




Distributed system of autonomous buoys for scalable deployment and monitoring of large waterbodies

Brandon M. Zoss¹ · David Mateo² · Yoke Kong Kuan² · Grgur Tokić¹ · Mohammadreza Chamanbaz² · Louis Goh² · Francesco Valleggra² · Roland Bouffanais²  · Dick K. P. Yue¹

Received: 18 January 2017 / Accepted: 16 January 2018 / Published online: 13 February 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

The design, construction, and testing of a large distributed system of novel, small, low-cost, autonomous surface vehicles in the form of self-propelled buoys capable of operating in open waters is reported. We detail the successful testing of collective behaviors of systems with up to 50 buoys, achieving scalable deployment and dynamic monitoring in unstructured environments. This constitutes the largest distributed multi-robot system of its kind reported to date. We confirm the robustness of the system to the loss of multiple units for different collective behaviors such as flocking, navigation, and area coverage. For dynamic area monitoring, we introduce a new metric to quantify coverage effectiveness. Our system exhibits near optimal scalability for fixed target areas and a high degree of flexibility when the shape of the target changes with time. This system demonstrates the potential of distributed multi-robot systems for the pervasive and persistent monitoring of coastal and inland water environments.

Keywords Multi-robot system · Collective behavior · Autonomous surface vehicle · Dynamic area coverage · Distributed communication

This is one of several papers published in *Autonomous Robots* comprising the “Special Issue on Distributed Robotics: From Fundamentals to Applications”.

✉ Roland Bouffanais
bouffanais@sutd.edu.sg

Brandon M. Zoss
bzoss@mit.edu

David Mateo
david_mateo@sutd.edu.sg

Yoke Kong Kuan
yokekong_kuan@sutd.edu.sg

Grgur Tokić
gtokic@mit.edu

Mohammadreza Chamanbaz
chamanbaz@sutd.edu.sg

Louis Goh
louis_goh@sutd.edu.sg

Francesco Valleggra
francesco_valleggra@sutd.edu.sg

Dick K. P. Yue
yue@mit.edu

1 Introduction

Lakes, reservoirs and coastal marine areas are home to a complex interplay between dynamic, physical, and biogeochemical processes (Beşiktepe et al. 2003). Human intervention and activity further complexifies the dynamics of these precious environments. In order to understand these processes, predict their intertwined dynamics, and study human impact, environmental scientists and oceanographers seek to access a wide array of measurements across a range of spatial and temporal scales: e.g. temperature, concentration in dissolved oxygen, salinity, and biological data. Remote sensing—primarily based on satellite imaging techniques—offers unique capabilities to track some of these dynamics processes over very large spatial scales (Alcântara et al. 2010). However, monitoring over spatial scales of the order of a few km² can only be achieved using in situ measurements due to current limitations in the resolution of satellite imag-

¹ Massachusetts Institute of Technology (MIT), Cambridge, MA, USA

² Singapore University of Technology and Design (SUTD), Singapore, Singapore

ing technologies. Moreover, in situ measurements offer the possibility to collect samples as well as visual recordings, thereby complementing the measured data with additional possibly useful information.

Traditional in situ monitoring techniques rely on either a single autonomous surface vehicle (ASV) or a fixed network of sensors. Neither existing technology is suitable or efficacious for the robust monitoring and tracking of dynamic environmental features at the surface of aqueous environments. Environmental monitoring with high temporal resolution over small spatial scales can now be considered thanks to recent developments in mobile sensor networks using fleets of autonomous surface/underwater vehicles (Leonard et al. 2007; Duarte et al. 2016). Such developments are tied to the progress in ASV technology as well as distributed multi-robot systems (MRS).

The development of new designs of ASVs (a.k.a. Autonomous Surface Crafts or ASC) experienced a significant growth over the past two decades (Manley 2008). These robotic innovations were primarily driven by the availability of effective and affordable positioning systems. Over the same period, an explosive growth in sensor development provided compact and low-cost sensory suites, which were key to achieving successful missions with ASVs. Missions demonstrated include science and defense applications (Orton et al. 2009), environmental monitoring (Bayat et al. 2017) [e.g. algae bloom detection Zicarelli et al. (2016), pollution monitoring, etc.] and bathymetric mapping (Ferreira et al. 2009). It is worth highlighting that although the number of ASV prototypes reported in the literature is on the rise, there are still few fully operational commercial options available. This is in stark contrast with autonomous underwater vehicles (AUV) for which many commercial options are available.

Autonomous surface vehicles or vessels often have a design reminiscent of boats (e.g. kayak-like vessels or gliders) to optimize for either seakeeping, drag reduction, maneuverability, or other design features (Zicarelli et al. 2016). Buoys offer excellent seakeeping capabilities. For example, the Chesapeake Bay Interpretive Buoy System is a smart, stationary buoy designed by the National Oceanic and Atmospheric Administration to monitor various environmental features related to water quality, oceanography and meteorology (<http://buoybay.noaa.gov/>). This and similar buoys (<http://www.pentairenvironmental.com/products/smart-buoys.html>; <http://cleverbuoy.com.au/>; Fernández-Hermida et al. 2011) are stationary, bulky, expensive and cannot easily be redesigned to become rapidly deployable.

Although buoys were initially developed (and defined) as anchored platforms, some have been left adrift following oceanic currents (Srinivasan et al. 2016). Recently, a number of groups have started to consider buoys equipped with a mobility apparatus (Orton et al. 2009). A station-keeping autonomous buoy platform suitable for rapid deployment is

presented in Curcio et al. (2006). This system is built on a large surface craft (a kayak) that can navigate autonomously using the positioning information provided by GPS. While floater-like designs are not appropriate for long-distance travel, they offer high maneuverability combined with excellent seakeeping capabilities. These features make mobile sensing buoys very well suited to high temporal sampling of relatively small spatial areas. The pressing need for small, low-cost and rapidly deployable autonomous buoys is well captured by several recent reports (Nishida et al. 2015; Pico et al. 2016; Matos et al. 2016; Vesecky et al. 2007a).

Water surface monitoring with scalable deployment is essential for many practical—e.g., search and rescue, exploration, surveillance, water quality monitoring, and pollution control—and scientific purposes, since the ecological, biogeochemical, and physical processes are regulated by the dynamics of the water surface. By deploying a distributed system of autonomous buoys at the surface of waterbodies, each platform can monitor and detect a number of local environmental features such as temperature, pH, salinity, dissolved oxygen concentration, etc. This has been accomplished with moored buoys, however, aqueous environments are inherently dynamic, which significantly hampers the effectiveness of any fixed sensor network. A dynamic monitoring system of autonomous sensing buoys is appropriate as it provides spatiotemporal environmental sampling (Duarte et al. 2016).

As recently acknowledged by Duarte et al. (2016), there is a complete lack of demonstration of swarm/distributed MRS operations in unstructured and real-world environments. The experiments recently reported by Duarte et al. (2016) and Costa et al. (2016) represent a first attempt towards this goal with a maximum of 10 commercial mono-hull boats modified to host a single-board computer, Wi-Fi communications, a GPS module, and a compass. Their proof-of-concept experiment was focused on developing an automatic design approach for the decentralized control strategy based on evolutionary computing techniques. As promising as automatic design is, more progress is required in order to successfully apply it to collective operations in uncontrolled environments with a much larger number of platforms. Another notable proof-of-concept is the one achieved by Vicsek and collaborators who successfully implemented flocking strategies on a small flock of 10 quadcopters (Virágh et al. 2014; Vászárhegyi et al. 2014), using GPS for localization and a distributed communication approach.

Here, we report the design, construction, and testing of a fully distributed system of autonomous buoys performing adaptive deployment for applications in environmental monitoring. To the best of our knowledge, this distributed multi-robot system—referred to as “Bunch of Buoys” or BoB for short—is the largest such systems reported in the literature, with 50 units capable of operating in unstructured environ-



Fig. 1 A small fleet of autonomous surface vehicles. Top panel: 25 buoys of a BoB system collectively operating at Bedok Reservoir, Singapore. Bottom left: Buoys stacked up during transportation to the field site. Bottom right: 48 buoys lined up before deployment

ments. This collective of buoys possesses all the attributes of a swarm robotics system as per the commonly accepted definition given by Şahin (2005). Specifically, swarm robotics encompasses robots that (i) are autonomously evolving in the environment—with the possibility to interact with it, (ii) have local sensing and communication capabilities, (iii) do not have access to centralized control and/or to global knowledge, and (iv) cooperate to tackle a given task (Brambilla et al. 2013). Except for the use of global information provided by GPS receivers—using GPS does not impose any practical limit on scalability—our system fulfills all other criteria of a swarm robotics one. To date, we have deployed and tested a system of up to 50 units that are dynamically deployed over large surface areas of an uncontrolled, open-water environment (see Fig. 1). The demonstrations reported here are significant for several reasons: (i) the complete absence of any supporting infrastructure, (ii) the large number of custom-made platforms, (iii) the performance of maritime tasks—the vast majority of reports in the literature are with ground and aerial robots (Chamanbaz et al. 2017; Brambilla et al. 2013; Virágh et al. 2014), and (iv) the implications for real-life applications such as pervasive monitoring of aqueous environments.

As a standalone robotic platform, the autonomous sensing buoy design reported here is original in many respects: (i) its relatively small size (30 cm diameter) and light-weight (8 kg) while being self-righting, see e.g. Vesecky et al. (2007a, b), (ii) its omni-directional vectored propulsion achieves effective station-keeping and maneuverability, (iii) its low cost (\sim \$1000) and rapid assembly (in \sim 1 h) with minimal effort, (iv) its versatility in sensing capabilities, and (v) its notable on-board computing capabilities (equivalent to a laptop computer). This combination of features makes this

platform unique and potentially attractive for many scientific and industrial applications (Valada et al. 2014; Murphy et al. 2011).

The collective operation is designed using a behavior-based approach with a cooperative control strategy supported by distributed communication. The system can easily switch between its implemented collective behaviors, including flocking, navigation, and area coverage, to achieve different tasks (Couzin et al. 2005). The effectiveness of the distributed communication setup is verified by studying the success ratio of communications between units in the field. The dynamic scalability and flexibility in area coverage is quantified using a new metric based on the area of the largest Voronoi cell. Using this metric, we show the near-optimal scalability of the system and its capacity to respond to changes in the shape of the target surface.

2 Autonomous surface vehicle: mobile sensing buoy

The BoB project is aimed at designing and developing novel methodologies and systems for the effective and robust monitoring and tracking of dynamic environmental features—including physical, mechanical, chemical, and biological processes—relevant to planning and operations in complex coastal and inland water environments. BoB consists of a dynamic cooperative array of mobile buoys afforded with sensing capabilities. The platform was designed with scalability and pervasiveness in mind.

We design a compact, omni-directional, self-righting, robust, and watertight platform which is effective in both spatial and temporal coverage. A vectored propulsion system (Sect. 2.3) implemented with three pairs of motors allows the buoy to move through water surfaces at speeds of up to 1.0 m/s. The buoy is capable of self-localization using a GPS module (Sect. 2.5.1) and is designed to host a range of sensors to characterize its local environment (Sect. 2.2.2). A distributed mesh communication system (Sect. 2.4) allows the units to exchange sensed data, send and receive commands, and broadcast their state to neighboring buoys. An integrated single-board computer (Sect. 2.2.3) provides the unit with enough computational power to integrate these sources of information and process them on-board in order to autonomously determine its local behavior.

Careful design considerations are taken into account in order to support scalable assembly. We utilize a mold for the mechanical body to enable inexpensive batch manufacturing (Sect. 2.1). Electrical components are wired through a printed circuit board (PCB), which minimizes the time and effort required for assembly (Sect. 2.2). Our software stack is designed in a modular manner, which facilitates inclusion of new features and behaviors (Sect. 2.5). Wireless networks

are the primary means by which we recover logged field data and push software alterations, however, these are also possible via local USB connections.

Physical scaling and design considerations were developed for ease of replication and for consistency in production. The BoB units stack to optimize container utilization during transport (see Fig. 1).

In this section, we provide a comprehensive technical description of the hardware and software components of our autonomous platform. While the platform is designed to be part of a large-scale distributed MRS, it can also be used individually as an autonomous sensing unit.

2.1 Body

The body is a spherical hull with a truncated cap, yielding a diameter of 320 mm and a flange of 80 mm above the buoy centerline.

The heavier equipment (motors, batteries) is placed as low as possible in order to bring the center of gravity lower than the center of buoyancy. This ensures that the buoy is self-righting.

A semi-exploded rendered view of the CAD design for the buoy is shown in Fig. 2a. All the ports for electrical components (two USB ports, a battery charging connector, and GPS and XBee antennas) are easily accessible directly on the lid (see Fig. 2c), and the six mounting holes just below the waterline are used to equip sensors.

The sensing and monitoring operation performed by this platform requires a high capacity for station keeping, and thus the design is driven by maneuverability rather than speed. For instance, the isotropy of the body facilitates a vectored propulsion system—described in Sect. 2.3—that allows for excellent agility with near instantaneous direction changes.

