

Vertex: A New Distributed Underwater Robotic Platform for Environmental Monitoring

Felix Schill and Alexander Bahr and Alcherio Martinoli

Abstract We present a new Autonomous Underwater Vehicle (AUV) system for cooperative environmental sensing. The AUV was specifically developed as a platform for distributed, cooperative sensing in lakes and coastal areas. In this paper we describe the prerequisite subsystems for a submersible multi-robot system and their interactions. In particular, we incorporate a distributed acoustic localisation system and distributed time-sliced communication systems into an agile, 5-DOF submersible robot that is small, easy to deploy and retrieve, with a modular environmental sensor payload for relevant scientific measurements. We also developed a distributed Hardware-In-the-Loop (HIL) simulation framework to facilitate early testing of algorithms in simulation while running final binary code on the actual robot hardware. To avoid communication overhead and real-time issues, the simulation of the vehicle dynamics and all proprioceptive sensors is performed on-board. Exteroceptive sensors are simulated by vehicle-to-vehicle communication where possible, supported by a central simulation supervisor where required. Finally, we present some preliminary experimental results of the system.

1 Introduction

Obtaining underwater measurements of biological, chemical and physical parameters is currently an expensive and time consuming activity. In environmental science such measurements are generally obtained manually by lowering a probe from a ship, or automatically with fixed moorings. This limits the amount of spatial data

Felix Schill and Alcherio Martinoli

École Polytechnique Fédérale de Lausanne (EPFL), Distributed Intelligent Systems and Algorithms Laboratory (DISAL), School of Architecture, Civil and Environmental Engineering, Lausanne Switzerland, e-mail: felix.schill@epfl.ch, alcherio.martinoli@epfl.ch

Alexander Bahr

Hydromea SA, Lausanne, Switzerland, e-mail: alexander.bahr@hydromea.com



Fig. 1 Left to right: The Vertex AUV with carry case, in the water at Lake Geneva, and on the ice at Lake Onego, Russia during a field campaign prior to deployment.

that is available to very few spots (often only the deepest point of a lake). Remote sensing techniques that are commonly used on land such as hyperspectral imaging and radar measurements do not work under water except for the surface layer, as water absorbs most of the electromagnetic spectrum with a very high attenuation. Sonar measurements can only resolve some physical features but are unsuitable for biological and chemical measurements. However, through continuous development and miniaturisation efforts there is a large range of small, precise in-situ sensors available for physical (e.g. conductivity, temperature, turbidity), chemical (e.g. pH, nitrate, chlorine, hydrocarbons) and biological parameters (e.g. fluorimeters for algae, chlorophyll, dissolved organic matter). Autonomous Underwater Vehicles (AUVs) have been used for many years to carry such sensors and extend the reach of these instruments. Due to their size and very high cost, AUVs are mostly used in oceanography, but very rarely in limnology and environmental monitoring applications. Additionally, most operators can afford at most one AUV, which limits the volume of water that can be explored within a given time frame. Particularly when studying spatio-temporal phenomena such as wastewater plumes, mixing, convection and biologically active layers, it is insufficient to measure only in one spot, or with only one AUV, as the phenomenon can change drastically over time and space. A much better picture can be obtained by measuring with many sensors at many places simultaneously. Our goal is to develop a fleet of 5-10 small AUVs that can cooperatively obtain high resolution water quality measurements of natural water bodies. For practical reasons, the AUVs should be small and portable for easy deployment and retrieval, and should be designed so that they can be produced cost-effectively in large quantities. For spatially coordinated underwater operations, they have to be able to sense the relative location of nearby vehicles, and be able to communicate to their peers. Lastly, a high number of degrees of freedom is helpful as it simplifies the control of the robots.

1.1 Related Work in Distributed AUV Systems

There have been multiple efforts in the past to create an underwater swarm. In 2003 two of the authors were involved in the Serafina project by Uwe R. Zimmer at

