Control Engineering Practice 146 (2024) 105902

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

A multi-platform Guidance, Navigation and Control system for the autosub family of Autonomous Underwater Vehicles

Davide Fenucci *, Francesco Fanelli, Alberto Consensi, Georgios Salavasidis, Miles Pebody, Alexander B. Phillips

Marine Autonomous and Robotic Systems, National Oceanography Centre, European Way, Southampton, SO14 3ZH, UK

ARTICLE INFO

Keywords:
Autonomous Underwater Vehicle
Embedded control systems
Guidance, Navigation and Control

ABSTRACT

Over the last decade, there has been a proliferation of marine robots such as Unmanned Surface Vehicles (USVs), Autonomous Underwater Vehicles (AUVs) and underwater gliders. The variety of vehicles' capabilities and styles has led to numerous strategies to address the Guidance, Navigation and Control (GNC) tasks. In particular, a recent trend in the advancements of GNC systems is to develop modular, multi-platform solutions. This paper illustrates the approach followed in the development of the GNC system for the different types of AUV of the Autosub family. The presented system is decomposed into several interconnected components, each implemented as a standalone software module. Each component is highly-reconfigurable and easily extensible, and the interactions between the modules rely on generic-purpose interfaces. Such a design enables the execution of a broad range of autonomous missions, as well as the integration and deployment of the system on marine platforms with very diverse characteristics, not limited to the vehicles of the Autosub class. Experimental results obtained in real-time during in water campaigns are provided to demonstrate the effectiveness of the proposed system and its versatility on a wide range of platforms and missions.

1. Introduction

Autonomous Underwater Vehicles (AUVs) are now ubiquitous for marine science, industrial and military applications throughout world's ocean. Supporting such a variety of applications has driven specialisation of the vehicle designs into distinct classes of similar form and function. Each class adopts a similar approach to propulsion and manoeuvring using a combination of thrusters, control surfaces, moving masses or buoyancy engines.

Torpedo shaped, flight-style AUVs, equipped with rear propulsor(s) and control surfaces are typically used for applications where broad area surveys at high altitude are required e.g. providing detailed understanding of the seabed topography and benthic ecosystems (Wynn et al., 2014). These vehicles have typically sacrificed maneuverability in favour of increased range.

Hover-capable vehicles are fitted with multiple thrusters in different orientations to provide direct actuation of multiple degrees of freedom. These vehicles are typically designed for low speed applications and therefore tend to adopt less hydrodynamic forms often resembling Remotely Operated Vehicles (ROVs).

Underwater gliders (Rudnick, Davis, Eriksen, Fratantoni, & Perry, 2004) propel themselves by changing their buoyancy and using wings to convert some of the resulting vertical motion into forward motion, resulting in a yo-yo motion through the water. The pitch of these vehicles is typically controlled through the use of a moving mass and the heading is controlled either using a rudder or by banking the glider by generating a roll moment via a moving mass. Gliders are typically used to provide detailed information regarding the chemistry and biogeochemistry of the top 1000 m of the water column (Testor et al., 2019).

Various hybrid designs also exist, seeking to exploit the benefits of two or more of the above strategies e.g. (Caffaz, Caiti, Casalino, & Turetta, 2010; Tanakitkorn, Wilson, Turnock, & Phillips, 2017). Such a variety of form and function has led to a variety of approaches to solving the challenges of the Guidance, Navigation and Control (GNC) of AUVs. In vehicle engineering, the GNC is the part of the autonomy system responsible for the autonomous control of the movement of a vehicle in the environment. In particular, the three components of the GNC are defined as follows (Fossen, 2002):

Guidance *is the action or the system that continuously computes the reference (desired) position, velocity and acceleration of a marine craft to be used by the motion control system.*

* Corresponding author.
E-mail addresses: davfen@noc.ac.uk (D. Fenucci), franel@noc.ac.uk (F. Fanelli), alensi@noc.ac.uk (A. Consensi), geosal@noc.ac.uk (G. Salavasidis), mxp@noc.ac.uk (M. Pebody), abp@noc.ac.uk (A.B. Phillips).

https://doi.org/10.1016/j.conengprac.2024.105902
Received 31 October 2023; Received in revised form 29 February 2024; Accepted 1 March 2024
Available online 5 March 2024

Navigation *is the science of directing a craft by determining its position, attitude, course and distance travelled. In some cases velocity and acceleration are determined as well.*

Control *or more specifically motion control, is the action of determining the necessary control forces and moments to be provided by the craft in order to satisfy a certain control objective. The desired control objective is usually seen in conjunction with the guidance system. Examples of control objectives are minimum energy, set-point regulation, trajectory-tracking, path-following and maneuvering control.*

Representing a fundamental component of vehicle control systems, GNC algorithms have been developed in the fields of aerospace (*A Diverse Team*, 2016; Kendoul, 2012; Silvestrini et al., 2023), automotive (Hussain & Zeadally, 2019), and mobile robotics (Azkarate, Gerdes, Joudrier, & Pérez-del Pulgar, 2020).

In marine robotics, development of autonomy and control strategies has progressed rapidly. The severe limitations imposed by the harsh underwater environment, together with the impracticality of providing autonomous marine platforms with redundant systems to keep the costs low, pose additional challenges to ensure the safety of the platform that must be carefully addressed to enable the execution of autonomous missions. The first GNC systems relied on monolithic implementations where a single process was in charge of executing all tasks (McPhail & Pebody, 1998), or on networked infrastructures in which the computational load was distributed between different processors (Hagen, 2001). The advent of new architectural paradigms (Eickstedt & Sideleau, 2010) and software tools (Benjamin, Schmidt, Newman, & Leonard, 2010; Quigley et al., 2009) has quickly led to the concept of modular GNC systems for both Unmanned Surface Vehicles (USVs) (D'Angelo et al., 2022; Nađ, Mišković, & Mandić, 2015) and AUVs (Manhães, Scherer, Voss, Douat, & Rauschenbach, 2016; Palomeras, El-Fakdi, Carreras, & Ridao, 2012; Pinto et al., 2013; Pollini et al., 2020, 2018). In such systems, responsibilities are split across different components that interact to provide the full set of functionalities to the platform. Modularity yields flexibility to the overall system, and allows for a better management of the interactions between the different components: any module can be replaced with a different implementation, as long as this complies with the defined interfaces. The possibility of leveraging these features, together with the emerging desire of supporting interoperability during marine operations (Costanzi et al., 2020), motivated the need to design control software architectures that could be easily adapted to multiple autonomous robotic platforms with diverse actuation, sensing and hydrodynamic characteristics. Many of the previous systems have been successfully deployed on different AUVs and demonstrated in water. For instance, the COLA2 system presented in Palomeras et al. (2012) provides the core functionalities of both the vehicles Girona 500 (Ribas, Palomeras, Ridao, Carreras, & Mallios, 2012) and Sparus II (Carreras et al., 2018), vehicles with very different thruster configurations.

1.1. Contribution of this paper

Inspired by this on-going trend, this paper presents the design of a GNC system explicitly targeted to multiple platform types belonging to the Autosub family of AUVs. The proposed system is integrated into the on-board control and autonomy system (OCS) developed by the Marine Autonomous Robotic Systems (MARS) group of the National Oceanographic Centre (Munafo et al., 2019). The OCS is based on the modular middleware Robot Operating System (ROS), and it is developed to address the peculiar set of requirements defined for the platforms of the Autosub family that makes the adaptation of available state-of-the-art software solutions difficult without considerable efforts. In particular, the main challenges that the OCS and the proposed GNC system try to overcome are:

- interact with different hardware components (such as electronics, sensors and actuators), both off-the-shelf and developed in-house, for whom embedded software modules are not available;
- be able to run on low-power hardware with limited computational resources, to enable the execution of long-term (i.e. over a month) continuous deployments, including under-ice operations;
- command a wide range of effectors commonly used in marine robotic platforms, such as propellers/thrusters and control surfaces, mounted on board with different configurations;
- allow the extension of the base set of GNC algorithms to provide the platforms with more advanced capabilities, e.g. integrating a third-party backseat system;
- assess the health of the platform during a mission, detecting and identifying faults and unexpected situations that may occur;
- react to potential actuation faults to keep the mission progressing, when redundancy allows it;
- provide direct access to actuators by-passing the control chain, in case of an unrecoverable emergency (e.g. depth sensor fault).