The fluid flow around the buoy is analyzed using a computational fluid dynamics (CFD) software available in SolidWorks (see Fig. 3) at buoy speeds up to 2 m/s, which is on the higher range of expected velocities (see Fig. 8). These speeds correspond to a maximum Reynolds number $Re \approx 6.4 \times 10^5$ and Froude number $Fr \approx 1.13$. Simulations were performed to (i) estimate power needed at maximum speed, and (ii) understand the moment induced on the body by drag forces alone. This was used to empirically optimize motor pod placement such that the thrust moment would neutralize the drag moment resulting in level trim during steady-state translation. This also maximizes motor performance, since the thrust vector remains in the plane of motion. The overall estimate of drag coefficient for the buoy is $C_D = 0.55$ at nominal operating speeds in the 0.3–1.2 m/s range (see Fig. 3).

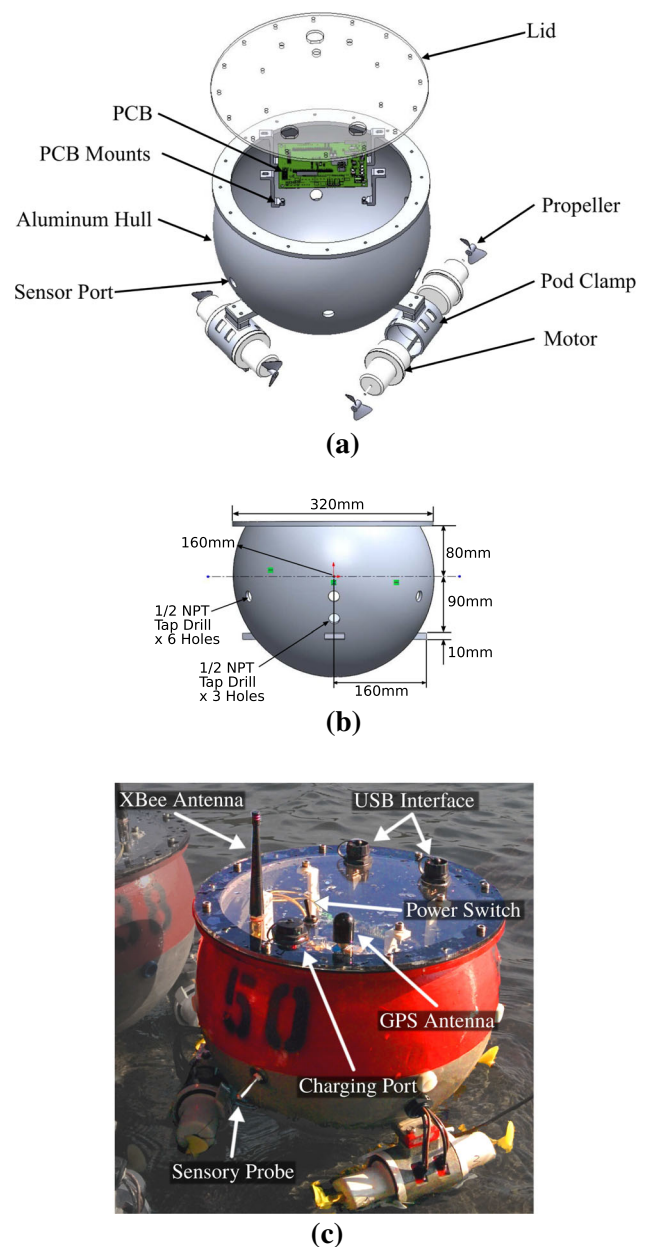


Fig. 2 Body of the buoy. **a** Semi-exploded rendered view of the CAD design. **b** Side-view with main dimensions. **c** Fully assembled BoB unit #50 in the field, ready to be deployed

2.2 Electrical subsystem

A block diagram of all the electric subsystems and their relation is shown in Fig. 4. In order to connect different components avoiding extensive wiring, a PCB was designed in-house to simplify the electronic assembly (see Fig. 5).

2.2.1 Power

The platform is powered by a Revolectrix® Blend435 Black Label 3-cell Lithium Polymer (LiPo) battery with 4.3 V per

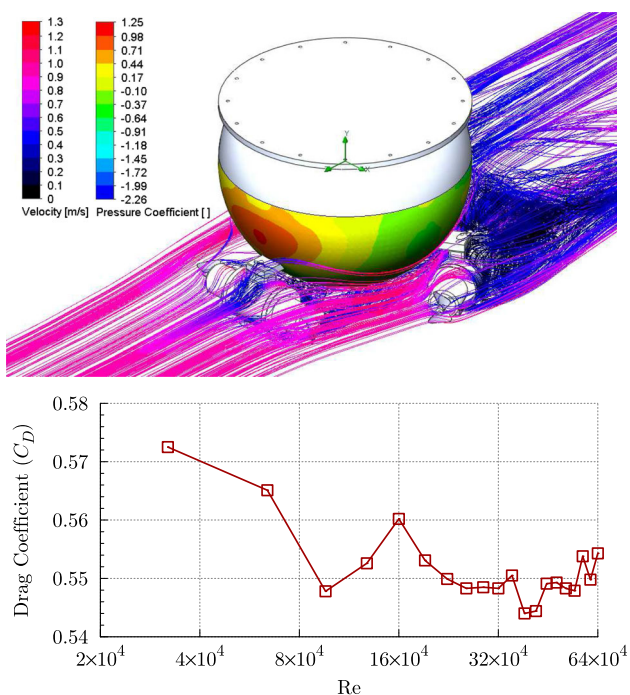


Fig. 3 CFD simulations of the hydrodynamics of the buoy. Top: streamlines around a towed buoy at 2 m/s (Reynolds number $Re \approx 6 \times 10^5$). Water is flowing from lower left to upper right. Bottom: drag coefficient C_D for a given Reynolds number Re (Color figure online)

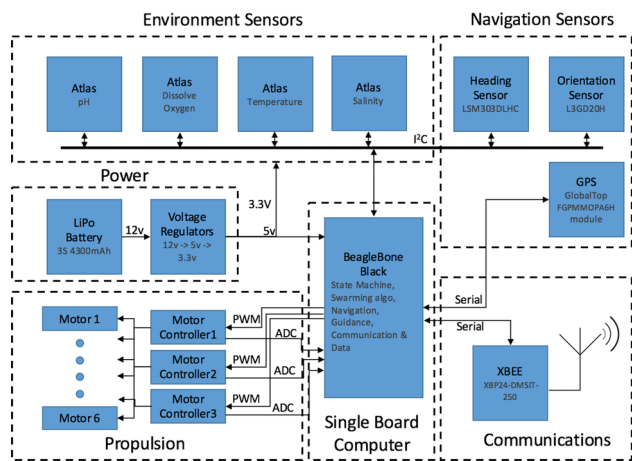


Fig. 4 Block diagram of the electric components. All the sensors, controllers, and communication modules are connected to a central single board computer that runs the cooperative control algorithms

cell and total capacity of 4600 mAh. The power consumption of the platform is estimated from the current drawn by the electronics and by the thrusters. For the electronics, one can consider a conservative fixed current consumption of 238 mA. For the thrusters, the built-in feedback from the motor controllers provides an accurate measure of the current consumption.

The battery provides a run time of about 2 h both in controlled lab settings and in field operations, given the usual

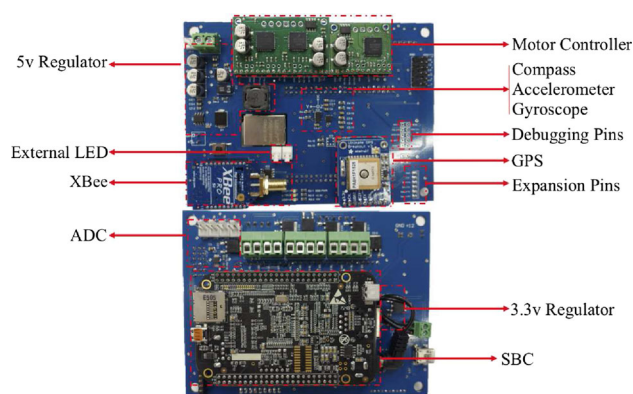


Fig. 5 Custom PCB design. Top: IMU, GPS, XBee and motor controllers; bottom: power supply, routing and interface to the Beagle Bone Black

tasks performed. Run time is highly dependent on the task, and it could be down to half the time in the worst case scenario (continuous translation at full speed.) The current design allows for extra battery packs to be loaded and thus extend the operating time of the platform.

The software limits the power that the motors can output depending on the estimated battery charge and, if it drops below 15%, the motors will be disabled. This is a safety measure to reduce the chance of battery degradation.

2.2.2 Sensors

A number of sensors are used both for navigation and characterization of the environment. These include:

- *Inertial Measurement Unit (IMU)* This module contains a compact STMicroelectronics LSM303 sensor, combining a three-axis digital accelerometer and three-axis digital magnetometer, and a STMicroelectronics three-axis digital L3GD20 gyroscope. The 16-bit linear acceleration reading can be scaled on the interval $\pm 2g$ (default) up to $\pm 16g$. Similarly, the magnetic field full-scale can be set on ± 1.3 up to ± 8.1 Gauss. The gyroscope provides a full scale of ± 4.36 rad/s up to ± 34.88 rad/s, low-power feature and high shock survivability.
- *Global Positioning System (GPS) Receiver* GPS signals are received by the Adafruit Ultimate GPS, capable of providing a fix rate up to 5 Hz and no-fix rate up to 10 Hz. The software implementation uses either 1–5 Hz on boot. The antenna reception sensitivity is -165 dBm with a capability of tracking up to 66 satellites (GPS and GLONASS constellations, not Beidou). The sensitivity of the signal is improved by using the JDGA XP263 Mini Active GPS Antenna, which provides a gain of 25 dB.
- *Environmental Sensing* Temperature, pH, dissolved oxygen, and salinity is measured with the Atlas Scientific

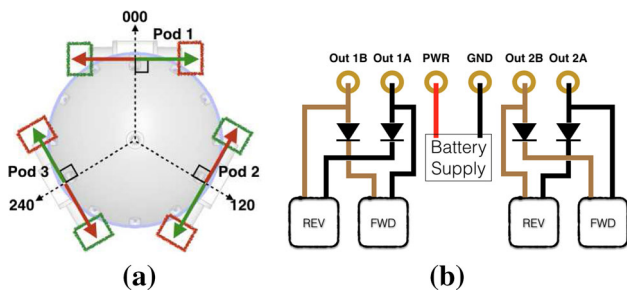


Fig. 6 Motors: **a** Pod 1 serves as the reference pod with each pod offset 120° from pod 1. Green arrows correspond to forward thrust and red arrows to reverse thrust as seen by the individual pods. **b** H-bridge diode configuration. For simplicity, only two pods are shown. **a** Thrust pod orientation and **b** diode configuration

sensors mounted on the standard, water-tight sealing interface of 1/2– and 3/4–inch NPT fitting. A dedicated circuit board for each module allows easy interface with serial and I2C communication protocols.

2.2.3 Processor

The processing unit of the platform is the single-board computer BeagleBone Black (BBB)—1 Ghz AM3358 processor, 512 MB DDR3 RAM, running a non-native Ubuntu distribution from a 16 GB SD-card as the operating system. The BBB was selected mainly for its 69 GPIO and 7 analog inputs that make it easy to expand and interface with other devices. Another important reason for choosing the BBB is that it provides software-defined peripherals as part of the Programmable Real-time Unit Industrial Control Subsystem (PRU-ICSS), allowing for instance pulse-width modulators (PWM) for the motors. With this integrated controller one does not need to add external microcontrollers to interface with the motors or other peripherals.

2.3 Motors

The buoy is afforded motor capabilities by implementing a vector propulsion system (Zoss 2016; Robert 2012). We use three separate pods to establish the overall thrust vector (see Fig. 6a). Each pod contains two motors (one forward and one reverse), requiring individual control of six motors. Only one of the two motors attached to a pod is active at a time due to inline diodes (see Fig. 6b). This configuration prevents any individual motor from producing negative work.

The motor controller in use is Pololu MC33926. It is able to supply a continuous current of 3A and a peak current of 5 A to a brushed motor at 5 – 28 V.