the Australian National University. An initial prototype was designed by Alexander Bahr and further built and developed by Felix Schill. The main focus of the project was to develop the prerequisites for underwater swarms, in particular scalable communication [13], and cooperative relative positioning [8][2]. The solutions to these prerequisites were individually evaluated, but the project concluded before a large-scale swarm could be built. The CoCoRo project succeeded in building a large underwater swarm [16], but was clearly focused on indoor in-vitro experimentation [10]. The AUVs in that project have a very limited endurance and depth capability, are not suitable for outdoor use and were not designed to carry a useful payload. The optical communication system is scalable to large numbers, but has very limited range which can not easily be extended as it is dependent on water turbidity. There have been experiments with multiple underwater vehicles where communication was limited to a surface uplink [6], and an experiment with one AUV communicating underwater with a multiple stationary sensor nodes [4]. Successful formation control with three AUVs has been demonstrated recently using acoustic modems by [18], where two vehicles were at the surface using GNSS, and one vehicle followed based on acoustic range measurements. A number of AUV designs appeared in recent years with significantly reduced cost and outdoor-capable performance, such as the OceanServer Iver3, LAUV [17], Starbug [5] and COTSbot [3]; however to our knowledge they were not designed to operate in large cooperative teams, and are significantly heavier and bigger than our design presented in this paper.

1.2 Related Work in Localisation and Communication

In any cooperative multi-robot system, robots need to have some awareness of nearby other robots. In most practical systems, this means an ability to measure the relative position of other robots in proximity. Additionally, many distributed control strategies assume some means of communication with other robots. As both localisation and communication are part of control loops, they should be real-time and scalable. Local coverage of nearby robots is more important than global coverage, and often sufficient, but some distributed control strategies do require global information exchange to reach consensus. The underwater environment poses unique challenges in both domains, mainly due to the very high attenuation of electromagnetic waves. With the exception of very low radio frequencies, and the green/blue part of the visible spectrum of light, the attenuation is prohibitive even for short distances. This excludes most technologies commonly used in air, such as Global Navigation Satellite Systems (GNSS), ZigBee, WiFi or other common radio systems. Poor visibility also often precludes the use of optical channels or cameras. The most common solution is acoustic transmission for communication, localisation and remote sensing (sonar). Unfortunately, acoustic communication does not scale very well to large networks due to the high distortion and interference, as sound travels long distances but gets distorted beyond decodability [7]. Common solutions for underwater localisation are either based on external reference beacons or

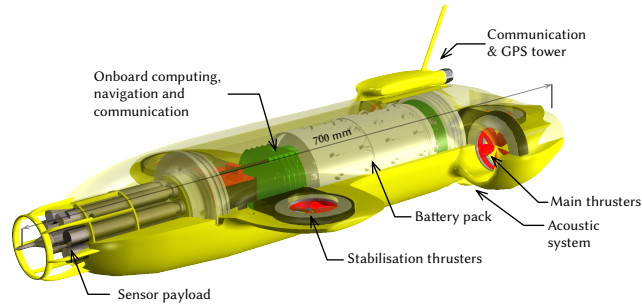


Fig. 2 CAD rendering of the Vertex AUV

trackers that require fixed infrastructure (e.g. LBL, USBL), or inertial systems supported by Doppler velocity measurements taken against the sea floor (DVL) which can drift over time. For relative position sensing, such systems would still require communication and a common coordinate frame. To address these issues, we propose a combination of acoustic relative positioning [2], VLF radio communication [13] and short-range optical communication [15], GNSS (surface only) and inertial navigation (see below in Sections 3 and 4).

2 AUV System Design

Our design goals were to develop an AUV that is small (portable in one hand), rated for outdoor operation in most inland and coastal waters, capable of carrying a useful water quality sensing payload, and designed for multi-robot operation in large groups. The final design is 70 cm long, weighs 7 kg, and is equipped with a YSI EXO2 sensor system which can be configured with up to 7 different physical, chemical and biological sensors. The maximum design depth is 300 meters, which covers the majority of inland waters and coastal areas. For heavier payloads a lighter pressure hull with reduced depth rating can be used. Two battery packs provide a total of 170 Wh of energy, resulting in an estimated endurance of 6-8 hours at a cruise speed of up to 1 m/s. The AUV can also operate on a single battery pack to increase payload capacity if required. The sensor payload and batteries together make up approximately 60% of the overall length of the vehicle. With the given payload we believe that the current design is close to the minimum size for such a vehicle; a further reduction in size would lead to significant performance losses.