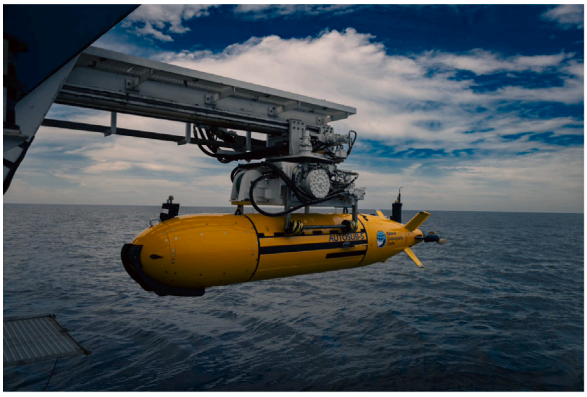
Having built the OCS on ROS, the structure of the GNC can be decomposed into different independent components interacting each other and with the other modules of the OCS. The interaction between the components relies on generic-purpose interfaces (i.e. ROS messages) that apply to a broad range of platform types and applications. This provides the GNC system architecture with a stable, unified infrastructure that can be deployed on all the different Autosub platform types, with associated benefits in terms of code maintenance and re-usability. At the same time, each module of the GNC system is designed to be highly configurable, allowing the underlying algorithms to be easily customised, or even replaced with more advanced/application-tailored implementations. Leveraging such a design, the addition of new capabilities focuses on the development of control/autonomy algorithms, rather than on their integration with the rest of the on-board software. Although the default implementation of the GNC algorithms presented in this paper is well established in the state of the art, the design of the system concept, the interfaces and interactions between its components are originally conceived to enable the execution of different types of missions with different types of AUV.

The intent of this paper is to serve as a guideline towards the development of a multi-platform GNC system, illustrating the motivations, and demonstrating the effectiveness of the proposed architecture with real-time results from experimental campaigns at sea. Finally, it is worth remarking that, despite being initially tailored to the various vehicles of the Autosub class, the design of the presented architecture is generic and can be integrated and deployed on other type of platforms.

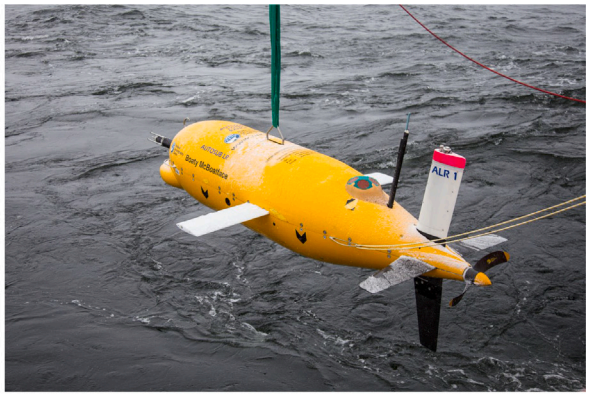
The remainder of this paper is organised as follows: Section 2 describes the current generation of Autosub AUVs, Section 3 articulates the notation used throughout this paper, Section 4 describes the control system architecture, Section 5 presents real world results and Section 6 summarises the findings from this work.

2. Autosub AUV platforms

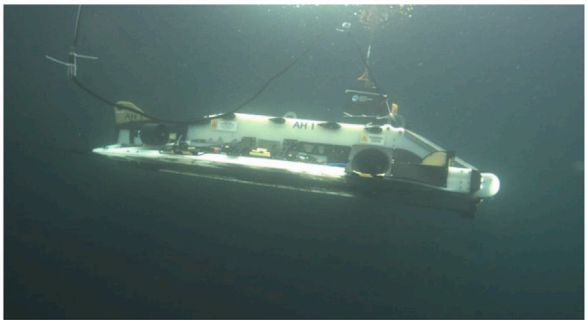
The Autosub AUV programme has been developing large work class AUVs to support oceanographic science applications since the 1980s, for a more complete history see Griffiths (2012). Key achievements include: Autosub 3 successfully exploring under the Pine Island Glacier in 2009 (McPhail et al., 2009) providing unprecedented multi-beam imagery of the underside of the ice (Graham et al., 2013; Jenkins et al., 2010) and Autosub6000 (McPhail, 2009) locating the deepest known hydro-thermal vent at nearly 5000 m in the Mid-Cayman Rise (Connelly et al., 2012). More recently the Autosub team have diversified the range of AUV types being developed and operated to include ultra long range AUVs (Alex et al., 2023; Roper et al., 2021) and a hover-capable AUV. This move to more diverse platform types drove a requirement for a generic multi-vehicle software architecture for both onboard control (Munafo et al., 2019) and piloting (Harris et al., 2020). The current portfolio of Autosub vehicles are described in the following subsections. Table 1 reports the specifications of each Autosub.



(a) Autosub5.



(b) Autosub Long Range (ALR).



(c) Autosub Hover (AH).

Fig. 1. Autosub AUVs.

Table 1
Specifications of the vehicles of the Autosub family.

Vehicle	Sensors	Actuators
ALR	PNI TCM XB compass Teledyne RDI 300kHz ADCP Sea-Bird CTD SBE 52-MP uBLOX M8 GPS	1x rear propeller 1x rudder 1x sternplanes
Autosub5	Sonardyne SPRINT-Nav 700 Sea-Bird CTD SBE 9+ uBLOX M8 GPS	2x rear propeller 4x control surfaces
AH	Sonardyne SPRINT-Nav Mini uBLOX M8 GPS	6x Tecnadyne thruster 540

2.1. Autosub5

Autosub5 (see Fig. 1(a)) is the latest deep-rated work-class Autosub vehicle, developed for high power acoustic and optical sensing in deep ocean and under ice, Consensi et al. (2022) and Phillips, Kingsland, Linton, Baker, Bowring, Soper, Roper, Johnson, Morrison, Ciaramella, et al. (2020). The AUV is 5.5 m long, has a diameter of 0.9 m, displaces 2.5 m^3 and can be depth rated to 2000 m or 6000 m depending on the application. The vehicle is equipped with two thrusters at the stern and four independently movable control planes in a 'x' configuration, providing redundancy to single failures. For primary navigation and orientation Autosub5 is equipped with a Sonardyne SprintNAV Doppler Velocity Log (DVL)-aided Inertial Navigation System (INS)/Attitude and Heading Reference System (AHRS). The vehicle is also fitted with a Sonardyne 6G acoustic modem, so it can be fed with Ultra-Short Base Line (USBL) position fixes from a support vessel when available. For shallow water near bottom operations, or operations in deeper water with robust USBL positioning the navigation from the SprintNAV is used directly. For very deep water operations the DVL and INS are used to feed a robust dead-reckoning algorithm.

2.2. ALR

The National Oceanography Centre (NOC) currently operate six Autosub Long Range (ALR) vehicles (see Fig. 1(b)), three 1500 m depth rated and three 6000 m depth rated, Alex et al. (2023) and Roper et al. (2021). All the vehicles are 3.6 m long, 0.9 m in diameter and displace 1.2 m^3 in water. Each vehicle is equipped with a single thruster at the stern and three movable control surfaces, one rudder and two jointly actuated sternplanes; completing the aft arrangement is a fixed vertical fin which houses secondary relocation beacons. For primary navigation the vehicles are equipped with a down DVL (Nortek 500 kHz or RDI Workhorse 300 kHz), a Micro-Electro-Mechanical System (MEMS) compass and orientation sensor (PNI) and an acoustic modem for acoustic-aided navigation.

