveying phase is done running two behaviours in parallel. An obstacle avoidance behaviour (the full sub-tree is not shown in the picture for simplicity), and a track at depth behaviour, which represents the main task of the mission.

3.4. Safety procedures

Given the challenging environments and operating conditions faced by the vehicles, for example when operating under-ice, a set of safety procedures is in place to reduce operational risks. Autosub2KUI is designed with a dual redundancy system for all critical components such as the inertial navigation system, and the primary (front-seat) and secondary (back-seat) computers. Energy constraints do not allow the same approach on ALR1500, which instead has to rely mainly on software safety mechanisms. The system health monitor continuously monitors the main vehicle variables and informs the mission executive when some variables go beyond pre-defined alarm/warning thresholds. Based on this information, the mission executive might re-plan (or in the worst case, abort) the mission to ensure vehicle safety.

http://doi.org/10.24868/issn.2631-8741.2018.003

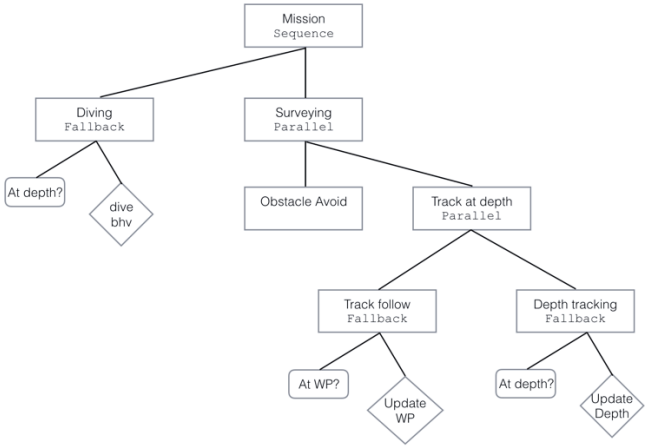

Figure 3. BT (simplified) used to define the mission executive of ALR1500. Condition nodes are represented as rounded rectangles, actions as diamonds.

4. Navigation System

Typical AUV navigation system relies on Dead-Reckoning (DR) techniques for maintaining position estimates (McPhail, 1993). Although DR can provide acceptable accuracy in short oceanographic missions, position estimates accumulate significant error due to inaccurate motion sensors. Hence, the localisation problem for AUVs becomes increasingly challenging as vehicle endurance increases, operating for months rather than hours or days. Terrain Aided Navigation (TAN) methods are localisation methods which use bathymetric measurements for bounding the growth in the dead-reckoning navigation error, given a-priori bathymetric reference maps (Melo & Matos, 2017). The performance of the TAN method is influenced by the quality of the available maps and by the morphology of the seabed (rougher terrains are more informative) as well as by the sensitivity of the bathymetric sensors to changes in the AUV pose (Meduna, 2011). The advantage of TAN with respect to more traditionally beacon-based acoustic methods (Hunt et al., 1974), (Munafo & Ferri, 2017), is that it does not need any external infrastructure, and it is hence particularly appealing for long-range AUV missions. The TAN algorithm developed for the ALR family of vehicles uses a Particle Filter (PF) to estimate the vehicle position. This is a light and tractable filter since it estimates only the vehicle 2D position and, optionally, water-currents (when the AUV is out of bottom-tracking range).

The typical sensor configuration for ALR consists of a GPS, magnetic compass/attitude module, Conductivity, Temperature and Depth (CTD) probe and a downward looking ADCP. In addition, to acquire altitude measurements in deep water missions when the ADCP cannot bottom lock, the AUV is equipped with a low-power and low-frequency single-beam echo-sounder (E/S). The range from the bottom calculated by the ADCP and by the E/S is used by the TAN filter together with the available reference map to correct the vehicle estimated position. Figure 2 shows an overview of the TAN system as implemented on board the vehicles. More details on the particle filters are reported in (Salavasidis et al., 2018).