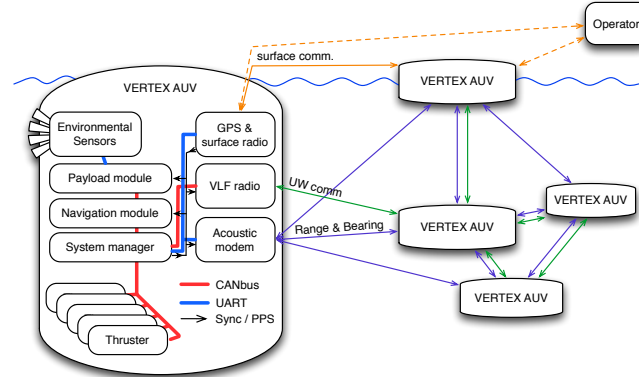


Fig. 3 System diagram of the AUV subsystems

2.1 Propulsion

The thrusters are a newly developed brushless rim-drive design without a central hub in the propeller. The absence of a central shaft greatly reduces the risk of entanglement with floating debris. Each thruster has an integrated motor controller, which implements 3-phase commutation and closed-loop speed control. The motor control system was optimised for fast reversal and quick response, which is important to achieve good attitude control. Five low-profile thrusters provide propulsion and attitude control to the AUV. The two lateral horizontal thrusters provide forward thrust and yaw control. Two lateral vertical thrusters mounted in the winglets in front of the center of gravity provide roll control, and contribute to pitch control and depth control together with a vertical thruster mounted centrally in the tail. The thrusters have a thickness of only 15 mm, which makes it possible to integrate them into hydrodynamically efficient winglets to reduce drag.

2.2 Hardware Architecture

The onboard embedded system consists of multiple modules built around 32-bit microcontrollers: a system management module for power management, data logging and supervision, a navigation module for inertial sensing and closed-loop control, a longwave radio module for underwater communication, and a payload module to interface to the payload sensors (Fig. 3). An external GNSS and communication module provides connectivity via a 868 MHz modem and GNSS positioning while at the surface. The modules are connected via a CAN bus, and dedicated UART connections. Each module can be switched on or off as needed to reduce power consumption, and all internal voltages and currents are continuously monitored. The hotel load of the complete system when fully operational without the motors run-

ning is 1.6 W without payload, and 3 W with the EXO2 payload active and sampling 5 sensors at 1 Hz.

2.3 Synchronised Clocks

A common time base is very useful in a distributed system for ordering events, measuring propagation delay, coordinated control, and analysing log files. The Vertex AUV has a precise reference clock that is synchronised at the surface using the GNSS time pulse. The reference clock is temperature compensated and an order of magnitude more precise than regular clock crystals used for microcontrollers (below 1 ppm). Internally, the reference clock system synthesises an electrical time pulse once per second that is distributed to all microcontroller modules, which run on regular crystals to reduce cost. Each module continuously synchronises its own clock to the time signal to within one microsecond. While submerged, without the GNSS reference, the internal clock system keeps synchrony within at most one millisecond per hour - measured drift rates are 0.3 microseconds per second. In most missions the AUVs can be expected to surface at least once per hour, therefore all clocks are synchronised to within one millisecond or less. We are working on a further improvement using long term drift observations against GNSS to auto-calibrate the reference clock.

2.4 Software Framework

A modular software framework has been developed in C which runs bare-bone on the microcontrollers. The overall software architecture was derived from the initial version of the MAV'RIC autopilot framework originally developed by Schill et al. for a swarm of quadrotors, which is now openly available in a similar, updated version [1]. For execution of subtasks at different rates, a simple fixed-priority scheduler calls the various subroutines at the specified frequency. We decided to implement subroutines with short, bounded execution times that operate on the system state, as opposed to full tasks or threads. The benefit is reduced memory usage, and lower complexity regarding task synchronisation and mutual exclusion as there are no context switches within a subroutine. Time-critical tasks are handled through interrupts and hardware events where appropriate, and concurrency is achieved by using a distributed architecture with dedicated microcontrollers. Communication between different modules (processors), as well as the telemetry link to the base station computer, is handled using the MAVlink protocol [9]. A second scheduler instance has been added for managing dynamically adjustable telemetry message update rates that can be adjusted during runtime through the operator interface. MAVlink also provides parameters that can be edited at runtime. A flexible message callback system was implemented in MAV'RIC for responding to incoming