By following the reference system depicted in Fig. 6a, two equations are solved to compute the individual thrust required of each pod. First, we impose zero torque,

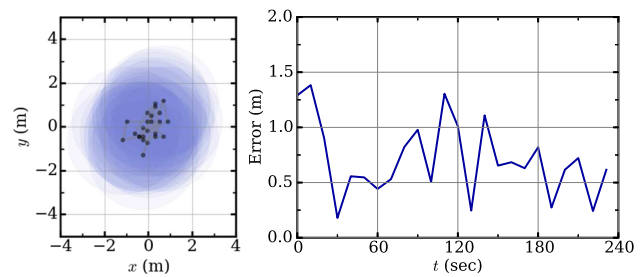


Fig. 7 Field tests taken with a moderate average current of 0.21 m/s and waves with height of 0.3 m and dominant period of 1 s. The buoy is able to maintain an average error of 0.71 m from the goal location (0,0) under GPS accuracy of ±2 m

$$\sum \mathbf{M}_{\text{Buoy}} = \mathbf{r}_1 \times \mathbf{T}_{\text{Pod1}} + \mathbf{r}_2 \times \mathbf{T}_{\text{Pod2}} + \mathbf{r}_3 \times \mathbf{T}_{\text{Pod3}} = \mathbf{0} \quad (1)$$

where $\mathbf{T}_{\text{Pod}i}$ is the thrust vector produced by pod i , and \mathbf{r} the moment radius. Since the pods are orientated such that $\mathbf{T}_{\text{Pod}i} = T_i(\hat{\mathbf{r}}_i \times \hat{\mathbf{e}}_z)$ (see Fig. 6a), Eq. (1) can be written as

$$\sum M_{\text{Buoy}_z} = T_1 + T_2 + T_3 = 0, \quad (2)$$

where M_{Buoy_z} is the moment about the local body z -axis (orthogonal to the base plane of the spherical cap of the body).

Second, we impose the sum of all thrust vectors to be in the direction of the desired buoy velocity

$$\sum \mathbf{F}_{\text{Buoy}} = \mathbf{T}_{\text{Pod1}} + \mathbf{T}_{\text{Pod2}} + \mathbf{T}_{\text{Pod3}} = \mathbf{T}_{\text{thrust}}, \quad (3)$$

where $\mathbf{T}_{\text{thrust}}$ is the requested thrust vector. We can obtain a solution for T_1, T_2, T_3 by solving the following set of equations:

$$\begin{bmatrix} 0 & \cos \theta_1 & \cos \theta_2 \\ 1 & \sin \theta_1 & \sin \theta_2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} T_{\text{thrust},x} \\ T_{\text{thrust},y} \\ 0 \end{bmatrix}, \quad (4)$$

with $\theta_1 = 210^\circ, \theta_2 = 330^\circ$.

This motor setup grants the buoy with the speed and maneuverability necessary to perform station keeping. Figure 7 shows a station keeping event over a period of just under 4 min with heavy wind and seas in the Charles River, Boston, MA. This test was performed in an environment with a temperature of 12.7 °C, a dew point of 3.89 °C, winds of 8.0 m/s, and a pressure of 999 hPa.

The characteristic speeds achieved in field experiments are presented in Fig. 8. When performing dedicated open water speed tests, the buoy achieved a modal speed of 0.82 m/s and mean speed of 0.70 m/s.

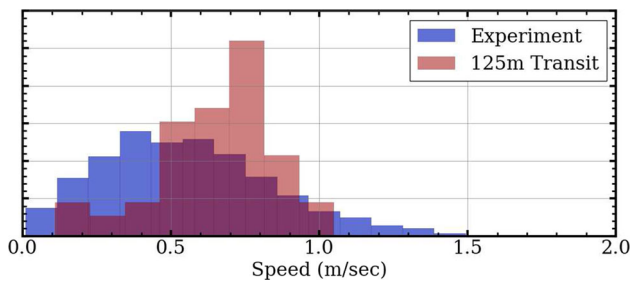


Fig. 8 Histogram of speeds achieved throughout two typical experiments in uncontrolled environments. Blue: aggregation behavior of 43 units reported in Sect. 4.1 and Fig. 15. Red: 125 m dedicated open water speed test. Water conditions were calm, with only mild surface ripples. Weather conditions are as reported in Sect. 4 (Color figure online)

2.4 Communications

Communication is achieved using an off-the-shelf, low-power wireless module. We use Digi's *XBee-PRO*® Series 1, based on the *DigiMesh 2.4* protocol.

The range of communication is extended using the Connectorized Quarter-Wave Monopoles (or Half-Wave Dipoles) Antenna from Linx Technologies®, providing 1.1 dBi of gain. With this, the communication range is about 310 m. The range was estimated with experiments in various environments: the SUTD football field, the East Coast Park car park, and the Bedok water reservoir. Using the 2.4 GHz frequency band, we observed interferences from nearby devices and the surrounding environment. Tests at Bedok reservoir were the least affected by interferences, being able to obtain a Received Signal Strength Indicator (RSSI) no lower than -82 dBm at 310 m.

The measured signal strength (RSSI) is shown in Fig. 9, where each data point is obtained by using the DigiMesh analysis tool between two XBee modules over 200 iterations of 32-byte messages. These results are in good agreement with the empirical formula $RSSI = -A - 10n \log_{10} D$ discussed in Kumar et al. (2009). Fitting the formula to our experimental data, we obtain

$$RSSI = -(17.8 \pm 2.5) - 10 \times (2.57 \pm 0.12) \log_{10} D, \quad (5)$$

where RSSI is expressed in dBm, and D is the distance between the modules measured in meters.

2.5 Software

The software is written in Python and C/C++ and built on the Robot Operating System (ROS) kinetic release (Quigley et al. 2009). In ROS, each subsystem is written as a package that runs as a standalone node, ensuring modularity and reusability.

The buoy operates with the following nodes:

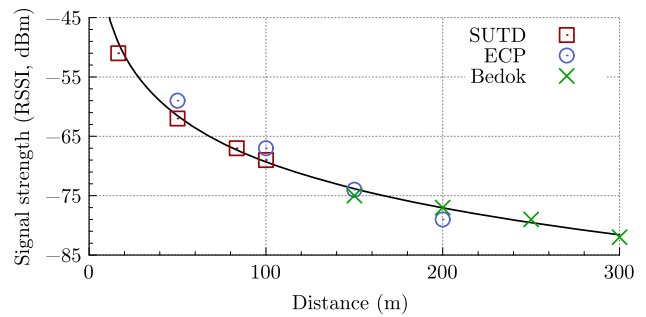


Fig. 9 Received signal strength indicator (RSSI) measured at the football field of SUTD, at the Singapore East Coast Park (ECP), and the Bedok reservoir. The continuous black line corresponds to the fit given by Eq. (5) from the empirical law in Kumar et al. (2009)

- *GPS* Used for navigation purposes; processes the GPS messages coming from the GPS module.
- *Attitude and Heading Reference System* Used for navigation purposes; processes data from the IMU sensor to give Attitude and Heading in the REP (ROS Enhancement Proposal) 103 reference system.
- *Finite State Machine Core* of the system; receives data from all other packages;
- *Scheduler* Evaluates timed commands from local files and/or received over radio;
- *Data recorder* Used for logging; enables/disables the recording of data exchanged over the TCP/IP stream between nodes.

2.5.1 Global positioning system (GPS)

The GPS node can be started with either a 1 or 5 Hz data rate. After initialization, the GPS node starts processing the following NMEA (National Marine Electronics Association) GPS messages:

- *GSA (Global Positioning System Fix Data)* We get the information of latitude, longitude, altitude and fix status; depending on the fix status, an error covariance matrix is defined and a *topic* message is created;
- *RMC (Recommended minimum specific GPS/Transit data)* This message is decoded only once, after a GPS fix is obtained. The purpose is to get the correct date and timing. This information is then used for setting the operating system's time to make sure all the buoys have consistent timing. The system time is used for logging purposes.

2.5.2 Attitude and heading reference system (AHRS)

The AHRS node reads data from the IMU (tri-axial gyroscope, accelerometers and magnetometers) to give an estimated attitude and heading expressed in ROS REP-103 stan-

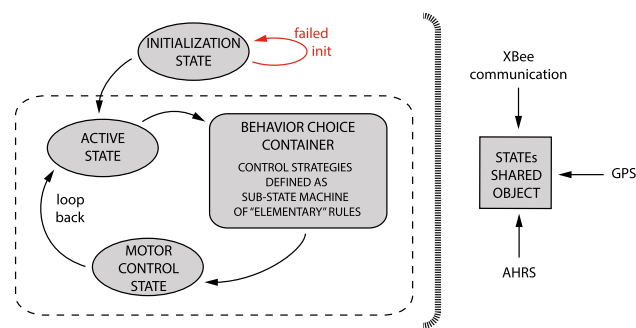


Fig. 10 A diagram representation of the Finite State Machine (FSM) (left). A *Shared Object* (right) is passed and can be modified by all the states, also receiving messages from other ROS nodes (GPS and AHRS) and communications from the XBee module

dard by using the Madgwick AHRS algorithm (Madgwick et al. 2011), which uses quaternion algebra and gradient-descent.

This node produces estimations at 25 Hz rate. Ideally, one would want to use this algorithm with an update rate of at least 100 Hz (Madgwick et al. 2011), but the limited computational power of the BBB restricts us to this relatively low update rate.

2.5.3 Finite state machine

The autonomous behavior of the platform consists of an iterative control function with three main stages: sensing (including localization and communication), decision-making (processing the sensed data to obtain a target destination), and movement (control the motors to move towards the destination). This behavior can be conceptualized as a finite state machine (FSM) where each of the stages corresponds to a state—or a set of states. The connection between states can be modified by external commands or environmental cues. The FSM is implemented using J. Bohren’s ROS *smach* package (Bohren and Cousins 2010).

Figure 10 shows the organization of the FMS: the machine starts in the *Initialization State* where communications and the shared resources are initialized. Once the initialization is complete and no error is reported, the machine enters the main loop via the *Active State*, where the contents of the communications received are processed and “flushed” into the shared memory. After that, the machine will enter the *Behavior Choice* state (a sub-machine itself) to generate the next waypoint, which will be sent to the *Motor Control State* in order to get body-fixed thrust instruction in the form of PWM signals that control how much power each motor-pair receives.

States communicate with each other and share information through a *Shared Object* that is initialized at start up and passed to every state. Any state can modify the content of the shared memory based on information obtained from other

ROS nodes, such as AHRS or GPS, or from communications with other platforms.

Particular attention is taken in defining a global, inertial, metric, right-handed coordinate system, with origin at the first GPS position received. All GPS and local measurements are converted to a metric displacement from the origin, simplifying the implementation of the cooperative control strategies. Note that since the coordinate system is defined with respect to the first GPS location, each platform uses its own private reference frame.

The *Behavior Choice* state container in Fig. 10 represents a cluster of states acting themselves as a sub-FSM, where each of the states corresponds to one of the “elementary” rules used to define the cooperative control strategies (see Sect. 3.1). This framework provides a flexible way to implement complex control strategies that can change on-the-fly during operation.

2.6 Design considerations and hindsight

2.6.1 Body

A first body prototype was manually fabricated out of styrofoam to enable initial tests of complete assembly and integration of the mechanical design and electronics. This made us realize the advantages of a modular design with the electronics mounted below the top lid for quick inspection and troubleshooting both in the lab and in the field—e.g. we can rapidly address faulty electronics by simply swapping the lid. Mounting holders were 3D printed as a fast solution to ensure the PCB is mounted parallel to the lid—to level the compass, though creating more screw holes leads to potential leaks. Furthermore, having access from the top lid to all ports connected to the main electrical components (two USB connecting ports, a battery charging connector, and GPS and XBee antennas) allows for charging and software inspection while keeping the platform fully assembled.

Plastic material was initially considered for the hull due to its light weight and good anti-biofouling properties. However, the cost of the mold for the injection process led us to consider a more cost-effective solution based on aluminum. Metal casting process turned out to be the fastest and cheapest way to fabricate 10–50 units of the body of our buoy. Lids were laser-cut from an acrylic sheet.

Upon reception of the first batch of 10 hulls, and after a complete assembly and initial testings, some previous design choices were modified for the subsequent batches: less side openings (future sensors placing) and less screw holes for securing the lid onto the hull, as fewer screws proved equally as effective in keeping the integrity of the unit while reducing overall assembly time.

A flange was included on the top to aid stackability and portability. Lastly, 3D printing of motor pod clamps is

impractical (slow manufacturing process), fairly expensive and not durable. In response to this costly inefficiency, aluminum casting was chosen for the pod clamps as well.

2.6.2 Electronics

The first prototypes were made out of many breakout boards, but then a PCB was designed in order to scale up production. The boards were sent out to a fabrication house to be assembled. Since the boards have to go through the oven once for the components mounted on the top and then a second time for the ones on the bottom, we placed all the sensitive components on the side that only goes through the reflow process once.

Having a reliable measure of remaining power in the system proved to be a complex task. The simplest way is to integrate the estimated current consumption assuming that the battery provides a constant voltage. However, this provided only a qualitative estimation that requires the battery to be fully-charged whenever the system is initialized.

Overestimating the remaining power can translate into an abuse of the Lithium-Polymer (LiPo) batteries, causing dangerous discharge that can compromise the whole platform beyond repair. For this reason, our team is currently working on an implementation of a proper battery management unit.

The experiments presented in this work were carried out in open spaces with good GPS coverage. But even in these conditions, we observed a high variability in the GPS localization. The estimated position of some buoys would in some occasions present short-lived pulse-like fluctuations of up to ~ 10 m. Since the cooperative control strategies rely on position reported by the GPS, these fluctuations would translate into sizable differences between the expected and observed collective behavior (see Fig. 19).

The Real-Time-Kinematic (RTK) technique is a potential improvement of this basic GPS localization. RTK requires an accurate base station and a secondary communication channel from this base to each agent, which may affect the scalability of the system.