2.3. AH

Autosub Hover (AH) (see Fig. 1(c)) is a prototype vehicle developed to overcome the challenge of operating near infrastructures and rough terrain. High-manoeuvrability and enhanced actuation capabilities are

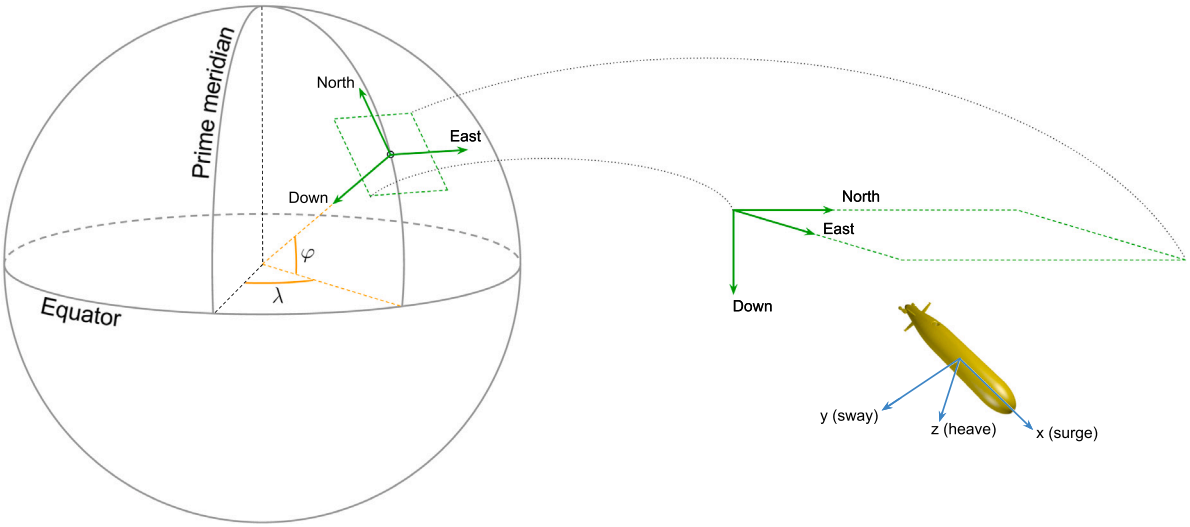


Fig. 2. Definition of notable reference frames: body-fixed b (blue), navigation n (green), geodetic g (orange). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

key aspects in such scenarios; the vehicle is therefore equipped with six tunnel ducted thrusters: three vertical in a triangle-shaped configuration, one lateral on the bow side, and two forward. Main navigation is provided by Sonardyne SprintNAV Mini DVL-aided INS/AHRS.

3. Notation

Throughout this paper the following notation will be used. Scalar quantities are indicated with letters in plain math style. Vectors and matrices are indicated with lowercase and capital letters, respectively, both in bold math style; unless otherwise stated, vectors are assumed to be column vectors. The unit vector specifying the direction of an axis w is indicated with the symbol $\overline{\mathbf{w}}$. The cross product operator between two vectors is indicated with the symbol '×'. Reference frames are indicated with lowercase letters enclosed within curly braces (e.g. $\{a\}$). Notable reference frames that are used in the following are:

- the *body-fixed* frame $\{b\}$, with the axes x, y and z defined according to the SNAME (1950) convention shown in Fig. 2;
- the *navigation* frame $\{n\}$, defined on the local tangent plane according to the North-East-Down (NED) convention.
- the *geodetic* coordinates system $\{g\}$, identifying a point on the Earth using latitude φ, longitude λ and depth d. As a special case, the reduced coordinate system that does not consider the depth is indicated with $\{g_2\}$.

When required, the reference system a vector $\mathbf{x}$ is expressed in is indicated adding a top-right superscript to the quantity's symbol, e.g. $\mathbf{x}^a$. The coordinate transformation matrix from a reference system $\{r_1\}$ to a reference system $\{r_2\}$ is indicated as ${}^{r_2}\mathbf{R}_{r_1}$. Finally, the estimate of a real quantity x is indicated as $\hat{x}$.

4. Guidance, Navigation and Controlsystem architecture

Fig. 3 illustrates the architecture of the GNC in the On-board Control System (OCS), with its components highlighted in blue. The inputs to the GNC system are decomposed into the mission plan (sequence of tasks) provided by the pilot and the measurements of the navigation sensors installed on the vehicle.

In the OCS domain, a mission is defined as a sequence of tasks, called *behaviours*, that are forwarded sequentially, one at a time, to the GNC system. Each behaviour includes:

- a *goal*, whose definition is specific for each behaviour, but in general it is composed of:
  - a *track* on the horizontal plane, i.e. a straight line defined between a start waypoint S and a destination waypoint D. Note that in case S and D are coincident, the track collapses into a single waypoint.
  - a desired *target depth* d_{des}, valid for the entire track;
  - a desired *cruise speed* s_{cr}, i.e. the desired magnitude of the velocity vector at steady-state;
  - a desired *heading* ψ_{des} (optional) for the vehicle at steady-state.
- a *set of constraints* associated with the mission goal that the vehicle needs to meet during the whole execution of the behaviour. This typically includes a timeout for the behaviour, desired tolerances for the position/depth, and desired limits for certain quantities determined at lower levels (e.g. min./max. pitch angle, min. altitude from the sea bottom, etc.).

The scheduling of the behaviours is entrusted to the mission executive, also referred to as the *helm* (Sprague et al., 2018), which evaluates the completion condition for each behaviour based on the given thresholds. In particular, a behaviour is considered to be completed according to the following logic:

$$\text{timeout expired } \textit{or } (\text{depth achieved } \textit{and } \text{track completed}), \tag{1}$$

where the "depth achieved" condition is met when the *depth error* $\tilde{d}$ is below a given depth tolerance d_{tol}:

$$\text{depth achieved} \iff \|\tilde{d}\| \leq d_{\text{tol}}, \tag{2}$$

whereas the "track completed" conditions is verified when the distance between the final destination waypoint and the current location of the vehicle, also referred to as *range to go* r_{togo}, is within a circle of acceptance with radius R_{tol}:

$$\text{track completed} \iff r_{\text{togo}} \leq R_{\text{tol}}. \tag{3}$$

Moreover, the helm is in charge of performing basic health checks at each time step to determine whether the mission can proceed, or an emergency countermeasure needs to be taken to react to an unexpected situation.

4.1. Navigation system

The OCS navigation system provides an estimation of the vehicle's:

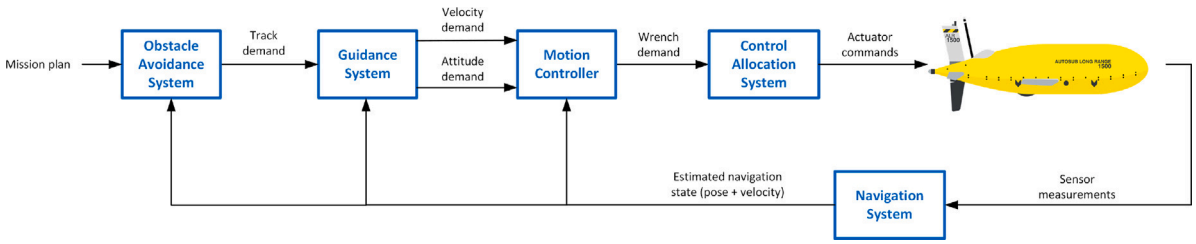


Fig. 3. Schematic architecture of the GNC system for the Autosub family. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

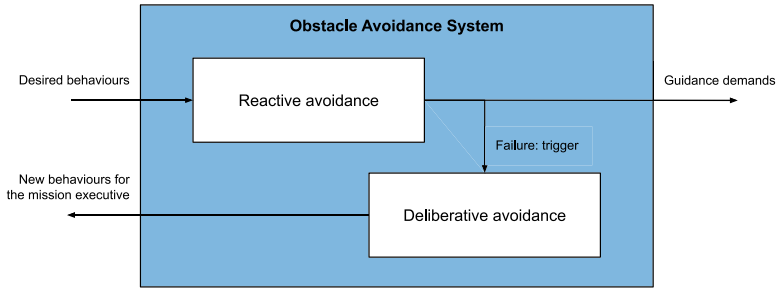


Fig. 4. Architecture of the obstacle avoidance system.

- geodetic position in latitude, longitude, depth (LLD) coordinates $\{g\}$: $\hat{\mathbf{p}}^g = [\hat{\varphi},\ \hat{\lambda},\ \hat{d}]^T$;
- orientation of the body-fixed frame b with respect to the navigation frame n, parameterised by the Euler angles: $\hat{\boldsymbol{\Theta}} = [\hat{\phi},\ \hat{\theta},\ \hat{\psi}]^T$;
- linear velocity in the body-fixed frame $\{b\}$: $\hat{\mathbf{v}}^b = [\hat{u},\ \hat{v},\ \hat{w}]^T$.

To deal with the variety of aiding sensors installed on-board the platforms of the Autosub family, the navigation solution is determined using a hierarchical approach: in case an estimate of the navigational state is available from high-precision INSs such as the SprintNAV integrated systems, it is used as the navigation solution for the vehicle. Otherwise, the calculation of the navigation solution falls back to a classic dead-reckoning algorithm integrating the raw measurements coming from the navigation sensors installed on-board.