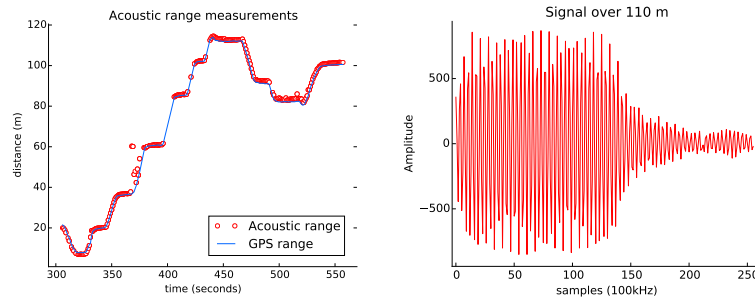


Fig. 4 Left: Acoustic range measurements (red) vs. GNSS (blue). Right: Recording of acoustic pulse at maximum distance of 110 m.

messages as they arrive. The communication channel is abstracted by a bytestream interface, which can be mapped to various interfaces such as a CAN-bus, UART, Radio or UDP. Above the hardware driver level the interface is fully transparent and hardware-independent. All internal and external message traffic is logged to an internal microSD card for post-mission analysis. A simulation layer, described later in Section 5, was added to simulate all relevant sensors on a low level for HIL simulation. The software framework was also ported to Linux by replacing the microcontroller hardware abstraction with a thin emulation layer. This makes it possible to compile and test almost identical code (only with a different hardware abstraction layer) on Linux, with minimal differences to onboard HIL simulation and real experiments. The internal communication streams as well as the telemetry link are transparently mapped to UDP to emulate the internal data bus as well as the telemetry uplink to the basestation software. This enables fast prototyping, debugging and testing of code on a PC of directly compatible code.

3 Inter-Vehicle Acoustic Localisation

For localisation purposes we developed a miniature acoustic modem to fit within the small scale of the Vertex AUV. Instead of relying on external, fixed infrastructure, we propose to use the AUV swarm as the reference for each vehicle, based on Bahr's previous work [2]. Each robot regularly sends a short acoustic pulse at a precisely defined time based on the synchronised reference clock. When nearby robots receive the pulse, they can immediately calculate the time of flight based on the arrival time, and derive the distance to the other robot. Additionally, together with the acoustic pulse, the sending robot also communicates its own position estimate and uncertainty. These distance measurements are then incorporated in the position state estimator, that updates the robot's own position as well as the estimated position of surrounding robots. If any of the robots is at the surface, it will use the GNSS to correct its absolute position, which will in turn improve the position estimates of

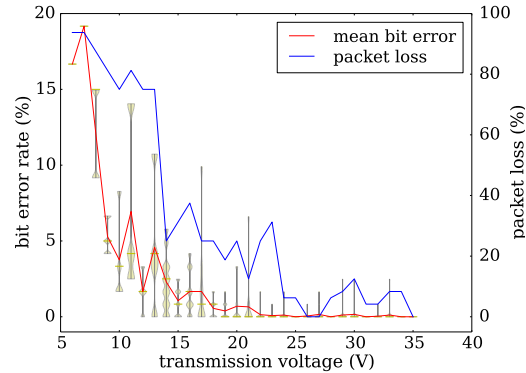


Fig. 5 Bit error rate and packet loss of VLF radio transmission over 20 m in air, for increasing transmitter drive voltage.

the other AUVs due to the lower uncertainty of the surfaced robot. The VLF radio system will be used for communication of the position estimates and uncertainty matrices. We are currently also working on using the acoustic system also for transmission of just the position estimate in a compressed form. While acoustics have known scalability issues, it may be possible to encode a very small amount of data in short pulses that remain separated even with many nodes present. The advantage is that partial functionality of the system can be maintained over much longer ranges, and increased reliability.

4 Inter-Vehicle Communication

The two challenges for an underwater communication network that is scalable to many nodes are how to transmit data through the water, and how to distribute access to the extremely limited communication channel to all nodes fairly while fully utilising the available bandwidth. The acoustic channel is not very well suited for continuous use by many nodes, due to high latency, reverberation and acoustic interference over long distances. Acoustics is therefore only suitable for sparse long-range signaling of only very small amounts of data. In previous work we developed a Very-Low-Frequency (VLF) radio system [12] that can transmit at up to 8192 bps over short to medium distances. An early prototype with 5 V transmitter drive voltage was tested in fresh water and seawater, achieving a range of approximately 4 meters, and 5 m in air. We redesigned the radio for the AUV with increased power, boosting the transmitter voltage to a software-programmable value up to 35 V. Range in air improved to above 20 m (Fig. 5); range in water still has to be tested. The results shown do not use error-correction - we are planning to implement a suitable error-correcting code to reduce packet loss. Additionally, we integrated an