Other challenges were encountered when dealing with environmental sensing. The calibration procedure itself is typically hard to automate, and the variability in data reported by the different low-cost sensing modules introduces errors in the reconstructed field that are hard to estimate or account for.

2.6.3 Communications

The swarming-inspired design principles and, in particular, the cooperative control strategies are such that the successful collective operation of this system does not require a reliable, dynamic, and global communication network between all the agents. The motion of each platform at a given time

is determined solely by its own state and the current state of neighboring agents. Therefore, only short-range local communications are required. The agents broadcast their state to any agent in range, and do not require specific knowledge of *which* agents are in range.

However, during development and experimentation one usually wants a finer-grained control of the system: ideally, an operator on a base station should receive information from all the buoys deployed and be able to operate on them. In order to have this level of control, we equipped the platforms with communication modules that can build a dynamic, global routing network such that messages can be sent from the base station to any specific buoy deployed, even if it is out of direct range of communication.

2.6.4 Software

During the initial stages of development, the code controlling the buoy was written in Python to facilitate fast prototyping and iterative design. Once the design was mature, we decided to port the code to ROS as it provides out-of-the-box solutions for two pervasive problems in MRS operating in uncontrolled environments, which are detailed in what follows.

After a number of iterations in the design of the platform, one naturally ends up with a heterogeneous MRS with different capabilities. These can range from slightly different motors or battery capacity to different sensor capabilities (most platforms have temperature sensors, but two of them have salinity and pH sensors instead). If one is not careful, different designs may require a diversity of code that has to be maintained independently. ROS offers an efficient answer to this problem by providing packages that can be run with parameters defined from a launch file.

Operating the MRS in an uncontrolled environment makes experimentation significantly costlier and more time-consuming than lab testing. It is therefore critical to maximize the actionable insight gained from each experiment. In this regard, the logging features of ROS and the possibility to “replay” any communication between nodes provided us with a way to solve problems, analyze data, and test hypothesis retroactively on previous field test.

3 System design

3.1 Cooperative control algorithms

Ideally, one wants to operate the collective at the system level, issuing global objectives. For example, commanding the collective to perform aggregation (a.k.a. rendezvous in space), pattern formation, dynamic area coverage, mapping, boundary detection, collective sensing, distributed search and rescue, etc. These global collective objectives have to

be mapped into individual, agent-specific commands. Such mapping is known as the *cooperative control algorithm*, and the particular form it takes determines the effectiveness of the large-scale collective behavior of the system (Couzin et al. 2005).

A straightforward strategy to ensure a certain degree of scalability and robustness is to impose spatial locality of the algorithm, meaning that the action of an agent is solely determined by the information gathered on a certain neighborhood of its location. Likewise, flexibility to changing environments benefits from imposing temporal locality, meaning that the action is purely determined by the current state of the said neighborhood.

In practice, we implement these conditions by considering iterative algorithms (or update rules) that control the trajectory of an agent by updating its target velocity according to

$$\mathbf{v}_i(t + \Delta T) = \mathbf{F}(t, \mathbf{r}_i(t), \{\mathbf{r}_j(t)\}_{j \sim i}, \{\mathbf{v}_j(t)\}_{j \sim i}), \quad (6)$$

where i is the agent index, ΔT the sampling time at which this rule is applied, \mathbf{v}_i the velocity of agent, \mathbf{r}_i its position, and $j \sim i$ is the set of agents in the neighborhood of i —its “neighbors”, excluding i .

The control rules presented in what follows do not specify how this neighborhood is constructed. The literature on flock modeling includes many operational examples of how to construct such neighborhoods. In particular, Fine and Shell (2013) present a unified framework and introduce a data-flow template composed of sensing, flock member detection, neighbor selection, motion computation, and physical motion. In our work, since the buoys sense other buoys exclusively via radio-communication, the neighborhood of a given agent will always be a subset of the agents present in its communication range. We have not imposed any selection rule for neighbors, so the neighbors of a given buoy are by default *all* the agents detected by it: i.e. all flock member detected are selected according to the lexicon introduced in Fine and Shell (2013). Since the set of sensed agents depends on their motion, the set of neighbors will change in time with dynamics coupled to the motion of the agents themselves (Komareji and Bouffanais 2013; Bouffanais 2016).

In the following, we discuss three collective behaviors—flocking, navigation, and area coverage—implemented in BoB and tested in field experiments. These behaviors are constructed as a superposition of nondimensional “elementary” rules—aggregation, repulsion, and geofencing force, see Appendix A—used as building blocks, so that

$$\mathbf{v}'_i = v_0 \sum_{\mathbf{g}_i \in \mathcal{G}_i} \mathbf{g}_i, \quad (7)$$

where v_0 is a constant characteristic speed of the buoy, and \mathcal{G}_i is the set of elementary dynamical behavioral rules con-

sidered for agent i . Explicit time-dependence is omitted for conciseness, i.e. $\mathbf{v}_i = \mathbf{v}_i(t)$ and $\mathbf{v}'_i = \mathbf{v}_i(t + \Delta T)$. Note that these behavioral rules can be agent-specific and at any given time different agents may be following different rules.

Together, these behaviors provide a useful toolkit to collectively deploy a large number of ASVs on aquatic surfaces, control their motion, and position them optimally to collectively monitor large surfaces with arbitrary shapes.

3.1.1 Flocking

Collective behavior, and flocking in particular, has received particular attention from several scientific communities—first by the computer graphics community (Reynolds 1987), and then followed by physicists (Vicsek and Zafeiris 2012). Subsequently, the control community established a formal framework (Olfati-Saber et al. 2007; Ren and Beard 2008; Jadbabaie et al. 2003), which has been put into practice and expanded in the context of multi-robot systems and swarm robotics (Brambilla et al. 2013; Turgut et al. 2008).

The classical “flocking” model, a staple of collective motion studies, assumes the motion of an agent is composed of three terms: aggregation, avoidance, and alignment. This model and its many implementations has been discussed elsewhere (Reynolds 1987; Vicsek and Zafeiris 2012). It has been successfully implemented by Vicsek et al. for a flock of 10 quadcopters (Vásárhelyi et al. 2014).

To achieve a cohesive flocking behavior with our autonomous buoys, we implement the rule

$$\mathbf{v}'_i = v_0 \sum_{j \sim i} \left\{ \left(\frac{1}{n_i} - \frac{r_0^2}{3r_{ij}^2} \right) \frac{\mathbf{r}_{ij}}{r_{ij}} + \frac{1}{n_i} \frac{\mathbf{v}_j}{v_0} \right\}, \quad (8)$$

where n_i is the number of neighbors, $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$, $r_{ij} = \|\mathbf{r}_{ij}\|$, and r_0 is a free parameter that controls the equilibrium nearest-neighbor distance. The first part attracts the agents towards the center of the collective and repels them from nearby agents. The second one causes the agents to align their velocities to move in a coordinated fashion.

The factor $1/3$ in the repulsion term is introduced phenomenologically so that a large number of agents following this rule will navigate in a lattice-like formation where each agent is approximately at a distance r_0 from its nearest neighbor, see Sect. A.1.

Flocking is the default behavior and it grants the system an extra layer of robustness against failures in the communication network. As discussed in Sect. A.4, the system can perform several collective actions even if fewer than all agents receive the appropriate command. This is made possible due to the “uninformed” agents passively mimicking the motion of nearby agents performing the same collective action the informed agents are actively performing.

3.1.2 Collective navigation

An elementary way to control a team of ASVs is to set a target location \mathbf{T} for the group to reach. Naturally, one does not wish for *each* agent to reach the same point, but for the collective to gather around it. In order to navigate the group to a specific location, we implement the behavioral rule

$$\mathbf{v}'_i = v_0 \left\{ \frac{\mathbf{T} - \mathbf{r}_i}{\|\mathbf{T} - \mathbf{r}_i\|} - \sum_{j \sim i} \frac{r_0^2}{3r_{ij}^2} \frac{\mathbf{r}_{ij}}{r_{ij}} \right\}. \quad (9)$$

This collective behavior is similar to flocking, with the difference that the agents aggregate at a specified point and station-keep at that location.

The target \mathbf{T} can be a fixed location or a time-dependent path. It may also be set to a particular agent's position, $\mathbf{T}(t) = \mathbf{r}_l(t)$, implementing a leader–follower scheme that allows an external controller to manually navigate the “leader” agent l and have the collective to follow its motion.

In principle, deploying the buoys by using this rule requires information to be transmitted globally to every single agent, as they all need to know the value of \mathbf{T} . When operating a large number of agents in open, uncontrolled environments, it is extremely difficult to guarantee that a particular message will reach all agents, so one would assume this mode of control is not very scalable nor robust. However, because of the default flocking behavior, only a fraction of the agents need to explicitly follow this behavioral rule in order for all of them to have a cohesive collective behavior where they assemble around the target (see Sect. A.4). This is a good example of an “indirect” flow of information, where an agent can react to an environmental change even if it is unable to detect the change itself. For instance, the simulations discussed in Sect. A.4 show how some agents can “passively” follow a leader without having explicit information on the leader's location.

Figure 11 shows an example of how an agent following Eq. (9) reaches a target when a group of aggregating agents is in the way. If this agent has r_0 comparable to that of the group members, they will move around their equilibrium position to open up space for the agent as it travels through the group (Fig. 11a). If the group members are fixed at their positions, the dynamics (9) allows the moving agent to circumvent the group and reach the goal (Fig. 11b). Alternatively, if the moving agent has a r_0 considerably smaller than the fixed group members, it will simply “sneak” through the collective to reach its goal (Fig. 11c).

3.1.3 Area coverage

The intended use of the large-scale networked array of mobile sensing units presented here is to monitor and characterize

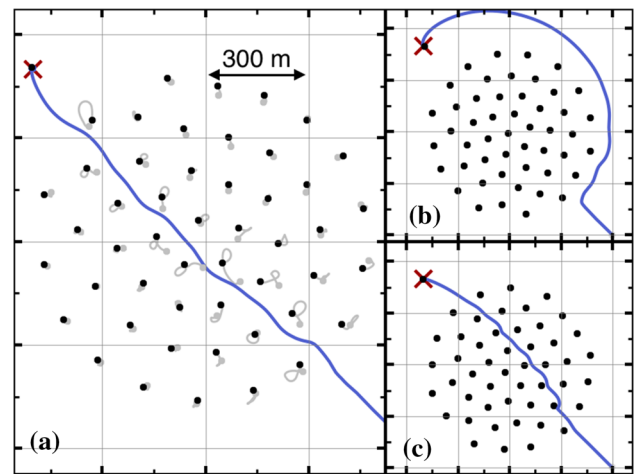


Fig. 11 Simulation of collective avoidance behaviors in group navigation. A group of $N = 49$ agents aggregate following the default flocking behavior with $r_0 = 100$ m. An additional agent i (lower right corner) is introduced and directed with Eq. (9) towards a given goal (red cross mark) placed opposite the collective. Depending on the parameters, one can observe **a** yielding behavior, **b** bypassing behavior, or **c** sneaking behavior (Color figure online)

aquatic environments in regions of interest, which may vary depending on the application. For example, the networked array may be deployed in a specified area in a harbor to assist in marine operations by monitoring key environmental and flow parameters. More interestingly, the area to monitor might not be specified externally or in advance, but instead be defined dynamically by the collective of agents itself. By local processing of the sensed data, the agents may determine the shape in which to self-deploy in order to track a particular temperature profile, oil spill, or a range of biological markers.

How to deploy the agents for an efficient collective monitoring of a given region may depend on the application and the feature being tracked. For example, if the agents are tasked with sensing a scalar field (such as temperature) they should be able to spread as uniformly as possible across the region of interest, i.e. perform what is known as a “blanket coverage” (Gage 1992). By contrast, if their task involves tracking how a substance spreads, such as a precursor to an algae bloom scenario—dissolved oxygen content, the agents should position themselves uniformly across the contour of the region instead, performing what is known as a “barrier coverage.” Here, we describe the algorithm implemented on BoB to perform a blanket coverage of simply connected regions.

In many cases, the shape of the region to monitor will evolve with some arbitrary, unknown dynamics. Thus, the agents should have a responsive behavior that allows them to dynamically spread across arbitrary shapes, and be capable of adapting to changes in a timely manner. To obtain this behavior, we define a control algorithm that follows Eq. (6), which is inspired by the potential-field approach to area coverage (Howard et al. 2002), and consists of a term attracting

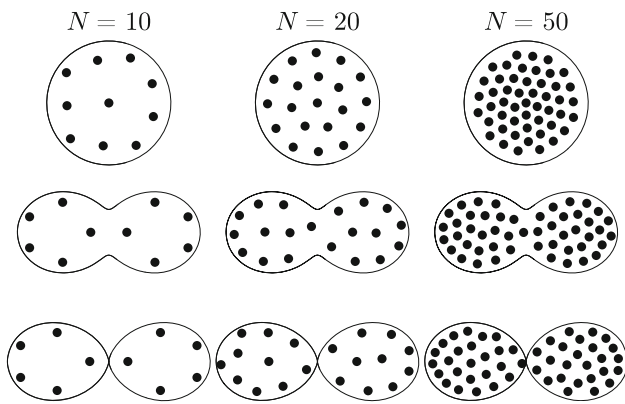


Fig. 12 Equilibrium configurations of the dynamic area coverage behavior. Long-time asymptotic configuration obtained with simulations of N agents following Eq. (11) for the area given by Eq. (12) with $\alpha = 0$ (top row), $\alpha = 1$ (middle row), and $\alpha = 2$ (bottom row)

the agents towards the interior of the area and a term repelling the agents from each other.