The dead-reckoner uses the orientation estimate provided by the AHRS / PNI compass; the remaining components of the navigation solution are determined through a two-step process. First, an estimation of the body velocity $\hat{\mathbf{v}}^b$ is obtained from the measurements of the downwards-facing DVL, when available, or from a steady-state model of the vehicle's dynamics otherwise. Then, the current vehicle's position is calculated from the previous one, adding the displacement given by the estimated velocity over the last time interval dt:

$$\hat{\mathbf{p}}^n(t+1)_{\mathrm{dr}} = \hat{\mathbf{p}}^n(t) + {}^n\mathbf{R}_b(\hat{\boldsymbol{\Theta}}(t))\hat{\mathbf{v}}^b(t)\,dt. \tag{4}$$

The vehicle's position can be further refined when a Global Positioning System (GPS)/USBL fix becomes available: in this case, the dead-reckoned estimation is fused together with the position measurement via a complementary filter:

$$\hat{\mathbf{p}}^n(t) = \alpha\hat{\mathbf{p}}^n(t)_{\mathrm{dr}} + (1-\alpha)\mathbf{p}^n_{\mathrm{gps/usbl}}, \tag{5}$$

where α is the filter gain.

4.2. Obstacle Avoidance System

The OCS Obstacle Avoidance System (OAS) (Fanelli, Fenucci, Marlow, Pebody, & Phillips, 2020) receives the goals and constraints from the helm and verifies if these are compatible with the safety of the vehicle, before using them to generate inputs to the guidance system. Fig. 4 shows the architecture of the avoidance system. The OAS is structured so that it allows for the simultaneous presence of *reactive* avoidance strategies, i.e. instantaneous countermeasures that alter the received helm inputs to produce the safest possible ones closest to those desired, and *deliberative* strategies, that work on longer spatial and temporal horizons to produce new paths for the vehicle. Note that as long as the helm produces an output, reactive strategies are always active and generate inputs to the guidance system (possibly signalling failure if the OAS deems the vehicle to be unsafe), while deliberative strategies only act in case failure is signalled by reactive strategies. Furthermore, the OAS not being a scheduler – this role is exclusively undertaken by the helm – the generated paths are not commanded directly to the guidance system, but are fed back to the helm who will ultimately decide whether to follow the "recommendation" of the avoidance system (any new path will follow the same route through the OAS and the rest of the guidance and control loop).

Different avoidance strategies (reactive and deliberative) can be developed and plugged in the system. By default, the system uses only a reactive strategy to ensure that the vehicle always maintains a minimum required *altitude*, i.e. the (vertical) distance from the sea-bottom, during the mission. Altitude control is then converted into a depth control problem: the depth set-point d_{des} received from the mission executive is saturated so that it never exceeds the depth obtained by subtracting the desired altitude h_{des} from the sea-bottom depth. The latter is calculated as the sum of the current vehicle's depth and the altitude $\hat{h}$ estimated using the measurements of the downwards-looking DVL, when they are available:

$$d_{\mathrm{des}} = \underbrace{\hat{d} + \hat{h}}_{\text{sea-bottom depth}} - h_{\mathrm{des}} \tag{6}$$

4.3. Guidance system

In the OCS, the guidance is the part of the GNC system that provides linear velocity and attitude set-points to the motion controller given a desired behaviour, and an estimate of the current vehicle's state from the navigation system.

To simplify the problem of the guidance system in the six degrees of freedom (DOFs), the movement of the vehicle is decoupled on three different planes, each one associated to a dedicated guidance component: frontal, vertical, horizontal (Fig. 5). Each component supports different working modalities, or *modes*; a triplet of modes $\{m_{\mathrm{front}},\ m_{\mathrm{horz}},\ m_{\mathrm{vert}}\}$

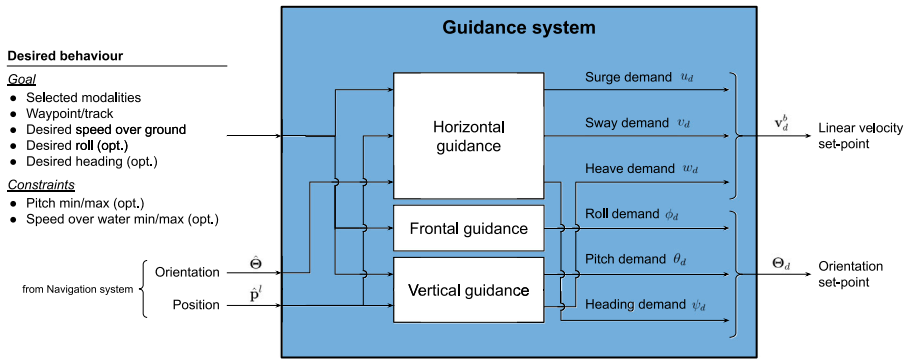
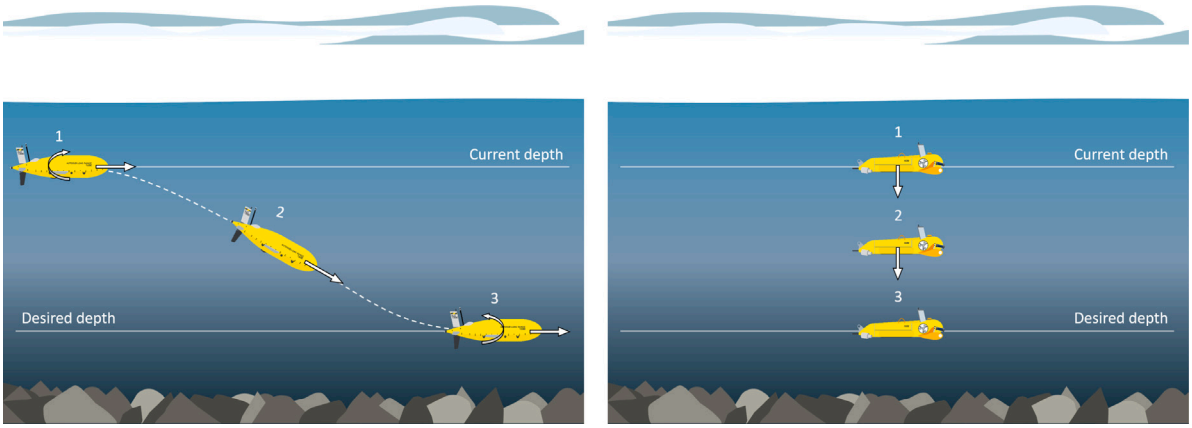


Fig. 5. Architecture of the guidance module.



(a) Depth control for flight-style vehicles. When the vehicle is commanded to change its depth (1), it performs an ascent/descent profile by changing its pitch angle while propelling (2), until it reaches the desired depth (3).

(b) Depth control for hover-capable vehicles. When the vehicle is commanded to change its depth (1), it performs a vertical (levelled) descent/ascent using the vertical thrusters (2) until it reaches the desired depth (3).

Fig. 6. Guidance modes for the vertical component.

unequivocally identifies a particular behaviour. All the components support a "disabled" mode that sets the corresponding outputs as "inactive", causing them to be ignored in the motion controller.

4.3.1. Frontal plane

The frontal guidance component is defined in the vehicle-fixed frame and deals with the roll DOF. In particular, the only active mode supported simply forwards the desired roll angle ϕ_{des} to the motion controller as the roll set-point ϕ_d. In practice, the roll set-point is typically set to zero.

4.3.2. Vertical plane

The vertical guidance component controls the motion of the vehicle on the lateral vertical plane, i.e. the plane perpendicular to the vehicle's sway axis. The mission goal for this component is represented by the desired target depth, that can be achieved essentially using two alternative strategies.

Regardless of the selected working mode, the vertical guidance module evaluates the depth error as the difference between the depth set-point and the current depth estimated by the navigation system: $\tilde{d} = d_{\text{des}} - \hat{d}$. The depth error is then communicated to the mission executive, where it is used to determine the "depth achieved" condition (2).

Depth control for flight-style vehicles. The first strategy to drive a vehicle to a desired depth is to change its inclination with respect to the horizontal line as it moves in the longitudinal direction, describing a sigmoid-shaped profile (if going in a straight line), or a spiral (if circling around a point). For small roll angles, the inclination can be approximated with the vehicle's pitch angle. This strategy is typically adopted by flight-style AUVs (whence the name) such as Autosub5 and ALR, where diving planes are actuated so that the relative motion between the vehicle and the body of water generates a lift force, resulting in a pitching moment. In addition, pitch angle variation can be produced using other actuation mechanisms, such as moving mass or vertical thruster(s), making this strategy feasible for other types of vehicles. Fig. 6(a) illustrates the described approach.