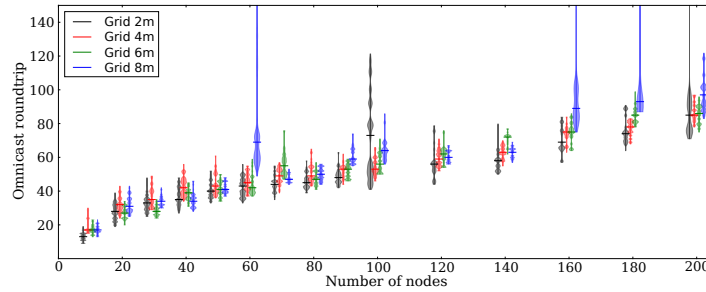


Fig. 6 Real-time simulation of the distributed PDAOS scheduling algorithm (from [14]): Communication time slots required for global information exchange (omnicast roundtrip) for varying network sizes, with nodes arranged in a 2D grid with 2, 4, 6 or 8 m distance (assuming 10 m communication range)

optical communication system for short ranges up to approximately 5 meters, based on [15]. Although optical links are strongly affected by water turbidity and have a very short range, they have a much higher bandwidth, in our case up to 115200 bps. While at the surface, a 868 MHz radio provides a long-range link to the operator and other vehicles that are at the surface.

Although the VLF radio offers bitrates that are higher than most acoustic modems, the available bandwidth is still very limited. To control fair access to the limited channel, we previously developed a very efficient, distributed Time Division Multiple Access (TDMA) algorithm that assigns time-slots to each node such that the channel can be fully utilised while avoiding message collisions. The algorithm is fully distributed without central coordinators; all nodes have the same behaviour. Two variants of this algorithm were previously published: DAOS (*Distributed Ad-hoc Omnicast Scheduler*) generates fully collision-free schedules, but is less efficient for dense networks with high connectivity. The *Pruned Distributed Ad-hoc Omnicast Scheduler* (PDAOS) dynamically prunes schedules based on signal strength or distance, to give priority to nearby nodes. In simulations, PDAOS has shown excellent performance in networks with different density, and fast dynamic changes in network topology. We tested on a benchmark called “omnicast” for global information exchange, which is the time required until all nodes have received information from all other nodes, assuming bounded message size (e.g., how long it would take to reach consensus on a global maximum/minimum). A conservative theoretical upper bound of $2n - 2$ time slots for n nodes has been shown previously [11], which is clearly outperformed by the distributed algorithm, as shown in Fig. 6. In fact, global information exchange can be achieved in significantly less than n time slots for larger networks above 40 nodes. This is possible because nodes that are far apart can communicate in parallel; additionally, even though PDAOS deliberately allows message collisions, some nodes may still receive valid messages and further spread information despite collisions. A detailed analysis of the static and dynamic performance of PDAOS can be found in [14].

5 Simulation Framework

An important feature for testing embedded control software is HIL simulation. Generally this is done by connecting the external interfaces of a control system to a simulation computer, which reads the system outputs, simulates the system dynamics and sends emulated sensor data back to the system under test. On small, integrated embedded systems, however, it is often difficult to gain access to the sensor interfaces, as many sensors are soldered directly onto the circuit board. Also, some of these sensors are sampled at a high rate - in our case, the inertial sensors are sampled at 200 Hz. It would therefore be challenging to send actuator commands to an external computer, and send back the full set of simulated sensor data through a serial link with acceptable latency for a faithful real-time simulation.

In our system, we therefore decided to simulate the vehicle dynamics and proprioceptive sensors directly on the microcontroller itself. The cut is done at the lowest-possible level: the simulator inputs are the actuator control commands as they are sent to the motor controllers. The simulator then calculates estimated rotor speeds, propeller lift and drag, as well as thruster inertia for each thruster. The calculated forces are combined to calculate 3D torque and linear forces on the AUV body, which are then integrated to obtain angular rates, attitude, velocity and position. From these results, sensor values are simulated for the accelerometers gyroscopes, compass, GNSS and depth sensor, using the real, inverse scale factors and offsets. The remaining state estimation and control code is unchanged between simulation and operational mode - the normal attitude estimation filter, attitude/velocity control and navigation code is applied in simulation mode as it is during normal operation. The simulation update takes less than 2 milliseconds, and runs at 200 Hz which is the same rate as the inertial sensor sampling rate during normal operation. The result is a simplified but reasonably faithful simulation of the AUV dynamics once the simulator parameters are calibrated. This allows testing of all high-level code as well as the majority of low-level code on the embedded system itself. As the project can also be compiled and run on Linux, the virtually identical code can also be tested on a PC which is often more convenient for initial development. Once the code has been successfully tested in the Linux emulation, the project is re-compiled and tested on the real hardware in the onboard simulation, which occasionally may reveal hardware-specific issues. It is then possible to switch off simulation mode, even at runtime, which activates the real actuators and sensors, and tests can continue in reality. Monitoring of the simulation occurs through the same telemetry link, using the same MAVlink message protocol, as during live experiments. This facilitates comparing results between simulation and reality, creating consistent log files and data processing pipelines.