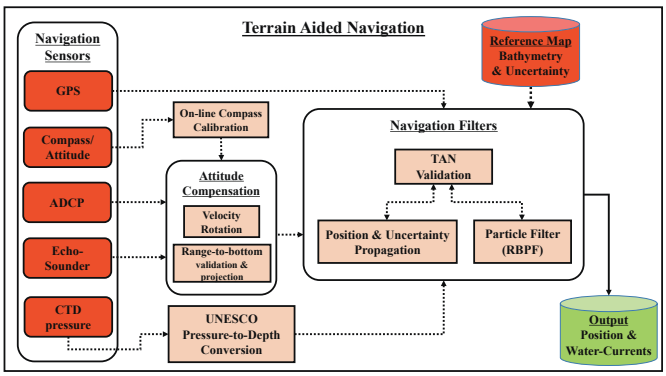

Figure 4. Terrain-aided navigation block diagram.

http://doi.org/10.24868/issn.2631-8741.2018.003

4.1. Sea trials

The performance of the TAN algorithm has been evaluated using field-data collected by ALR6000 (Furlong et al., 2012) during science expeditions in the Southern Ocean as part of the Dynamics of the Orkney Passage Outflow (DynOPO) research program at Orkney Passage, a submarine valley of the Southern Ocean that connects the Weddell Sea and the Atlantic Ocean, held on April 2017 (Naveira Garabatoet al, 2017). This is an area with approximately 5300m maximum water-depth, steep terrain morphology and subject to strong water currents.
ALR performed three deep long-range missions (M41, M42 and M44) diving to maximum depth of 3700m and operating in proximity to the ocean bottom for more than 3 days (195km). In terms of localisation, the ALR navigated using GPS fixes (when on the surface) and dead-reckoning with ground velocity when submerged. No TAN system was used in real-time during the mission since the vehicle was always at bottom tracking distance. The trial, however, represented a very good opportunity to collect relevant datasets to evaluate the TAN performance off-line.
To numerically assess the TAN performance, a set of Monte Carlo (MC) runs has been executed and the TAN results have been compared to Ultra-Short BaseLine (USBL) measurements collected from the RSS James Clark Ross Ship and to the results obtained from the on-board real-time navigation system (Salavasidis et al., n.d.).
Since ALR operated within bottom-tracking range, altitude measurements from the ADCP (averaging the available beams) were used as bathymetric data.
Figure 5 shows the ALR path for the longest mission performance (M44). Note that although mission requirements required bathymetric contour tracking (low-informative along-track terrain morphology) the TAN system was not only able to robustly maintain convergence but to substantially decrease the estimation error compared to the DR estimates. This is visible in Figure 5 and reported in Table 1 for the three missions performed.

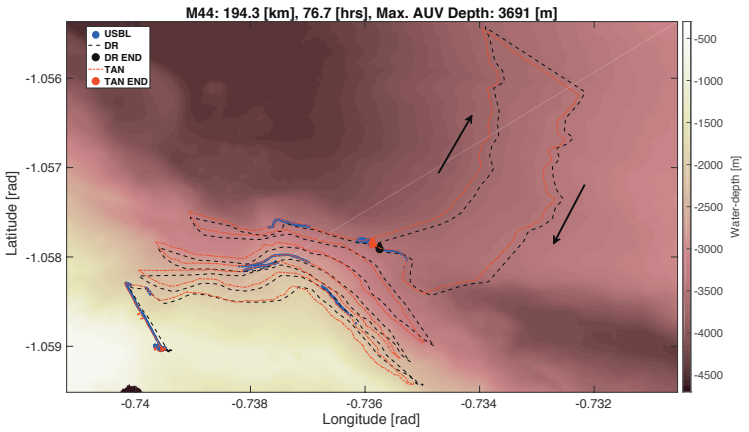

Figure 5. Position estimates for mission M44. The black dashed line shows the path estimated by the real-time on-board navigation system. The TAN algorithm is shown in red and the USBL measurements in blue. Note the increase in agreement between the TAN output and the USBL measurements.

Error	Mission 41/42/44		Units
	TAN	DR	
RMSE (average)	158/105/152	1168/355/400	m
RMSE*	51/59/126	1088/309/398	m
Distance travelled*	0.09/0.05/0.06	2/0.23/0.21	%

*Error corresponding to the latest USBL measurement for each mission.

5. Conclusions

This paper described two new and highly capable AUVs that are currently being developed under the OCENIDS Programme at NOC, Autosub 2000 Under-Ice AUV and Autosub Long Range 1500. The objective for these vehicles is that of undertaking ambitious, long-range, long-endurance deployments in extreme and hazardous

http://doi.org/10.24868/issn.2631-8741.2018.003

environments, such as the deep ocean or under-ice environments. A general description of the two vehicles was reported together with details on their target operational envelope.
Details on the vehicles on-board control and navigation systems were also reported. These are critical components to enable the high-risk high-payoff missions envisaged for these AUVs in the future.
At sea results have been reported for the TAN system obtained using data collected during the DynOPO sea trial held in the Southern Ocean in 2017. Results show that the TAN has the potential to prolong underwater missions to a range of hundreds of kilometres without the need for intermittent surfacing to obtain GPS fixes.