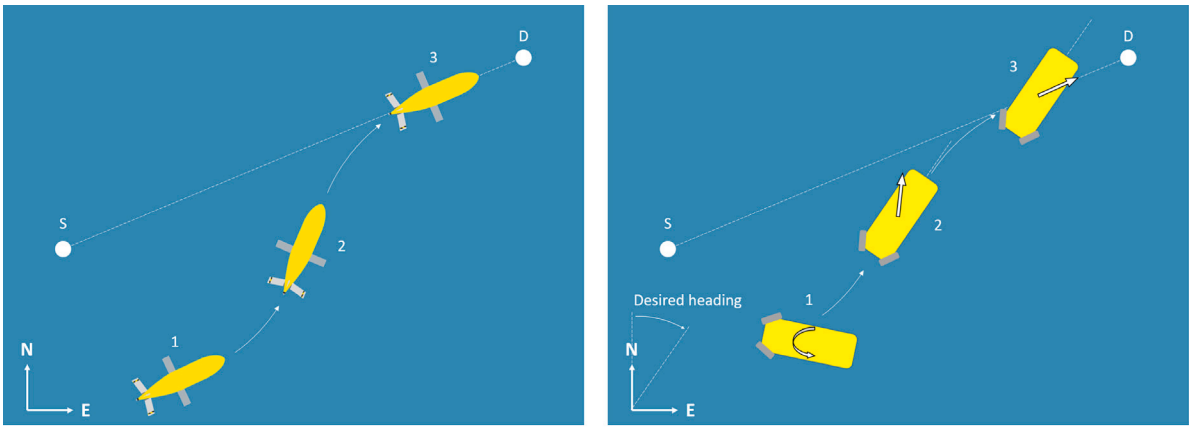
Flight-style depth control is implemented with a PID controller taking as input the depth error, and giving as output the unconstrained pitch demand. The output of the PID is then constrained between the pitch limits θ_{min} and θ_{max} given as input to obtain the pitch demand θ_d for the motion controller. Finally, the heave velocity demand w_d is set to be inactive.

Depth control for hover-capable vehicles. An alternative strategy to change the vehicle's depth is to execute vertical ascents/descents. This typically requires that the vehicle is capable of producing a heave force while keeping the body levelled (i.e. with roll and pitch angles set to zero), e.g. employing one or more vertical thrusters. Fig. 6(b) illustrates the described approach.

Similarly to its flight-style counterpart, depth control for hover-capable vehicles is implemented with a PID controller taking as input the depth error and giving as output the unconstrained heave demand. The output of the PID is then constrained between the configured heave limits w_{min} and w_{max} to obtain the heave demand w_d for the motion controller. Finally, the pitch angle demand θ_d is always set to zero.

4.3.3. Horizontal plane

The horizontal guidance component controls the motion of the vehicle on the local horizontal tangent plane according to the requested



(a) Track-following. When enabled (1), the vehicle heads towards the track defined between the start waypoint S and the destination waypoint D (2). Once the vehicle joins the track, it follows the straight line until the destination waypoint is reached (3).

(b) Track-following strafe. When enabled (1), the vehicle moves to reach the desired heading while approaching the track (2). Once the vehicle has joint the track, it proceeds towards the end waypoint along the straight line maintaining a slant heading (3).

Fig. 7. Guidance modes for the horizontal component.

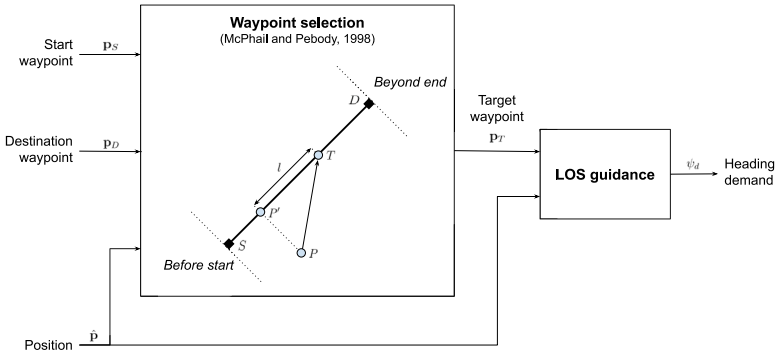


Fig. 8. Block diagram of the track-following mode.

modality, providing set-points for the heading angle ψ_d, the surge velocity u_d and/or the sway velocity v_d.

The mission goal in the horizontal plane is represented by the track, the desired cruise speed, and the desired heading, when required. The track is specified by the start and destination waypoints S and D, whose positions $\mathbf{p}_S = [n_S, e_S]^T$ and $\mathbf{p}_D = [n_D, e_D]^T$ are defined on the local tangent plane in North-East coordinates. In the following, it is assumed that the current vehicle's position estimate is transformed into the same coordinate system, consistently with the track definition:

$$\hat{\mathbf{p}}^{g_2} \rightarrow \hat{\mathbf{p}} = [\hat{n}, \hat{e}]^T. \tag{7}$$

Regardless of the selected working mode, the horizontal guidance module evaluates the range to go $r_{\text{togo}} = \|\mathbf{p}_D - \hat{\mathbf{p}}\|$, which is communicated to the helm and used to determine the "track completed" condition (3).

Track-following. Track-following is used in applications that require for the vehicle to follow a track, e.g. side-scan sonar surveys. In this case, a simple line of sight (LOS) guidance does not ensure that the desired path is precisely followed: in presence of disturbances that push the vehicle off the track (e.g. lateral currents), the LOS guidance does not attempt to rejoin this line, as it aims for the destination waypoint only. On the other hand, in track-following the vehicle first heads towards the track defined between the start waypoint S and the destination waypoint D. Once the vehicle joins the track, it follows the straight line until the destination waypoint is reached (Fig. 7(a)).

Originally presented in McPhail and Pebody (1998), the track-following mode prefixes the LOS guidance with a waypoint selection algorithm that chooses the target position demand such that the vehicle follows the straight line track defined between the two waypoints, as shown in Fig. 8. Given the start waypoint position $\mathbf{p}_S$, the destination waypoint position $\mathbf{p}_D$, and the current vehicle's location $\mathbf{p}_P = \hat{\mathbf{p}}$, the position of the target waypoint T is selected projecting the vehicle's position on the track line (point P'), and proceeding a look-ahead distance l towards the destination waypoint D:

$$\mathbf{p}_T = \mathbf{p}_S + \bar{\mathbf{p}}_{SD}\left(l + \frac{\mathbf{p}_{SD}^T \mathbf{p}_{SP}}{\|\mathbf{p}_{SD}\|}\right) \tag{8}$$

where $\mathbf{p}_{XY} = \mathbf{p}_Y - \mathbf{p}_X$ indicates the relative position of point Y with respect to point X, and $\bar{\mathbf{p}}_{SD}$ therefore represents the unit vector aligned with the track line (*along-track* direction). Moreover, an optional correction term, proportional to the integral of the *cross-track* error, can be added to the target position in order to counteract the drift caused by any potential lateral current:

$$\mathbf{p}_{T'}(t) = \mathbf{p}_T(t) + k_{I_{\text{CT}}} \int_0^t \mathbf{p}_{PP'}(\tau)\, \mathrm{d}\tau. \tag{9}$$

It is worth remarking that the cross-track error $\mathbf{p}_{PP'}$ is perpendicular to the track at every time instant: the corrected target waypoint T' is therefore obtained by shifting the original target waypoint T off the line in the cross-track direction. To prevent undesirable effects due to the integrator windup (Åström & Hägglund, 2006), the integral term is saturated such that its norm never exceeds a given threshold $T_{I_{\text{CT}}}$. Finally, a consistency check is carried out; if the waypoint T' lies before the start of the track/beyond the end of the track, the selected target position is set to the start/end of the track:

$$\mathbf{p}_{T''} = \begin{cases} \mathbf{p}_S & \text{if } \mathbf{p}_{SD}^T \mathbf{p}_{ST'} < 0 \text{ (target lies before the start)}, \\ \mathbf{p}_D & \text{if } \mathbf{p}_{SD}^T \mathbf{p}_{DT'} > 0 \text{ (target lies beyond the end)}, \\ \mathbf{p}_{T'} & \text{otherwise.} \end{cases} \tag{10}$$

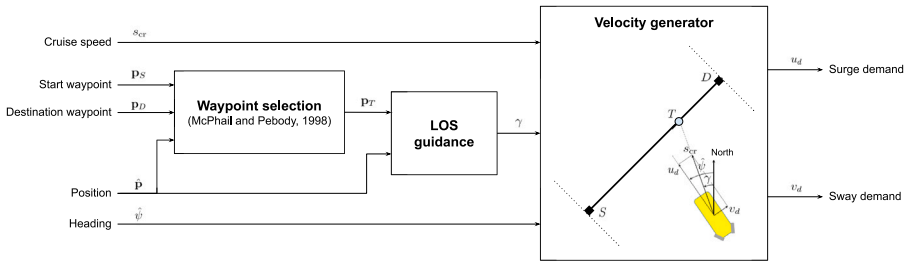


Fig. 9. Block diagram of the track-following strafe mode. The surge and sway demands are determined using the geometric construction illustrated inside the velocity generator.