Exteroceptive sensors can not be simulated entirely on the vehicle or vehicle processes without external inputs. We chose a distributed approach, using direct robot-to-robot communication to transmit the minimal required information, and carry out the simulation of exteroceptive sensors within the target vehicle software. This approach scales well with the number of robots in the simulation, and in many cases is very close to the real scenario. A simple example is a formation control task of multi-

ple robots. Each robot requires range and bearing information from the other robots. This information would normally be acquired by the acoustic system of the AUV, but for tests of the higher-level formation control algorithms it can also be computed from vehicle positions. In this case, we chose to send global position messages between the simulated vehicles, as a detailed simulation of acoustics was not required. We recently showed the effectiveness of this simulation protocol in the framework of a student project with four Autonomous Surface Vehicles (ASVs) based on the same hardware and software architecture as the Vertex AUV, using a combination of radio communication and GNSS-based localisation for their inter-vehicle positioning. Experiments were carried out in PC-based and HIL simulation, as well as on Lake Geneva. This example illustrates that the flexible simulation framework can be tuned to desired abstraction levels, while carrying out meaningful simulation tests on final binary code on real hardware with multiple robots. Furthermore, to test the acoustic range and bearing measurement algorithms, the simulation was extended to synthesize acoustic samples that emulate real recorded data. In this case, the information sent between robots in simulation is their position as well as the sampled signal they transmit. Instead of sending samples through the DAC, the transmitting side sends them through the MAVlink channel to all other robots at the DAC sampling rate of 1 MHz. As other robots receive the sample messages, they look up the transmitting robot's position based on the last received position message. Based on the samples, message timestamp and position, the robots can calculate the time of flight and attenuation of the signal. The samples are then mixed into a mixing buffer which adds all received signals together. Some IIR bandpass filtering and noise is applied to emulate the frequency response and resonance of the real piezo transducers. When the receiving robot requests samples from the simulated ADC, the mixing buffer is downsampled to the ADC frequency (in this case usually 100 kHz). All remaining code beyond this mechanism is now exactly identical to the code running in reality with the real DAC, transducers and ADC. There are of course deviations between simulation and real data, but the main goal of this framework is to test code in the target language in a scenario that is realistic enough to detect bugs early, and to test algorithms under idealised conditions before subjecting them to the real world. Being able to test code early and often, initially on a PC, then on the target hardware, significantly reduces the overhead of experimental campaigns.

6 AUV Field Tests

The first AUV prototype was completed in October 2014, and tested in an indoor tank. Shortly after we were given the opportunity to participate in the field campaign "Ladoga - Life under ice" in mid-March 2015, as well as in March 2016, organised by the Limnology Center at EPFL. This campaign is a Russian-Swiss multidisciplinary collaboration, to study under-ice dynamics, aquaphysics and biology at Lake Ladoga, the largest lake in Europe, and how it is affected by climate change. Due to the unusually warm winter in 2015, the ice cover at the planned lo-

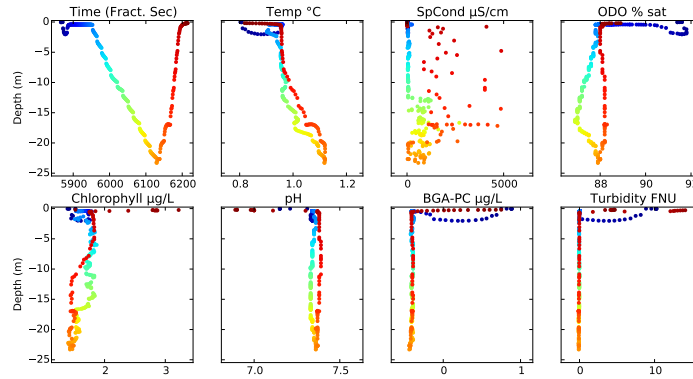


Fig. 7 Data collected during an autonomous dive at Lake Onego, 25.03. 2015. The vertical axis is the depth below the surface. The color coding indicates time, starting at dark blue, through green, yellow, orange to red.