Given an arbitrary region described by

$$A(\mathbf{r}) < 0, \tag{10}$$

where A is a signed distance function (or at least a function that increases monotonically outside the region), we define the area coverage behavioral rule as

$$\mathbf{v}'_i = v_0 \left\{ \frac{1}{1 + \exp(-A(\mathbf{r}))} \frac{-\nabla A}{\|\nabla A\|} - \sum_{j \sim i} \frac{r_0^3}{r_{ij}^3} \frac{\mathbf{r}_{ij}}{r_{ij}} \right\}, \tag{11}$$

where the attraction term (proportional to $-\nabla A$) is scaled in such a way that it is $\simeq 0$ outside the area to cover ($A > 0$) and $\simeq 1$ inside of it ($A < 0$).

The equilibrium distribution of agents following this behavior is presented in Fig. 12 for different number of agents and the different shapes of the area given by Eq. (12).

3.2 Distributed communication

The collective behavior requires some level of *interaction* between the agents, as they need to have access to information on the state of nearby agents. In the framework of distributed and swarm robotics, this interaction can be accomplished by three means: (i) interaction via their environment, (ii) interaction via sensing, and (iii) interaction via communication (Bayindir and Şahin 2007). In the context of monitoring of large bodies of water, the “interaction via communication” paradigm is more appropriate, so that each agent actively broadcasts a range of state variables to nearby agents. Indeed, the ability to control the content of each message being broadcast allows for more sophisticated information exchanges as

compared to what can be achieved using the “information via sensing” paradigm.

To ensure the robustness, scalability, and flexibility of the collective, the agents should be able to communicate in a distributed fashion establishing a dynamic—i.e., switching—communication network where nodes can be added or subtracted during operations. To grant these features to the buoys, we have equipped them with XBee-PRO modules that are capable of creating a distributed mesh network (see Sect. 2.4). This network automatically reconfigures as the agents move and enter or leave each other’s communication range.

An example of a typical communication network obtained during field tests is depicted in Fig. 13. At the given instant, this network is composed of an isolated node (top-right), a clique of three nodes (top-left), and a cluster of highly connected nodes with an articulation node (green) that bridges the cluster with the clique. The temporary isolation of one or more agents does not hinder the collective operation of the rest—the XBee modules allow for unconnected components to work independently, and for these to dynamically join other components or split into smaller groups.

This communication network is used by the buoys to continuously broadcast their state—that is, their current GPS coordinates, heading, behavior, and environmental data sensed (if any)—at a rate of 0.1 Hz. Maximum expected communication range is about 310 m (see Sect. 2.4) and the modules are capable of relaying messages through multiple hops in the network, which means that in principle any module can broadcast their state globally to all the agents in the collective. However, our field experiments show that when tens of buoys are operating in a real and uncontrolled environment, the communication is far from perfect, and the effective communication range is significantly smaller.

The ratio of successful communications obtained during a dynamic monitoring field experiment (see Sect. 4.3) is presented in Fig. 14 as a function of the distance between buoys. These results provide a measure of the effective communication range in a large and dynamic network of mobile XBee units. As one increases the number of buoys deployed, interference between them will cause more messages to drop (compare $N = 40$ against $N = 20$ in Fig. 14). This presents a clear example of why the distributed control algorithm should be designed to provide a robust collective behavior under imperfect communication.

4 Field experiments

We have conducted a series of field experiments deploying up to 45 buoys on a calm body of water (Bedok Reservoir, Singapore) to test the efficacy of the platform in performing aggregation, leader–follower, and dynamic mon-

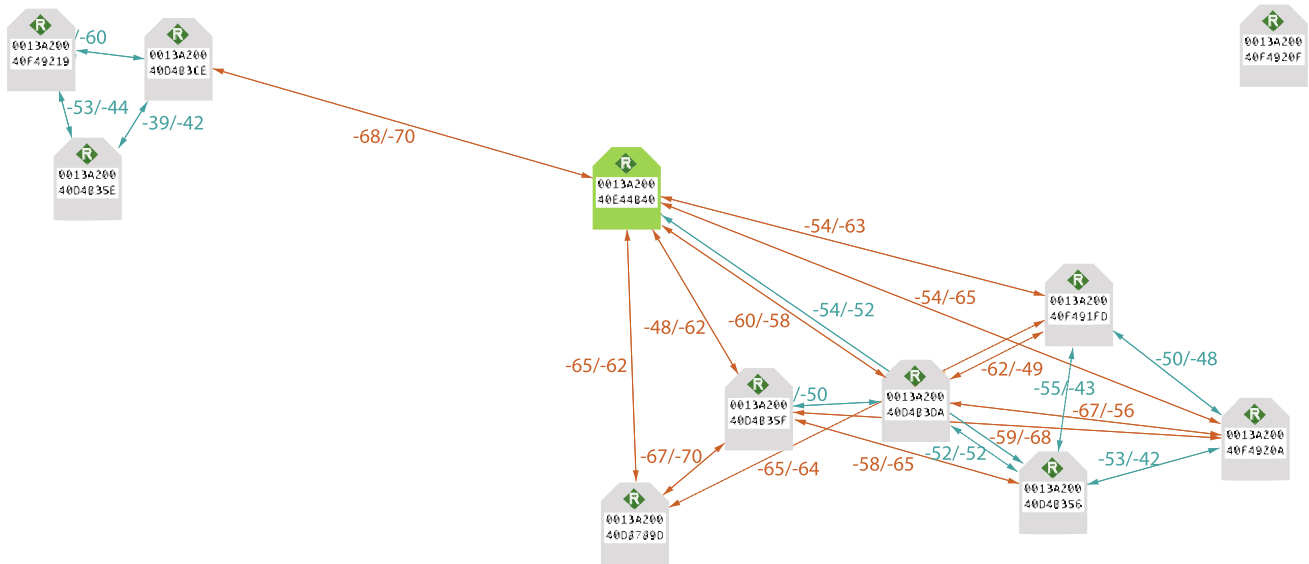


Fig. 13 Sample of the mesh network established by 11 buoys deployed in the field (configuration qualitatively equivalent to the configuration reconstructed from GPS). The edges between modules denote direct communications, which may be strong (blue) or weak (orange). Numbers adjacent to the edges indicate the signal strength (RSSI, see

Sect. 2.4) in dB, in each direction. The green module serves as an articulation node, establishing a bridge that allows communication between the top-left clique and the central cluster. This screenshot was obtained using Digi's XCTU free software, with XBee modules configured as routers (Color figure online)

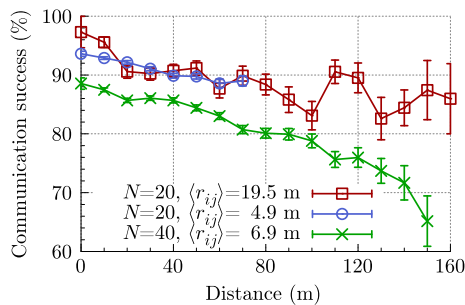


Fig. 14 Histogram of the relative successful communications established between two buoys separated by a certain distance inside a collective of N buoys spread with a mean nearest-neighbor distance of $\langle r_{ij} \rangle$. Data measured during the Dynamic Monitoring experiment (see Sect. 4.3)

itoring under real-world conditions. The experiments were performed under usual weather conditions of a equatorial climate—including rain showers, with average temperatures of around 29 °C, dew point of 25 °C, winds of 1.8 m/s, and pressures around 1009 hPa.

These experiments confirmed that, at the unit level, the buoys are capable of goal seeking, station keeping, and distributed communication. At the system level, the tests provided clear evidence that the system can efficiently perform large-scale collective behavior in the face of distributed, fragmented communications. In what follows, we show that the collective behavior of the system and its typical response times are in good agreement with the predictions from simu-

lations, thereby demonstrating that the mesh network strategy and the decentralized control algorithms provide a robust framework that can be scaled up to these system sizes.

4.1 Aggregation

The paradigmatic use case of BoB is to measure scalar fields such as temperature, wave height, or density of a certain chemical in waterbodies. The precision of the field reconstruction will depend on the spread of the buoys and their density in the region of interest. Thus, given a number of agents, we can deploy them in a loose configuration to have a large area covered or with a compact configuration to have a high resolution over a small area.

To illustrate the situation where we want to switch from one to the other, we perform an aggregation field test where a group of 45 buoys is initially deployed following Eq. (9) with a target destination set 100 m from the insertion point. After the buoys reach the goal and form the equilibrium lattice arrangement around it, the aggregation test is performed by suddenly reducing the initial r_0 from 50 m down to 5 m.

The trajectory of the buoys during the aggregation test is shown in Fig. 15. The entire aggregation event takes 8.2 min, during which the area covered by the buoys decreases from about 125,000 m² to 1250 m². Average buoy speed for the outer ring is 0.4 m/s, while the ones in the inner rings aggregate at proportionately lower speeds of 0.2 and 0.1 m/s respectively.

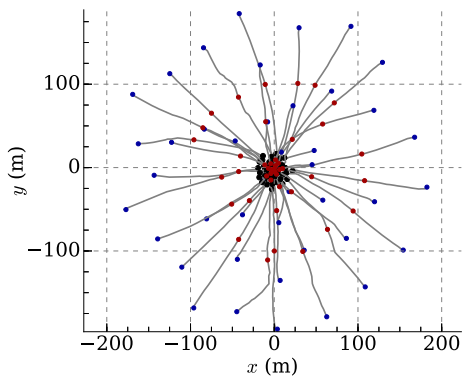


Fig. 15 Field tests of aggregation behavior of a collective of 45 buoys, conducted on a $\sim 1 \text{ km}^2$ body of water on a moderately windy day. Buoy trajectories for an aggregation event where the repulsion strength is suddenly decreased from $r_0 = 50$ down to 5 m. Blue dots indicate the initial position of the buoys, while red and black indicate the 50% and terminal positions respectively. Note that, after the experiment, the logs of only 43 units could be retrieved (Color figure online)

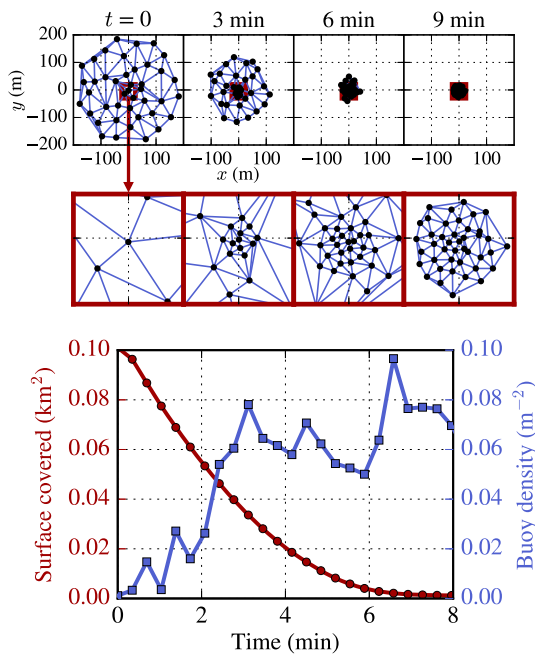


Fig. 16 Evolution of the distribution of buoys during the aggregation field test. Top: delaunay triangulation of the buoys' positions during the aggregation event. The central region is zoomed in the red squares below. Bottom: total area covered and density at the center of the collective

Figure 16 shows the evolution of the total surface covered by the buoys and the density at the center of the collective during field test. We use a Delaunay tessellation field estimator (Schaap 2007) to measure the density of buoys at a given point. Although the whole aggregation process takes approximately 8 min, the density at the origin grows to its saturation value in about 3 min.

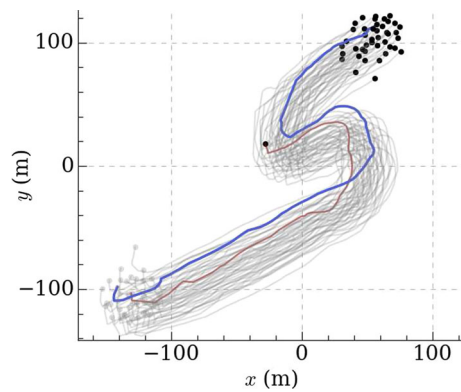


Fig. 17 Field tests of leader–follower behavior of a collective of 45 buoys with $r_0 = 5 \text{ m}$. The group traverses $\sim 400 \text{ m}$ in calm waters. The trajectory of the leader buoy is marked with a blue trail and that of the followers with gray. The red trail highlights the trajectory of a buoy lagging behind due to low battery. Note that, after the experiment, the logs of only 43 units could be retrieved (Color figure online)

4.2 Leader–follower

The leader–follower behavior (Couzin et al. 2005) was tested by setting one “leader” buoy to manual control while switching the rest to group navigation [Eq. (9)] setting the leader’s position as the target. The trajectories of the buoys (Fig. 17) show the collective is capable of maintaining a tight formation while following a leader even in the face of sharp turns in its trajectory. Even though there is a degraded member (red trail), the system successfully executes the intended behavior. Videos of this experiment are available at (https://youtu.be/fhg1rIX_y3A; <https://youtu.be/Qe-wZOi3ONs>).