Once the target position has been determined, the LOS guidance block computes the heading demand ψ_d for the vehicle. Furthermore, the desired cruise speed is used as the surge set-point u_d for the motion controller, whereas the sway demand is set to be inactive:

$$u_d = s_{\text{cr}}, \quad \psi_d = \text{atan2}(e_{PT''}, n_{PT''}), \tag{11}$$

where $n_{PT''}$ and $e_{PT''}$ are the North and East coordinate, respectively, of the target's position with respect to the vehicle.

Finally, it is worth remarking that in case the track collapses into a single point, the track-following mode falls back to the classic LOS guidance as a consequence of (10).

Track-following strafe. Track-following strafe mode is used in applications where the vehicle is required to follow a straight line while maintaining a given heading, e.g. camera surveys of a wall/vertical structure, and it is hence designed for highly-manoeuvrable robots such as AH-1. In this guidance mode, the vehicle uses horizontal thrusters to achieve the desired heading while approaching the track. Once the vehicle has joint the track, it proceeds towards the end waypoint along the straight line maintaining a slant heading, as shown in Fig. 7(b).

This guidance mode runs an instance of the track-following algorithm described above. Rather than being used to set the heading demand for the motion controller, the output of the track-following is instead considered as the angle γ between the velocity vector and the direction of the North. The angle γ is then given as input to a velocity generator block, together with the desired cruise speed and the current heading estimate, to determine the surge and sway demands under the assumption that the vehicle is levelled, as shown in Fig. 9. Finally, the heading set-point for the controller is set to the given desired heading.

The output of the track-following strafe guidance hence results:

$$u_d = s_{\text{cr}} \cos(\gamma - \hat{\psi}), \quad v_d = s_{\text{cr}} \sin(\gamma - \hat{\psi}), \quad \psi_d = \psi_{\text{des}}. \tag{12}$$

Note that in the OCS behaviours this guidance mode is always used in combination with depth control for hover-capable platforms, which outputs a zero pitch demand. This means that during the execution of the behaviour, the vehicle will effectively be levelled, hence verifying the above assumption.

4.4. Motion controller

The motion controller is the part of the control chain that evaluates the virtual forces and torques, $\mathbf{f}_d^b$ and $\mathbf{m}_d^b$, (i.e. the *virtual wrench* $\boldsymbol{\tau}_d^b = [\mathbf{f}_d^b, \mathbf{m}_d^b]^T$) to be applied on the vehicle's body in order to achieve the desired linear velocity and orientation set-points, $\mathbf{v}_d^b$ and $\boldsymbol{\Theta}_d$, commanded by the guidance system.

The motion controller is composed of two decoupled 3-DOFs feedback regulators executing in parallel to achieve the demanded linear velocity and orientation, respectively, as shown in Fig. 10.

Both controllers have a generic structure that allows any feasible control law to be developed and plugged-in, replacing the default implementations. In particular, a feasible control law for the linear velocity (orientation) is any control law that takes at least the linear velocity error $\tilde{\mathbf{v}}^b = \mathbf{v}_d^b - \hat{\mathbf{v}}^b$ (orientation error $\tilde{\boldsymbol{\Theta}} = \boldsymbol{\Theta}_d - \hat{\boldsymbol{\Theta}}$) as input, and gives a force demand $\mathbf{f}_d^b$ (moment demand $\mathbf{m}_d^b$) as an output.

The default implementation of the linear velocity regulator is represented by a classic 3D-PID control law:

$$\mathbf{f}_d^b(t) = \mathbf{K}_{p,v}\tilde{\mathbf{v}}^b(t) + \mathbf{K}_{i,v}\int_0^t \tilde{\mathbf{v}}^b(\tau)\,\mathrm{d}\tau + \mathbf{K}_{d,v}\dot{\tilde{\mathbf{v}}}^b(t), \tag{13}$$

where $\mathbf{K}_{p,v}, \mathbf{K}_{i,v}, \mathbf{K}_{d,v} \in \mathbb{R}^{3\times3}$ are the proportional, integral, and derivative gain matrices, respectively. Similarly, the default orientation controller evaluates the moment demand as a result of a 3D-PID control law whose output is transformed back to the body-fixed frame through the transpose of the Jacobian $\mathbf{J}_2(\boldsymbol{\Theta})$:

$$\mathbf{m}_d^b(t) = \mathbf{J}_2^T(\boldsymbol{\Theta}(t))\left(\mathbf{K}_{p,\Theta}\tilde{\boldsymbol{\Theta}}(t) + \mathbf{K}_{i,\Theta}\int_0^t \tilde{\boldsymbol{\Theta}}(\tau)\,\mathrm{d}\tau + \mathbf{K}_{d,\Theta}\dot{\tilde{\boldsymbol{\Theta}}}(t)\right), \tag{14}$$

where $\mathbf{K}_{p,\Theta}, \mathbf{K}_{i,\Theta}, \mathbf{K}_{d,\Theta} \in \mathbb{R}^{3\times3}$ are the proportional, integral, and derivative gain matrices, respectively (Antonelli, 2006; Fossen, 2002).

Both default controllers (and, more in general, the OCS implementation of the integral contribution) include an anti-windup strategy based on the element-wise clamping of the integral error.

4.5. Control allocation system

The control allocation system aims at determining a feasible vector of input commands $\mathbf{u} = [u_1, \ldots, u_n]^T$ for the n actuators mounted on a vehicle, such that the joint action produced by the corresponding effectors is as close as possible to the wrench demand generated by the vehicle's motion controller.

The OCS control allocation system relies on a generic, fully configurable structure that allows the system to be adapted to the diverse characteristics of the Autosub platforms (see Fig. 11). Different actuator/effector models are supported; currently implemented models include thrusters and control surfaces, but additional ones can be defined and plugged-in. Each effector i is assumed to provide a force $\mathbf{f}_i^b \in \mathbb{R}^3$. For instance, a propeller/thruster produces a force directed along its rotational axis, whereas for control surfaces the lift force only is considered. For the purposes of the control allocation system, it is convenient to decompose the force given by the ith effector as:

$$\mathbf{f}_i^b = f_i\,\bar{\mathbf{f}}_i^b, \quad f_i = \begin{cases} \|\mathbf{f}_i^b\| & \text{if } (\mathbf{f}_i^b)^T\bar{\mathbf{f}}_i^b > 0 \\ -\|\mathbf{f}_i^b\| & \text{otherwise.,} \end{cases} \tag{15}$$

where $\bar{\mathbf{f}}_i^b \in \mathbb{R}^3$ indicates the positive direction of the force provided by the effector, and $f_i \in \mathbb{R}$ is a scalar whose absolute value is equal to the norm of the force, positive if $\mathbf{f}_i^b$ is oriented as $\bar{\mathbf{f}}_i^b$ and negative otherwise. The force intensity f_i can be modelled, in general, as a non-linear function $h_i(\cdot)$, assumed to be invertible, of the actuator's input command u_i:

$$f_i = h_i(u_i). \tag{16}$$

For thrusters and control surfaces the input command u_i is represented by the rotational speed and the hydrofoil deflection angle, respectively; models implemented in the OCS for these effectors include the ones available in the UUV simulator package (Manhães et al., 2016). Defining the position of the effector with respect to the vehicle's

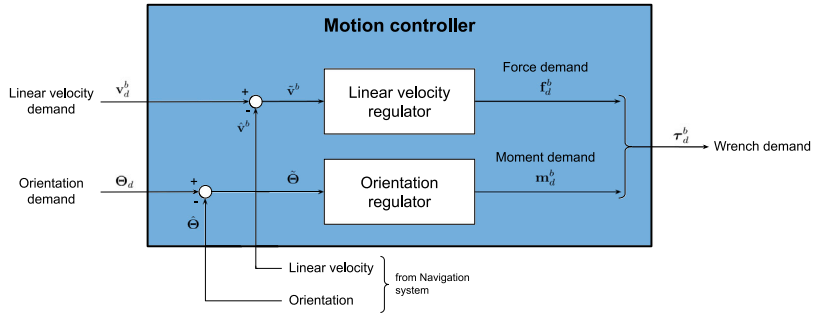


Fig. 10. Architecture of the motion controller module.