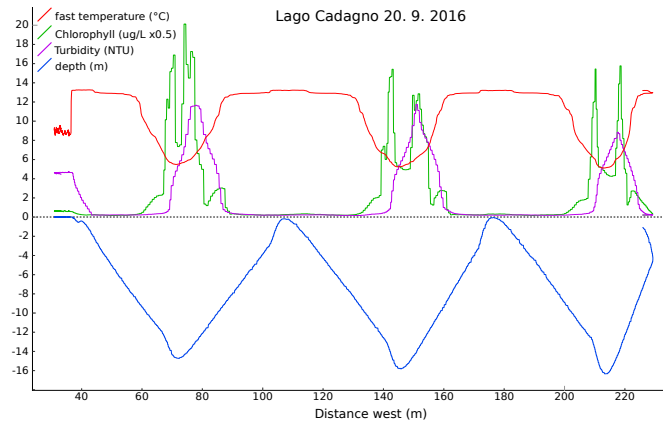


Fig. 8 Autonomous transect (east-west), excerpt from data collected at Lago Cadagno, Sept. 2016. Sensor payload: YSI EXO2: Turbidity, Chlorophyll (shown), CTD, oxygen, BGA-PE, BGA-PC (not shown); additionally a fast temperature sensor (FP07) from Hydromea measuring at 250 Hz. The AUV repeatedly traversed a distinct bacterial layer of interest at 13 m depth, detectable by a drop in Chlorophyll and sharp increase in turbidity, as well as a distinct temperature signature.

cation was already disappearing. The location was therefore changed to the nearby Lake Onego, close to the city of Petrozavodsk. Ice cover at the location was 40 cm, the air temperature varied between -15 and 10 degrees. Despite a few days of unusually high temperatures the ice cover remained safe for the duration of the campaign, but started breaking up the week after. The AUV was deployed through a hole in the ice to collect measurements horizontally in a 25-30 m radius around the hole to assist in a study of convection patterns. The payload consisted of sensors for conductivity, temperature, oxygen, pH, chlorophyll, blue-green algae, turbidity

and fluorescent dissolved organic matter. During the field campaigns, the acoustic localisation system was not fully operational yet, and due to the ice cover it was impossible to surface for a GNSS fix or retrieval. A safety line was therefore attached to avoid loss of the vehicle and to aid with recovery to the ice hole. The deployment provided an opportunity for thorough system testing in harsh real-world conditions. The dry air and cold temperatures caused a significant amount of static electricity that our system was not previously exposed to, and prompted a design improvement of the electronics which has now been incorporated into the AUV. The collected sensor data (Fig. 7) was provided to the limnologists on the campaign to augment their own measurements.

More recently the Vertex AUV was also deployed at Lago Cadagno in Switzerland to investigate the spatial properties of a bacterial layer. During this campaign the AUV carried out multiple autonomous transects following a vertical zig-zag pattern in depth. An excerpt of the collected data is shown in Fig. 8. The sensor payload was similar to that of the Onego campaign, but with an additional 250 Hz fast temperature sensor for microstructure turbulence measurements. The data shown here represent an example of the possible applications of the vehicle; an in-depth discussion of the data will be published separately in collaboration with our limnology collaborators.

7 Conclusions

We presented a new underwater robot system natively designed for distributed, cooperative aquatic sensing and environmental research. The Vertex AUV is a major effort to integrate a number of important technologies, including scalable underwater localisation and communication into one coherent system. A powerful and flexible simulation framework facilitates experimentation and research with new distributed algorithms, which can be evaluated and tested in the lab first before actual real-world experiments. We also presented preliminary experimental results of the acoustic ranging system and the VLF communication system. As a demonstration of the AUV's capabilities we also presented some results from the first field campaigns in a challenging under-ice environment as well as a mountain lake with a scientifically relevant payload. Current and future work will focus on validation of the multi-robot capabilities of the system, underwater formation control, and further field campaigns with multiple AUVs in collaboration with our partners in limnology.