4.3 Dynamic monitoring

To test the capacity of BoB to perform dynamic monitoring, we deployed 20 buoys following the area coverage behavior [Eq. (11)] and tasked them with covering the area

$$A_{\alpha, \mathbf{e}}(\mathbf{r}) = \frac{r^2 - R_{\alpha, \mathbf{e}}^2(\hat{\mathbf{r}})}{10R_0^2}, \tag{12}$$

where

$$R_{\alpha, \mathbf{e}}(\hat{\mathbf{r}}) = \frac{R_0}{2} \frac{2 - \alpha + 3\alpha(\hat{\mathbf{r}} \cdot \hat{\mathbf{e}})^2}{\sqrt{1 + \alpha/2 + 11\alpha^2/32}}, \tag{13}$$

$R_0 = 25 \text{ m}$, and initial $\alpha = 0$. The buoys were injected sequentially over the span of 8 min (see Fig. 18 and <https://youtu.be/RPJSvC-X-Vs>) and they autonomously spread through the exploration circle, covering it homogeneously.

One way of measuring the quality of the deployment is to partition the exploration area into a Voronoi tessellation based on the position of the buoys, such that each agent has an

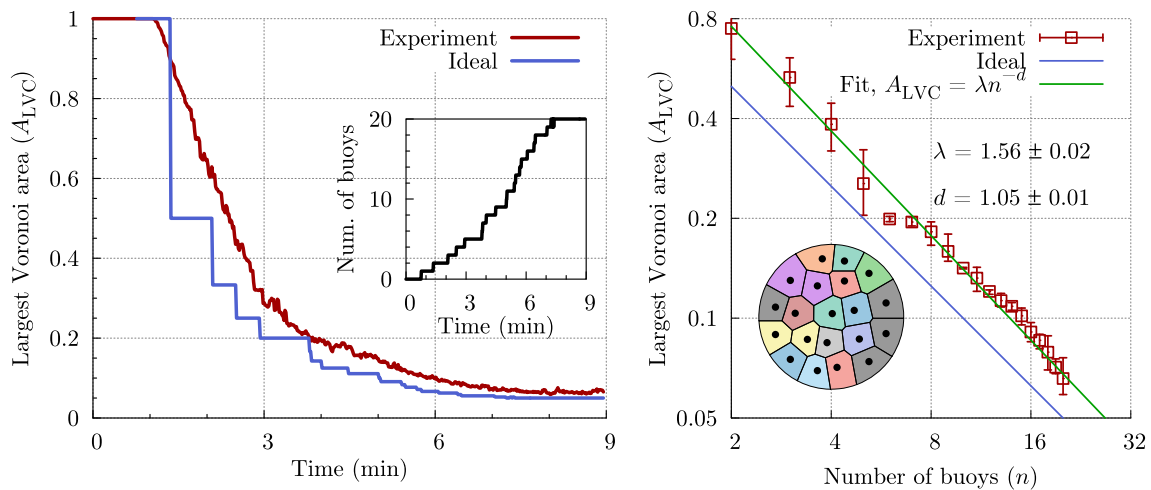


Fig. 18 Dynamic scalability of a group of 20 buoys performing dynamic area coverage. Left: evolution of the largest Voronoi cell’s area with respect to the total exploration area, A_{LVC} (red), compared with the “ideal” case of $1/n$ (blue). The inset shows the number of buoys

n inside the circle at a particular time. Right: correlation between A_{LVC} and number of buoys n for the same data as the top panel. Fitting the experimental data shows that the efficiency of the system grows roughly linear with the number of agents, $A_{LVC} \sim n^{-1.05}$ (Color figure online)

associated cell consisting of all the points of the exploration region closer to it than to any other agent. These cells can be considered the areas “designated” to each buoy to monitor, and ideally each of these would encompass $1/N$ of the total area.

With this tessellation, the efficiency of the system in area coverage can be quantified using the area of the largest Voronoi cell (LVC) as a measure of how solicited the most solicited buoy is. The LVC area relative to the total exploration area, A_{LVC} , is presented in Fig. 18. As expected, the size of the cell decreases with time as more buoys are deployed. By fitting A_{LVC} to a power-law of the number of buoys n currently deployed, we find that

$$A_{LVC}(n) = (1.56 \pm 0.02)n^{-1.05 \pm 0.01}, \quad (14)$$

which corroborates the scalability of the system as A_{LVC} scales roughly as $1/n$, for n up to ~ 20 .

Once all the buoys are deployed, we test the flexibility of the system by changing the shape of the exploration area with time. We let α oscillate between 0 and 2 and ascertain that the buoys are able to successfully adapt to these changes and re-position themselves accordingly, see Fig. 19. Corresponding videos showing the trajectories of the individual buoys corroborated by GPS tracking are at (<https://youtu.be/KBEURgyPxXI>; https://youtu.be/hlBNjHS_Q7s).

The time evolution of A_{LVC} for this flexibility experiment with 20 buoys is presented in Fig. 20. For the duration of the experiment, A_{LVC} oscillates between 5.6% ($N \times A_{LVC} = 1.12$) and 12.3% ($N \times A_{LVC} = 2.46$), with an average value slightly below 8%. The values obtained in the field experiment (red line) follow closely the prediction from simulations

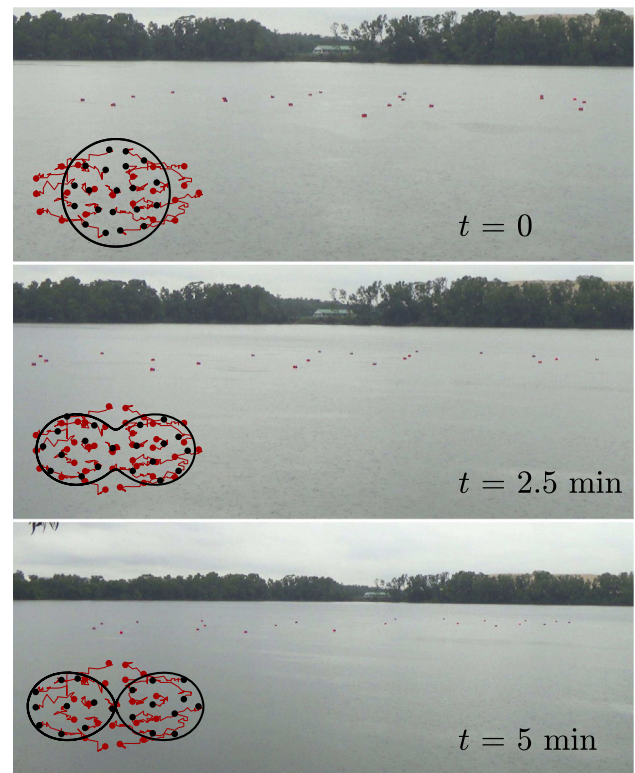


Fig. 19 Snapshots of the flexibility field experiment. For full video, see Ref. https://youtu.be/hlBNjHS_Q7s. Insets: Position (black) and path (red) of the buoys reconstruction from GPS localization data (Color figure online)

(blue line). Interestingly, the experimental A_{LVC} can have large and short-lived pulse-like oscillations where the value is considerably larger than expected (e.g. at $t = 9, 17, 20$,

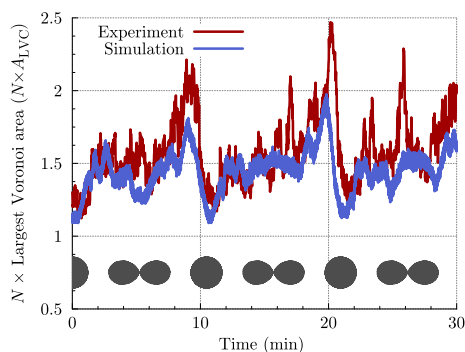


Fig. 20 Flexibility of a group of $N = 20$ buoys performing dynamic area coverage. The buoys are tasked with covering an area of approximately 200 m^2 by following Eq. (11) while the surface dynamically oscillates between a circle to a two-lobbed “dumbbell”. Evolution of the largest Voronoi cell’s area with respect to the total exploration area scaled with the number of agents, $N \times A_{LVC}$. The ideal (minimum) value of this metric depends on the geometry of the area. We find that for the circle ($\alpha = 0$) it is $N \times A_{LVC}^I \simeq 1.1$ and for the dumbbell ($\alpha = 2$) around $N \times A_{LVC}^I \simeq 1.5$. The dynamic shape of the exploration area is depicted in gray

and 26 min). These “pulses” are a manifestation of the many perturbations one faces when operating MRS in uncontrolled environments, from brief inaccuracies in the GPS localization to adverse environmental conditions (winds and currents) and limited communication rate between agents (currently set at 0.1 Hz).

Despite all the potential sources of stochasticity in the system’s performance, the fact that these oscillations are short-lived and do not seem to accumulate over time illustrates the stability of the system. The high responsiveness of each buoy, given by its omni-directional motion design, and the robustness provided by the distributed control setup and mesh-network communications allow the system to quickly recover a near-optimal configuration after deviating appreciably from it.

4.4 Experimental hindsight

Performing field experiments with a relatively large MRS requires addressing a number of challenges: scout for field test locations, organize transportation, setup and deploy the system, troubleshoot and recover, post-process and analyze the data.

The size of the system imposes serious limitations on the waterbodies adequate for testing and experimentation. Small-scale experiments were carried out on land by moving the units on trolleys to manually simulate the thruster directions. The data gathered from such tests is qualitative at best and they are hard to scale up because they require about as many researchers as buoys.

Transportation of a fairly large number of units to the field test site is a logistical challenge. A design of the hull

that allows to easily stack many units in a compact formation proved to be of paramount importance (see bottom left quadrant in Fig. 1). However, having to transport these in a van further limited the set of waterbodies eligible for testing, and is currently on the critical path to performing large-scale open-sea testings.

Setup and initialization of the agents in the field poses challenges that should not be underestimated. With BoB, the longer the time elapsed between activating the first buoy and the last, the shorter the testing time will be. We found that this setup time could be shortened if, instead of initializing the buoys on land and injecting them in the waterbody from a single point, we used a kayak to first spread the buoys around the body and then initialize all of them.

In order to maximize the usefulness of field tests during development, special attention was granted to having a fast recovery of faulty units and on-site debugging. For this reason, we developed a graphical interface for data processing and monitoring of the MRS, which gathers the messages broadcast by the buoys to report the state of the whole system (estimated locations, current behavior, battery levels, etc.) that would allow us to identify potential problems of faulty units in situ. Having instant feedback through the graphical interface would, in many instances, be enough to be able to make an educated guess on whether a particular observed issue was related to the mechanics, the electronics, or the software.

Post-processing and data analysis of the experimental results can introduce a considerable overhead time if not dealt with properly. For instance, post-processing can be automated by having the units connect to a wireless network in the lab and uploading their logged data to a computer that processes them to generate a video of the reconstructed GPS trajectories along with any relevant metric (see e.g. <https://youtu.be/KBEURgyPxXI>).

In our experience, we found that having post-processed data in under 2 h from the completion of the experiment translates into a responsive research as it allows us to tune the field tests based on the feedback, especially when several field tests were performed on consecutive days.

5 Conclusion

The paradigm of a large fleet of small, low-cost, autonomous robots with sensing capabilities offers huge potential for the pervasive and persistent monitoring of coastal and inland water environments. Here, we demonstrate the possibilities of this paradigm by deploying up to 50 autonomous buoys over an area of about 1 km^2 in an inland water body and tasking them with collective operations relevant for monitoring, such as aggregation, leader–follower, and area coverage.

We have implemented cooperative control algorithms enabling the team of buoys to perform collective flocking, navigation, and area coverage behaviors with new expressions for the local update rules. These collective behaviors have been tested in a series of field experiments in open waterbodies that allow us to characterize the performance of the system under real-world conditions and imperfect, distributed communications.

We introduce a novel metric to quantify the scalability and flexibility of the deployment for area coverage, and find that the performance of the system scales approximately linearly with the number of deployed buoys. The flexibility, tested by dynamically changing the target area, is found to be in good agreement with predictions from simulations.

The fleet of buoys were developed with the swarm robotics design principles of scalability, robustness, and flexibility in mind. In addition, each unit of this distributed MRS consists in a self-propelled buoy with an original design enabling effective operations as a standalone autonomous surface vehicle. The reported field experiments allow us to quantify the scalability of the system up to the group sizes considered and its flexibility to adapt to intrinsically dynamic environments. Given the system design, its efficiency is expected to hold for a significantly larger number of agents, provided that the distributed communication strategy can be scaled accordingly.