6. Acknowledgements

This work has been done under the NERC OCEANIDS Programme.

7. References

Brugali, D., & Shakhimardanov, A. (2010). Component-Based Robotic Engineering (Part II). *IEEE Robotics & Automation Magazine*, *17*(1), 100–112.

Colledanchise, M., & Ogren, P. (2018). Behavior Trees in Robotics and AI (pp. 1–198).

Colledanchise, M., & Ögren, P. (2017). How Behavior Trees Modularize Hybrid Control Systems and Generalize Sequential Behavior Compositions, the Subsumption Architecture, and Decision Trees. *IEEE Transactions on Robotics*, *33*(2), 372–389.

Colledanchise, M., Parasuraman, R., & Ogren, P. (2015, April 23). Learning of Behaviour Trees for Autonomous Agents.

Eickstedt, D., & Sideleau, S. (2010). The Backseat Control Architecture for Autonomous Robotic Vehicles: A Case Study with the Iver2 AUV, 1–9.

Ferrati, M., Settimi, A., Muratore, L., Cardellino, A., Rocchi, A., Mingo Hoffman, E., et al. (2016). The Walk-Man Robot Software Architecture. *Frontiers in Robotics and AI*, *3*, 664–16.

Ferri, G., Munafo, A., & LePage, K. D. (2018). An Autonomous Underwater Vehicle Data-Driven Control Strategy for Target Tracking. *IEEE Journal of Oceanic Engineering*, 1–21.

Furlong, M., Paxton, D., Stevenson, P., Pebody, M., McPhail, S. D., & Perrett, J. R. (2012). Autosub Long Range: A Long Range Deep Diving AUV for Ocean Monitoring (pp. 1–7). Presented at the 2012 IEEE/OES Autonomous Underwater Vehicles (AUV).

Hunt, M., Marquet, W. M., Moller, D. A., Peal, K. R., Smith, W. K., & Spindel, R. C. (1974). *An acoustic navigation system* (pp. 1–76).

McPhail, S. D. (1993). Development of a simple navigation system for the Autosub Autonomous Underwater Vehicle (pp. 1–6). Presented at the OCEANS'93.

McPhail, S. D., Furlong, M. E., Pebody, M., Perrett, J. R., Stevenson, P., Webb, A., & White, D. (2009). Exploring beneath the PIG Ice Shelf with the Autosub3 AUV (pp. 1–8). Presented at the OCEANS 2009-EUROPE (OCEANS09).

McPhail, S., Furlong, M., & Pebody, M. (2010). Low-Altitude Terrain following and Collision Avoidance in a Flight-Class Autonomous Underwater Vehicle. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, *224*(4), 279–292.

Meduna, D. K. (2011, August 26). *Terrain relative navigation for sensor-limited systems with application to underwater vehicles*.

Melo, J., & Matos, A. (2017). Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean Engineering*, *139*, 250–264.

Munafo, A., & Ferri, G. (2017). An Acoustic Network Navigation System. *Journal of Field Robotics*, *34*(7), 1332–1351.

Naveira Garabato, A. (2017). RRS James Cook Cruise JR16005, 17 Mar - 08 May 2017. The Dynamics of the Orkney Passage Outflow (DynOPO).

Pebody, M. (2008). Autonomous underwater vehicle collision avoidance for under-ice exploration. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, *222*(2), 53–66.

 http://doi.org/10.24868/issn.2631-8741.2018.003

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., et al. (2009). ROS: an open-source Robot Operating System (pp. 1–6). Presented at the IEEE International Conference on Robotics and Automation (ICRA).

Roper, D., Phillips, A. B., Harris, C., Salavasidis, G., Pebody, M., Templeton, R., et al. (2017). Autosub Long Range 1500: an ultra-endurance AUV with 6000 Km range. Presented at the OCEANS 2017 - Aberdeen.

Salavasidis, G., Munafo, A., Harris, C., Phillips, A. B., McPhail, S., Pebody, M., Eric Rogers (2018). Terrain-Aided Navigation for Long-Endurance and Deep- Rated Autonomous Underwater Vehicles. *Journal of Field Robotics*, *In press*.

Teck, T. Y., & Chitre, M. (2012). Hierarchical Multi-Agent Command and Control System for Autonomous Underwater Vehicles.

 http://doi.org/10.24868/issn.2631-8741.2018.003