Acknowledgements This work has been financially supported over multiple years by the following sponsors (in chronological order): National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 51NF40-111400, through the Spin Fund project "Serafina - Large Scale Underwater Exploration using Groups of Autonomous Underwater Vehicles"; Swiss Commission for Technology and Innovation under Grant No. 16348.1 PFES-ES; the Technology Transfer Office of EPFL, through the Enabling Grant No. TTO 6.1419; FEEL Foundation

supported by Ferring Pharmaceuticals, under the project “Ladoga - Life under ice”; and the Swiss National Science Foundation under the Sinergia Grant No. CRSII2_160726/1.

References

1. MAVRIC autopilot project. https://github.com/lis-epfl/MAVRIC_Library
2. Bahr, A., Leonard, J.J., Fallon, M.F.: Cooperative localization for autonomous underwater vehicles. *The International Journal of Robotics Research* **28**(6), 714–728 (2009)
3. Dayoub, F., Dunbabin, M., Corke, P.: Robotic detection and tracking of crown-of-thorns starfish. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1921–1928 (2015)
4. Dunbabin, M., Corke, P., Vasilescu, I., Rus, D.: Experiments with cooperative control of underwater robots. *The International Journal of Robotics Research* **28**(6), 815–833 (2009)
5. Dunbabin, M., Roberts, J., Usher, K., Winstanley, G., Corke, P.: A hybrid AUV design for shallow water reef navigation. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2105–2110 (2005)
6. Fiorelli, E., Leonard, N., Bhatta, P., Paley, D., Bachmayer, R., Fratantoni, D.: Multi-AUV control and adaptive sampling in Monterey Bay. *IEEE Journal of Oceanic Engineering* **31**/4, 935–948 (2006)
7. Frater, M.R., Ryan, M.J., Dunbar, R.M.: Electromagnetic communications within swarms of autonomous underwater vehicles. In: *Proceedings of the 1st ACM International Workshop on Underwater Networks*, pp. 64–70. ACM Press, New York, NY, USA (2006)
8. Kottege, N., Zimmer, U.R.: Underwater acoustic localization for small submersibles. *Journal of Field Robotics* **28**(1), 40–69 (2011)
9. Meier, L., Tridgell, A., Goppert, J.: Mavlink micro air vehicle communication protocol. <http://qgroundcontrol.org/mavlink/start>
10. Mintchev, S., Donati, E., Marrazza, S., Stefanini, C.: Mechatronic design of a miniature underwater robot for swarm operations. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2938–2943 (2014)
11. Schill, F., Trumpf, J., Zimmer, U.R.: Towards optimal TDMA scheduling for robotic swarm communication. In: *Proceedings Towards Autonomous Robotic Systems*, pp. 197–203 (2005)
12. Schill, F., Zimmer, U.R.: Effective communication in schools of submersibles. In: *Proceedings IEEE OCEANS 06*, pp. 1–5 (2006)
13. Schill, F., Zimmer, U.R.: Pruning local schedules for efficient swarm communication. In: *Proceedings of the International Symposium on Underwater Technology*, Tokyo, Japan, pp. 594–600 (2007)
14. Schill, F., Zimmer, U.R.: A scalable electro-magnetic communication system for underwater swarms. *9th IFAC Conference on Manoeuvring and Control of Marine Craft* pp. 97–102 (2012)
15. Schill, F., Zimmer, U.R., Trumpf, J.: Visible spectrum optical communication and distance sensing for underwater applications. In: *Proceedings of the Australasian Conference on Robotics and Automation*, pp. 1–6 (2004)
16. Schmickl, T., Thenius, R., Moslinger, C., Timmis, J., Tyrrell, A., Read, M., Hilder, J., Halloy, J., Campo, A., Stefanini, C., Manfredi, L., Orofino, S., Kernbach, S., Dipper, T., Sutantyo, D.: CoCoRo – the self-aware underwater swarm. In: *Proceedings of Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops*, pp. 120–126 (2011)
17. da Silva, J.E., Terra, B., Martins, R., de Sousa, J.B.: Modeling and simulation of the LAUV autonomous underwater vehicle. In: *13th IEEE IFAC International Conference on Methods and Models in Automation and Robotics* (2007)
18. Soares, J., Aguiar, A., Pascoal, A., Martinoli, A.: Joint ASV/AUV range-based formation control: Theory and experimental results. In: *IEEE International Conference on Robotics and Automation*, pp. 5579–5585 (2013)