For the ultimate goal of large-scale pervasive area monitoring, the buoys have been equipped with rudimentary sensors for temperature, pH, and salinity. Our current research effort focuses on ways to incorporate the environmental sensing into the collective behavior.

Acknowledgements This work was supported by Grants from the Temasek Lab (TL@SUTD) under a Seed Grant #IGDS S15 01021, a MOE-Tier 1 Grant #SUTDT12015003, and the National Research Foundation Singapore under its Campus for Research Excellence and Technological Enterprise programme. The Center for Environmental Sensing and Modeling is an interdisciplinary research group of the Singapore MIT Alliance for Research and Technology.

A decentralized control algorithm: elementary rules

The collective behaviors described in the main text are obtained with control algorithms composed of five elementary behavioral rules. Here, we present each rule and discuss their implementation details.

A.1 Repulsion

Based on classical two-body repulsion forces in physics, we model the repulsive behavior as

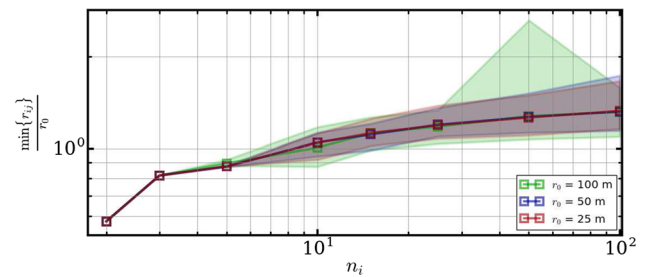


Fig. 21 Nearest-neighbor distance achieved during an aggregation event, where $\mathcal{G} = \{\mathbf{g}^R, \mathbf{g}^{AG}\}$ as a function of the number of neighbors, n_i

$$\mathbf{g}_i^R = - \sum_{j \sim i} \frac{\alpha_R^d \mathbf{r}_{ij}}{r_{ij}^d r_{ij}}, \quad (15)$$

where $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ and $r_{ij} = \|\mathbf{r}_{ij}\|$. The parameter α_R determines the strength of the repulsion and $d > 1$ its multipole order, controlling how the repulsion decreases with distance. We have used $d = 2$ for the flocking and navigation behaviors, and $d = 3$ for dynamic monitoring.

The repulsion strength can be used to control the “equilibrium” distance between neighboring agents when the repulsion term is combined with some attractive term. The relation between the nearest neighbor distance and the repulsion strength for agents behaving according to $\mathcal{G} = \{\mathbf{g}^R, \mathbf{g}^{AG}\}$ is presented in Fig. 21. The simulations show that

$$\alpha_R = r_0 / \sqrt{3} \quad (16)$$

is a good phenomenological relation to have agents separated by $\simeq r_0$ for the numbers of agents considered in this work.

In the case of dynamic area coverage, we are not interested in fixing a certain inter-agent distance but to make them spread as much as possible in the area. One should thus scale the repulsion strength with the total area to explore S per agent, i.e. $r_0^2 \propto S/N$. To explore the area (12), we set in Eq. (11) to

$$r_0 = 1.3R_0 / \sqrt{n_i}, \quad (17)$$

where R_0 is the same as in Eq. (12). We use the number of neighbor of each agent, n_i , instead of the total number of buoys N because the latter is a global information that is, in general, inaccessible to the agents.

A.2 Alignment

Flocking behavior typically contains an element of orientation alignment or “viscosity” between agents, essential to synchronize the collective motion in realistic dynamical settings. It provides a “smoother” collision avoidance behavior

and allows the collective to move while the agents keep in formation.

We have implemented velocity alignment as

$$\mathbf{g}_i^{AL} = \frac{1}{n_i} \sum_{j \sim i} \frac{\mathbf{v}_j}{v_j}. \tag{18}$$

If the behavior of the agents is dominated by this term, the collective will perform heading consensus and align themselves in a common direction of movement.

A.3 Aggregation

This dynamical behavioral rule leads the agents to collectively undergo an aggregation process. It is implemented as

$$\mathbf{g}_i^{AG} = \frac{1}{n_i} \sum_{j \sim i} \frac{\mathbf{r}_{ij}}{r_{ij}}. \tag{19}$$

Setting $\mathcal{G}_i = \{\mathbf{g}_i^R, \mathbf{g}_i^{AG}\}$ in Eq. (7) makes the collective perform a collision-free rendezvous in space at an unspecified location. We have chosen to sum the normalized distances instead of using the center of mass because simulations showed that this yields an aggregation process where the sum of all the agents' displacements is typically lower than if one were to use $\mathbf{g}^{AG} \propto \sum \mathbf{r}_{ij}$, and thus the present form is less demanding on the total power consumed by the collective.

A.4 Target navigation

In order to deploy the collective to a specific target location \mathbf{T} at the surface of water, we simply set the following rule in the direction of the target destination:

$$\mathbf{g}_i^T = \frac{\mathbf{T} - \mathbf{r}_i}{\|\mathbf{T} - \mathbf{r}_i\|}. \tag{20}$$

In principle this behavior requires the location of the target be sent globally to each agent in the collective. However, since the agents perform flocking by default, only a fraction of them need to explicitly use \mathbf{g}^T in order for the whole system to aggregate around \mathbf{T} .

This is illustrated in Fig. 22, obtained performing a simulation of 51 agents performing a leader–follower scenario (i.e., the target \mathbf{T} is the position of a particular “leader” agent.) The agents are initially placed in the equilibrium configuration corresponding to $\mathcal{G} = \{\mathbf{g}^{AG}, \mathbf{g}^R\}$. A leader starts moving in a straight line at constant speed when a certain ratio of the agents become “active” followers by changing \mathbf{g}^{AG} to \mathbf{g}^T . These so-called active followers are units receiving a direct command from the leading platform. The long-time asymptotic behavior of the system shows that only around half of the agents need to be actively following in order for the whole group to collectively follow the leader.

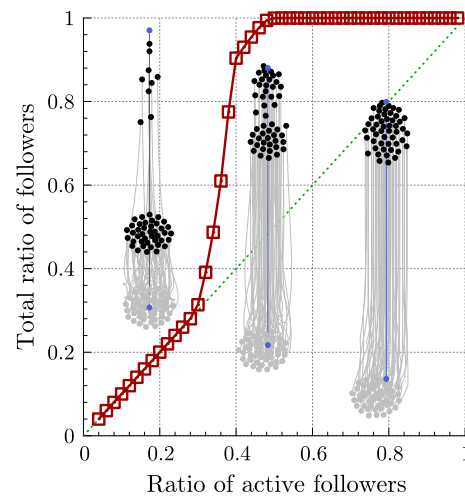


Fig. 22 Collective behavior in the face of imperfect communication. Simulation of a leader–follower scenario where one leader moves in a straight line at constant velocity and a certain fraction of agents are “active” followers behaving according to $\mathcal{G} = \{\mathbf{g}^T, \mathbf{g}^R\}$ while the rest behave according to the default $\mathcal{G} = \{\mathbf{g}^{AG}, \mathbf{g}^R\}$. The total ratio of followers is defined as the ratio of agents that have the same velocity as the leader in the long-time asymptotic state. The insets illustrate three typical outcomes: if only a small fraction of agents is actively following the leader, the collective will split in two and the majority of agents will aggregate and remain stationary. If half of the agents are active followers, the whole collective will follow the leader albeit in an elongated, diffuse formation. If the active followers compose a large fraction of the agents, the rest of them will “passively” follow the leader without requiring to know its position

A.5 Geofencing

To perform dynamic area coverage, we define a “geofencing” rule that attracts the agents towards the area $A < 0$ as

$$\mathbf{g}_i^{GF} = -\frac{1}{1 + \exp(-A(\mathbf{r}))} \frac{\nabla A}{\|\nabla A\|}. \tag{21}$$

The direction of this term ($-\nabla A$) causes the agents to move towards decreasing A , i.e. towards the interior of the surface. The scaling $(1/(1 + \exp(-A)))$ is such that $\|\mathbf{g}^{GF}\| \simeq 1$ outside the area and $\|\mathbf{g}^{GF}\| \simeq 0$ inside, so this term will only affect the agent significantly when outside the region of interest.

References

Alcántara, E. H., Stech, J. L., Lorenzetti, J. A., Bonnet, M.-P., Casamitjana, X., Assireu, A. T., et al. (2010). Remote sensing of water surface temperature and heat flux over a tropical hydroelectric reservoir. *Remote Sensing of Environment*, 114(11), 2651–2665.

Applied Complexity Group. 51 networked buoys swarming, https://youtu.be/fhg1rIX_y3A.

Applied Complexity Group. Dynamic area coverage (geofencing) field test, https://youtu.be/hlBNjHS_Q7s.

Applied Complexity Group. Dynamic environmental monitoring using swarming mobile sensing buoys, <https://youtu.be/Qe-wZOi3ONs>.

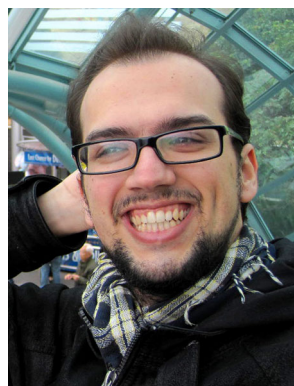
- Applied Complexity Group. Field test—Dynamic area coverage: Flexibility experiment. <https://youtu.be/KBEURgyPxXI>.
- Applied Complexity Group. Field test—Dynamic area coverage: Scalability experiment. <https://youtu.be/RPJSvC-X-Vs>.
- Bayat, B., Crasta, N., Crespi, A., Pascoal, A. M., & Ijspeert, A. (2017). Environmental monitoring using autonomous vehicles: A survey of recent searching techniques. *Current Opinion in Biotechnology*, 45, 76–84.
- Bayindir, L., & Şahin, E. (2007). A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering & Computer Sciences*, 15(2), 115–147.
- Beşiktepe, Ş. T., Lermusiaux, P. F. J., & Robinson, A. R. (2003). Coupled physical and biogeochemical data-driven simulations of Massachusetts Bay in late summer: Real-time and postcruise data assimilation. *Journal of Marine Systems*, 40, 171–212.
- Bohren, J., & Cousins, S. (2010). The SMACH high-level executive [ROS News]. *IEEE Robotics Automation Magazine*, 17, 18–20.
- Bouffanais, R. (2016). *Design and control of swarm dynamics Springer briefs in complexity*. Singapore: Springer.
- Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7(1), 1–41.
- Buoy, Clever. <http://cleverbuoy.com.au/>.
- Chamanbaz, M., Mateo, D., Zoss, B. M., Tokić, G., Wilhelm, E., Bouffanais, R., et al. (2017). Swarm-enabling technology for multi-robot systems. *Frontiers in Robotics and AI*, 4, 12.
- Costa, V., Duarte, M., Rodrigues, T., Oliveira, S. M., & Christensen, A. L. (2016). Design and development of an inexpensive aquatic swarm robotics system. In *OCEANS 2016-Shanghai* (pp. 1–7). IEEE.
- Couzin, I. D., Krause, J., Franks, N. R., & Levin, S. A. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025), 513.
- Curcio, J., McGillivray, P., Fall, K., Maffei, A., Schwehr, K., Twigg, B., et al. (2006). Self-positioning smart buoys, the "un-buoy" solution: Logistic considerations using autonomous surface craft technology and improved communications infrastructure. In *Proceedings of OCEANS 2006* (pp. 1–5).
- Duarte, M., Costa, V., Gomes, J., Rodrigues, T., Silva, F., Oliveira, S. M., et al. (2016). Evolution of collective behaviors for a real swarm of aquatic surface robots. *PLoS ONE*, 11(3), e0151834.
- Duarte, M., Gomes, J., Costa, V., Rodrigues, T., Silva, F., Lobo, V., et al. (2016). Application of swarm robotics systems to marine environmental monitoring. In *OCEANS 2016-Shanghai* (pp. 1–8). IEEE.
- Fernández-Hermida, X., Durán-Neira, C., Lago-Reguera, M. D., Rodríguez-Alemparte, C., & Martín-Rodríguez, F. (2011). Hidroboya: An autonomous buoy for real time high quality sea and continental water data retrieval. In *OCEANS 2011 IEEE-Spain* (pp. 1–7). IEEE.
- Ferreira, H., Almeida, C., Martins, A., Almeida, J., Dias, N., Dias, A., et al. (2009). Autonomous bathymetry for risk assessment with ROAZ robotic surface vehicle. In *OCEANS 2009-Europe* (pp. 1–6). IEEE.
- Fine, Benjamin T., & Shell, Dylan A. (2013). Unifying microscopic flocking motion models for virtual, robotic, and biological flock members. *Autonomous Robots*, 35(2), 195–219.
- Gage, D. W. (1992). Command control for many-robot systems. Technical report, DTIC Document.
- Home | Chesapeake Bay Interpretive Buoy System, <http://buoybay.noaa.gov/>.
- Howard, A., Matorić, M. J., & Sukhatme, G. S. (2002). Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In H. Asama, T. Arai, T. Fukuda, & T. Hasegawa (Eds.), *Distributed Autonomous Robotic Systems 5* (pp. 299–308). Tokyo: Springer.
- Jadbabaie, A., Lin, J., & Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48, 988–1001.
- Komareji, M., & Bouffanais, R. (2013). Resilience and controllability of dynamic collective behaviors. *PLoS ONE*, 8, e82578.
- Kumar, P., Reddy, L., & Varma, S. H. (2009). Distance measurement and error estimation scheme for RSSI based localization in wireless sensor networks. In *2009 Fifth IEEE conference on wireless communication and sensor networks (WCSN)* (pp. 1–4). IEEE.
- Leonard, N. E., Paley, D. A., Lekien, F., Sepulchre, R., Fratantoni, D. M., & Davis, R. E. (2007). Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1), 48–74.
- Madgwick, S., Harrison, A., & Vaidyanathan, R. (2011). Estimation of IMU and MARG orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics* (pp. 1–7). IEEE.
- Manley, J. E. (2008). Unmanned surface vehicles, 15 years of development. In *OCEANS 2008* (pp. 1–4). IEEE.
- Matos, A., Almeida, R., & Cruz, N. (2016). Man portable acoustic navigation buoys. In *Proceedings of OCEANS 2016-Shanghai* (pp. 1–6).
- Murphy, R. R., Steimle, E., Hall, M., Lindemuth, M., Trejo, D., Hurlebaus, S., et al. (2011). Robot-assisted bridge inspection. *Journal of Intelligent & Robotic Systems*, 64(1), 77–95.
- Nishida, Y., Kojima, J., Ito, Y., Tamura, K., Sugimatsu, H., Kim, K., et al. (2015). Development of an autonomous buoy system for AUV. In *Proceedings of OCEANS 2015-Genova* (pp. 1–6).
- Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215–233.
- Orton, P. M., McGillis, W. R., Moisan, J. R., Higinbotham, J. R., & Schirtzinger, C. (2009). The mobile buoy: An autonomous surface vehicle for integrated ocean-atmosphere studies. In *AGU Spring Meeting Abstracts*.
- Pico, G., Miranda, J., Marentes, K., & Tosunoglu, S. (2016). Multi-purpose autonomous buoy. In *Proceedings of the 29th Florida Conference on Recent Advances in Robotics, Miami, Florida* (pp. 128–144).
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., et al. (2009). ROS: An open-source robot operating system. In *ICRA workshop on open source software* (Vol. 3).
- Ren, W., & Beard, R. (2008). *Distributed consensus in multi-vehicle cooperative control*. Berlin: Springer.
- Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21, 25–34.
- Robert, A. H. (2012). Infrastructure for large-scale tests in marine autonomy. Master's thesis, Massachusetts Institute of Technology.
- Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. In E. Şahin, & W. M. Spears (Eds.), *Swarm Robotics: SAB 2004 International Workshop Santa Monica, CA, USA, July 17, 2004* (pp. 10–20). Springer, Berlin.
- Schaap, W. E. (2007). *The Delaunay tessellation field estimator*. Ph.D. thesis, University of Groningen.
- Srinivasan, R., Zacharia, S., Sudhakar, T., & Atmanand, M. A. (2016). Indigenous drifting buoys for the Indian ocean observations. In *OCEANS 2016 MTS/IEEE Monterey* (pp. 1–6). IEEE.
- Smart Buoys | Products | Pentair Environmental Systems, <http://www.pentairenvironmental.com/products/smart-buoys.html>.
- Turgut, A. E., Çelikkanat, H., Gökçe, F., & Şahin, E. (2008). Self-organized flocking in mobile robot swarms. *Swarm Intelligence*, 2(2–4), 97–120.
- Valada, A., Velagapudi, P., Kannan, B., Tomaszewski, C., Kantor, G., & Scerri, P. (2014). Development of a low cost multi-robot autonomous marine surface platform. In K. Yoshida, & S. Tadokoro (Eds.), *Field and Service Robotics. Springer Tracts in Advanced Robotics* (Vol. 92, pp. 643–658). Berlin: Springer.

- Vásárhelyi, G., Virágh, C., Somorjai, G., Tarcai, N., Szorenyi, T., Nepusz, T., et al. (2014). Outdoor flocking and formation flight with autonomous aerial robots. In *Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3866–3873).
- Vesecky, J., Laws, K., Petersen, S., Bazeghi, C., & Wiberg, D. (2007a). Prototype autonomous mini-buoy for use in a wireless networked, ocean surface sensor array. In *Proceedings of IEEE international geoscience and remote sensing symposium* (pp. 4987–4990).
- Vesecky, J. F., Laws, K., Petersen, S. I., Bazeghi, C., & Wiberg, D. (2007b). Autonomous minibuoy prototype for a coordinated, wireless networked, ocean-surface-sensor array. In *OCEANS 2007-Europe* (pp. 1–5). IEEE.
- Vicsek, T., & Zafeiris, A. (2012). Collective motion. *Physics Reports*, 517, 71–140.
- Virágh, C., Vásárhelyi, G., Tarcai, N., Szörényi, T., Somorjai, G., Nepusz, T., et al. (2014). Flocking algorithm for autonomous flying robots. *Bioinspiration & Biomimetics*, 9(2), 025012.
- Ziccarelli, L., Dellor, R., Johnson, R., Schmitz, H., O'Reilly, T., & Chavez, F. (2016). A novel method of obtaining near real-time observations of phytoplankton from a mobile autonomous platform. In *OCEANS 2016 MTS/IEEE Monterey* (pp. 1–5). IEEE.
- Zoss, B. M. (2016). Design and analysis of mobile sensing systems: An environmental data collection swarm. Master's thesis. Massachusetts Institute of Technology.



Brandon M. Zoss is currently serving as a Lieutenant (LT) in the United States Navy. His tours of duty have consisted of several fast attack (SSN) and a guided missile (SSGN) submarine platforms. In his roles he assisted in the successful completion of the first ever east coast Dual Dry Deck shelter Naval Special Warfare mission, other missions vital to national security, multiple Operational Reactor Safeguards Examinations, two Diving Operational Readiness Assessments, as well as an extensive and successful Major Maintenance Period. As a LT in the Navy, Brandon is qualified in Submarines, Naval Reactors Engineer and as a Salvage Diving officer. Brandon earned his S.M. Mechanical Engineering and S.M. Naval Architecture and Marine Engineering from MIT in 2016 concurrent with military obligations, during which time he conducted research into the design and application of swarm robotics systems, such as B.o.B. Additionally, Brandon holds a Masters in Engineering Management from Old Dominion University (also completed in 2016), in which his research efforts focused on developing qualitative metrics to streamline engineering programs on the micro-level. Brandon has future aspirations of continuing his work into the distributed autonomous systems as military obligations permit.

As a LT in the Navy, Brandon is qualified in Submarines, Naval Reactors Engineer and as a Salvage Diving officer. Brandon earned his S.M. Mechanical Engineering and S.M. Naval Architecture and Marine Engineering from MIT in 2016 concurrent with military obligations, during which time he conducted research into the design and application of swarm robotics systems, such as B.o.B. Additionally, Brandon holds a Masters in Engineering Management from Old Dominion University (also completed in 2016), in which his research efforts focused on developing qualitative metrics to streamline engineering programs on the micro-level. Brandon has future aspirations of continuing his work into the distributed autonomous systems as military obligations permit.



David Mateo was born in Barcelona, Catalonia, in 1985. He received the B.Sc. degree in Physics and M.Sc in Astrophysics and Particle Physics from the University of Barcelona in 2008 and 2009, respectively. In 2008, he joined the Atomic and Nuclear Physics group at University of Barcelona and obtained the Ph.D. degree in 2013 studying the emergent behaviors of low-temperature many-body systems. He is currently a Postdoctoral Research Fellow at the Singapore University of Technology and Design (SUTD) where he holds the position of Chief Swarming Officer at the Applied Complexity Group. His research interests include emergent collective behaviors, information transmission and diffusion in networks, swarm robotics, and control of complex systems.

His research interests include emergent collective behaviors, information transmission and diffusion in networks, swarm robotics, and control of complex systems.



Yoke Kong Kuan is a Ph.D. candidate at SUTD. He received his Bachelor's Degree in Mechanical Engineering from the University of Minnesota and a Master of Science in Mechanical Engineering from the University of Illinois at Chicago. His Master's thesis was in the field of reconstruction of blood vessel medical images (2D–3D) using Radial Basis Functions. Prior to SUTD, he worked in the U.S.A, A*star, Ngee Ann Poly, and BCA. His research areas are in swarming

systems and networked control systems. He also interested in Mechanobiology and Product Design (esp for extreme affordability).



Grgur Tokić is a postdoctoral associate in Vortical Flow Research Laboratory at MIT. He has a Diploma Engineer degree in Naval Architecture from the University of Zagreb (2006), and a Ph.D. in Mechanical Engineering from MIT (2016). His research interests include the hydrodynamics of ocean waves, fish swimming, and swarming.



Mohammadreza Chamanbaz was born in Shiraz, Iran in 1985. In 2008, he received his B.Sc. in Electrical Engineering from Shiraz University of Technology, Shiraz, Iran. In 2014, he received his Ph.D. from the Department of Electrical and Computer Engineering, National University of Singapore in control science. He was with Data Storage Institute, Singapore as research scholar from 2010 to 2014 and with Motion Energy and Control lab, Singapore University of Technol-

ogy and Design (SUTD) as postdoctoral research fellow from Sep 2014 to Sep 2015. He is now postdoctoral research fellow in Applied Complexity Group, SUTD. He is a member of IEEE Technical Committee on Computational Aspects of Control System Design (CACSD). His research activities are mainly focused on probabilistic and randomized algorithms for analysis and control of uncertain systems, convex optimization and robust control.



Roland Bouffanais is an Assistant Professor at the Singapore University of Technology and Design (SUTD). He received his Ph.D. from EPFL (Lausanne, Switzerland) in computational science for which he received the prestigious IBM Research Prize in Computational Sciences (2008) and the ERCOFTAC Da Vinci Award Silver Medal (2007). He has been a postdoctoral fellow and associate at MIT and still is a research associate with the Department of Mechanical Engineering at MIT.

Bouffanais' research group—the Applied Complexity Group—focuses on fundamental and applied interdisciplinary problems rooted in the field of complexity science. Bouffanais leads a number of active projects at SUTD related to complex networks and self-organizing systems, including swarming systems. He has recently authored a monograph titled “Design and Control of Swarm Dynamics”, published by Springer in their Complexity Series in 2016.



Louis Goh graduated with a Bachelor's degree in electrical engineering from the National University of Singapore. He is very intrigued by how things work around him and likes working on embedded system and robotics projects. As an engineer, he tries to live by the quote “How hard can it be” by Jeremy Clarkson and usually ends up regretting the decision. When not questioning why everything is not working or extinguishing magic smoke, he enjoys travelling, hiking and is

also an avid football player.



Dick K. P. Yue is the Philip J. Solondz Professor of Engineering, and Professor of Mechanical and Ocean Engineering at MIT. Yue received all his degrees (S.B., S.M. and Sc.D.) from MIT. He has been a faculty member in the MIT School of Engineering since 1983. He is active in research and teaching in wave hydrodynamics, fluid mechanics and computational methods with applications to coastal and ocean engineering. Yue is the Director of the Vortical Flow Research Laboratory,

supervising an active research group of about 20 members. His main research focus is in theoretical and computational hydrodynamics, and he is internationally recognized for his expertise on ocean and coastal wave dynamics and for his extensive work in nonlinear wave mechanics, and large-amplitude motions and loads on offshore structures. Yue has made seminal contributions in developing modern numerical methods for these problems, notably the development of the high-order spectral method for nonlinear wave-wave, wave-body, and wave-bottom interactions. Yue has also made important contributions to the understanding of hydrodynamics of fish swimming, and the complex mechanisms at the air-sea interface and their effects on interfacial processes. He has authored/co-authored some three hundred papers and a two-volume textbook on theory and applications of ocean wave hydrodynamics. Professor Yue served as the Associate Dean of Engineering at MIT from 1999 to 2007, during which time he helped create and launch the MIT OpenCourseWare (OCW) project that has transformed the global higher education landscape. Professor Yue was the Engineering School Director of International Programs 2007–2013.



Francesco Valleggra was born in Asti, Italy, in 1990. He received a B.Sc. degree in aerospace engineering from Politecnico di Torino, Turin, Italy, in 2012. In 2015 he completed the M.Sc. degree in aerospace engineering from Politecnico di Torino, Turin, Italy and Royal Institute of Technology (KTH), Stockholm, Sweden. In 2015 he joined the KTH Alfvénlaboratoriet as Research Engineer, developing star-tracker sensor and working on nanosatellite project SEAM (Small

Explorer for Advanced Missions). In fall 2015 he joined the KTH team responsible for the development of the EGSE (Electrical Ground Support Equipment) for the ESA JUICE mission. In June 2016, he started working as Research Assistant at Singapore University of Technology and Design (SUTD), Singapore, working on the development of swarming platforms for environmental sensing.