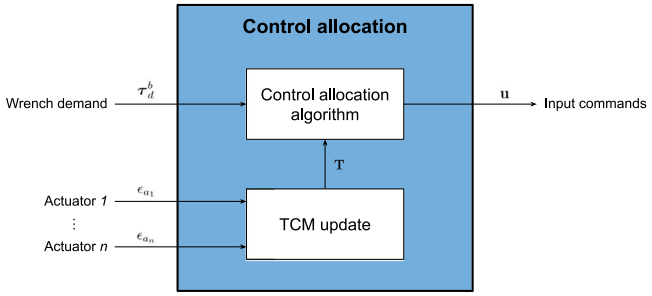


Fig. 11. Architecture of the control allocation module.

centre of gravity (COG) as $\mathbf{r}_i^b$, the wrench exerted on the vehicle's body by the ith effector can be expressed as:

$$\boldsymbol{\tau}_i^b = \begin{bmatrix} \mathbf{f}_i^b \\ \mathbf{r}_i^b \times \mathbf{f}_i^b \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{f}}_i^b \\ \mathbf{r}_i^b \times \bar{\mathbf{f}}_i^b \end{bmatrix} h_i(u_i) = \mathbf{t}_i^b h_i(u_i). \tag{17}$$

The joint action produced on the vehicle's body by the n effectors is described by summing up all the individual contributions; in matrix form this is expressed as follows:

$$\boldsymbol{\tau}^b = \begin{bmatrix} | & & | \\ \mathbf{t}_1^b & \cdots & \mathbf{t}_n^b \\ | & & | \end{bmatrix} \begin{bmatrix} h_1(u_1) \\ \vdots \\ h_n(u_n) \end{bmatrix} = \mathbf{T}\,\mathbf{h}(\mathbf{u}), \tag{18}$$

where the matrix $\mathbf{T} \in \mathbb{R}^{6\times n}$ is referred to as the *Thrust Configuration Matrix (TCM)*. Within the OCS, the TCM is built dynamically, based on the actuators' current availability $\epsilon_{a_1}, \ldots, \epsilon_{a_n} \in \{0, 1\}$, where 1 means "available". In particular, only the columns $\mathbf{t}_i^b$ and the inputs u_i corresponding to the actuators whose indicator ϵ_{a_i} are equal to 1 are selected to build the TCM and the input vector, respectively. The availability of an actuator is updated in response to unexpected situations identified by the actuator's internal health system, for instance in presence of overcurrents or faults.

The aim of the control allocation system is to find a vector $\mathbf{u}^*$ such that the resulting wrench (18) equals the desired wrench $\boldsymbol{\tau}_d^b$. Depending on the characteristics of the TCM, this problem has different solution sets:

(i) $\text{rank}(\mathbf{T}) = 6$, $n = 6$ (TCM is a full rank, square matrix): the control allocation problem has a unique solution $\mathbf{u}^* = \mathbf{h}^{-1}\left(\mathbf{T}^{-1}\boldsymbol{\tau}_d^b\right)$ and the vehicle is said to be *fully-actuated*;

(ii) $\text{rank}(\mathbf{T}) = 6$, $n > 6$ (TCM is a full row rank, non-square matrix): the control allocation problem admits infinitely many solutions, and the vehicle is said to be *over-actuated*. Redundancy can be exploited to introduce an additional secondary objective to the problem, e.g. minimise the power consumption;

(iii) $\text{rank}(\mathbf{T}) < 6$ (TCM is not full row rank; a trivial case that verifies this condition is $n < 6$): no wrench can be generated in the subspace of $\mathbb{R}^6$ corresponding to $\ker(\mathbf{T}^T)$, and the vehicle is said to be *under-actuated* (Fantoni & Lozano, 2001). As a result, the control allocation problem admits a solution if and only if the desired wrench belongs to the subspace of admissible wrenches, i.e. the column space of $\mathbf{T}$.

Physical limitations that affect each actuator/effector, such as saturations, dead-zones, etc., restrict the range of admissible values for the input vector to a *feasibility set* $\mathbb{U} \subset \mathbb{R}^n$. As a result, the control allocation problem admits no solutions when the demanded effort goes beyond the actuators' capabilities, even in case of over-actuated vehicles.

The control allocation problem is therefore typically formalised as a constrained optimisation problem of the form:

$$\begin{aligned} \underset{\tilde{\boldsymbol{\tau}}^b, \mathbf{u}}{\arg\min} \quad & \left\| \mathbf{Q}\tilde{\boldsymbol{\tau}}^b \right\| + J(\mathbf{u}) \\ \text{s. t.} \quad & \tilde{\boldsymbol{\tau}}^b = \boldsymbol{\tau}_d^b - \mathbf{T}\,\mathbf{h}(\mathbf{u}) \\ & \mathbf{u} \in \mathbb{U} \end{aligned}, \tag{19}$$

where $\tilde{\boldsymbol{\tau}}^b$ represents the allocation error; $\mathbf{Q} > 0$ is a positive definite weight matrix (usually diagonal) that prioritises the DOFs to be controlled in case the desired control wrench cannot be attained; and $J(\cdot)$ is a cost function that models the secondary objective, typically chosen as $\|\mathbf{W}\mathbf{u}\|$, $\mathbf{W} > 0$. Several approaches are available in literature to solve this problem; for a thorough discussion about the control allocation problem, the interested reader is referred to Johansen and Fossen (2013) and references therein. The OCS control allocation system provides an interface that allows custom allocation algorithms to be used as a replacement of the default algorithm implementation. The default algorithm is composed of two consecutive steps. First, a solution to the unconstrained allocation problem is computed, using the generalised inverse of the TCM (Johansen & Fossen, 2013):

$$\mathbf{u}_\infty = \mathbf{h}^{-1}\left(\mathbf{T}^\dagger \boldsymbol{\tau}_d^b\right) = \mathbf{h}^{-1}\left(\mathbf{W}^{-1}\mathbf{T}^T(\mathbf{T}\mathbf{W}^{-1}\mathbf{T}^T)^{-1}\boldsymbol{\tau}_d^b\right). \tag{20}$$

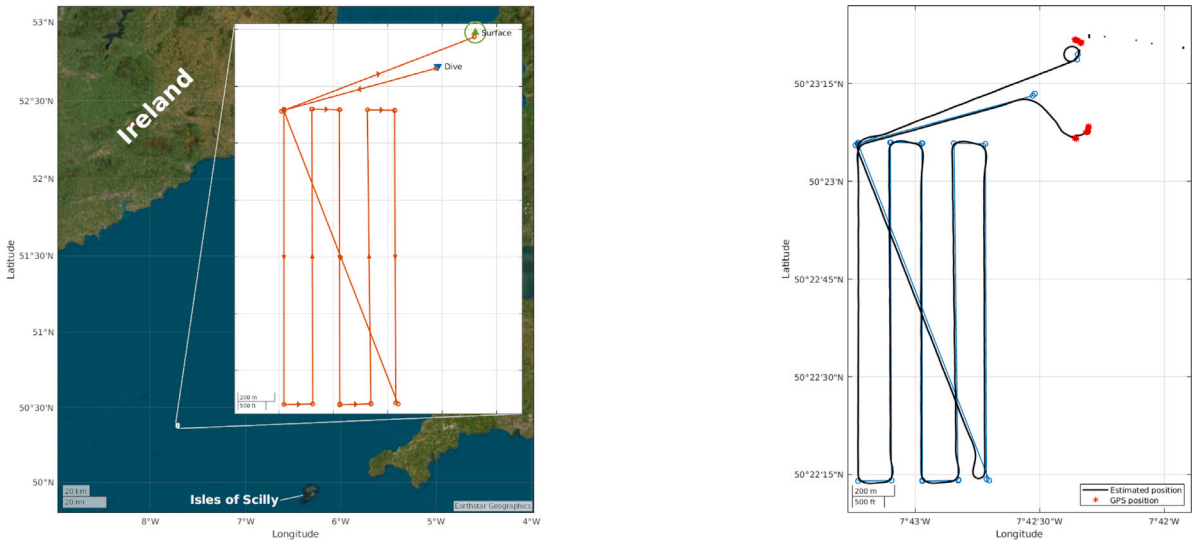
Then, the elements of the input vector $\mathbf{u}_\infty$ are clamped to fall within the bounds the feasibility set $\mathbb{U}$:

$$\mathbf{u}^* = \text{clamp}(\mathbf{u}_\infty, \mathbb{U}) \tag{21}$$

In case of under-actuated vehicles, it is sometimes convenient to restrict the control allocation problem to the $m < 6$ controllable DOFs only. For example, rudder/sternplanes installed on an AUV such as ALR produce lift forces in the sway/heave directions that result in yaw/pitch moments, respectively. However, the motion for this class of vehicles is controlled using steering torques; as a consequence, the sway/heave dynamics are of no interest, and it would be therefore beneficial to exclude these DOFs from the control allocation problem, to avoid considering their contribution. This can be done via a selection matrix $\mathbf{E}_{\text{DOF}} \in \mathbb{R}^{m\times 6}$, whose rows are selected among the rows of the 6-by-6 identity matrix so that the reduced system given by:

$$\boldsymbol{\tau}_{d,\text{red}}^b = \mathbf{E}_{\text{DOF}}\boldsymbol{\tau}_d^b, \quad \mathbf{T}_{\text{red}} = \mathbf{E}_{\text{DOF}}\mathbf{T} \tag{22}$$

includes only the DOFs of interest.



(a) Planned mission.

(b) Estimated path.

Fig. 12. Autosub5, mission AS5M062.

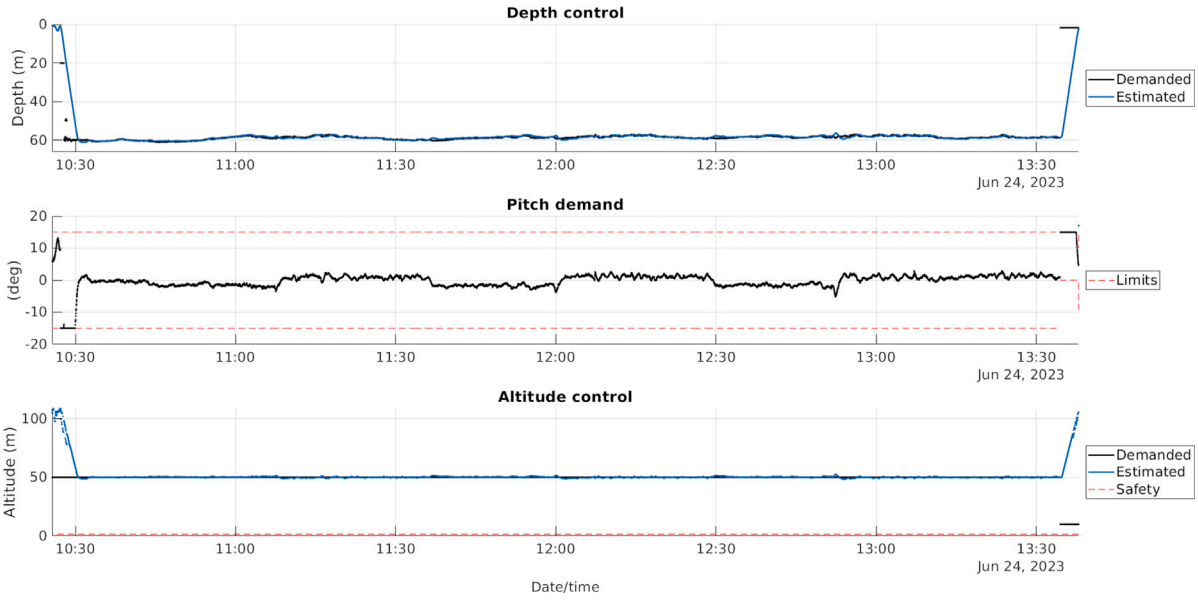


Fig. 13. Autosub5, mission AS5M062: guidance component for the vertical plane. Top: depth control obtained converting the altitude demand with (6). Middle: pitch demand output from the flight-style depth regulator. Bottom: resulting altitude control. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. Results

The GNC system described above has undergone extensive tests during in-water experimental campaigns starting from 2021, exploring a wide range of diverse behaviours and environmental conditions for all the platforms. In the following, results from a selection of three different types of mission are presented to highlight the versatility of the proposed design to vehicles with different capabilities.

5.1. Autosub5

Mission AS5M062 was a high frequency multi-beam survey executed with Autosub5 in the Greater Haig Fras area (Celtic Sea, north-west of the Isles of Scilly) during the DY166 engineering cruise, conducted with the support of RRS Discovery in June 2023. The mission consisted in a lawn-mower path at a constant distance (altitude) of 50 m from the sea-bottom. The lawn-mower was composed of five legs, each approximately 1.6 km long and with a separation of about 200 m. An initial track to drive the vehicle from the launch position to the first point of the lawn mower while diving was additionally defined. Finally, two legs to bring the vehicle back on surface the recovery waypoint were added after the end of the lawn-mower. The complete mission is illustrated in Fig. 12(a).

The terrain-following behaviour is obtained as a by-product of the reactive avoidance strategy defined in the OAS, as described in Section 4.2. The altitude demand is converted into a depth demand via (6); Fig. 13 shows the correspondence between altitude and depth control. For each track composing the desired lawn-mower path, the desired cruise speed, the depth set-point and the track itself represent the mission goal, whereas mission constraints are defined by minimum

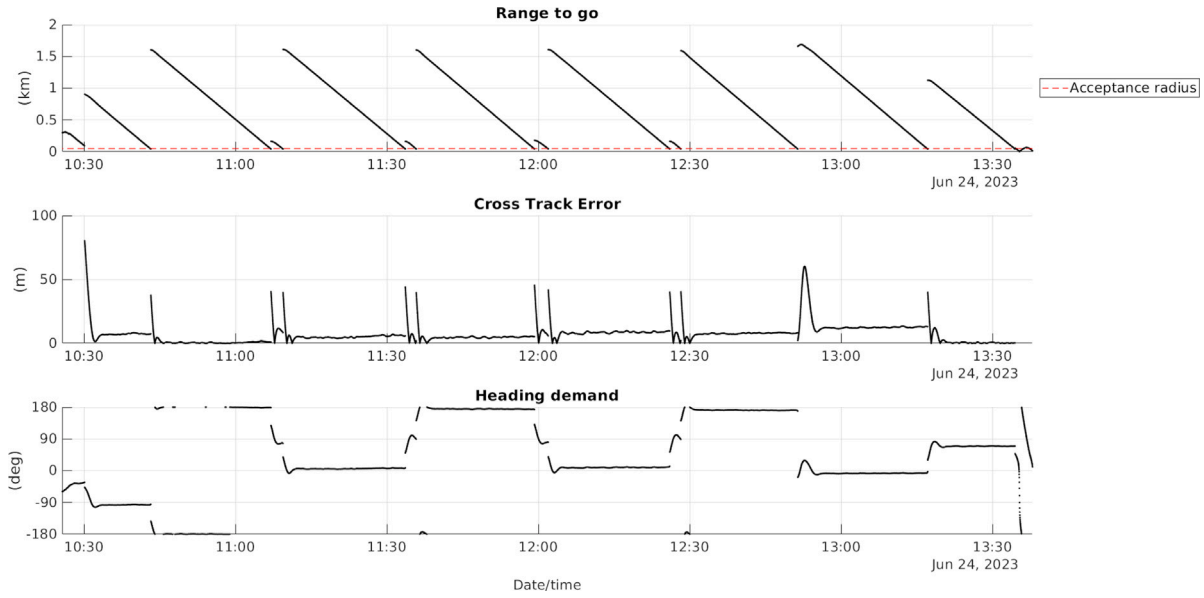


Fig. 14. Autosub5, mission AS5M062: guidance component for the horizontal plane. Top: range to go, calculated as the distance on the horizontal plane between the vehicle's position and the final point of the current track. Middle: cross-track error; note that the cross-track correction term was disabled. Bottom: heading demand resulting from the track-following algorithm.

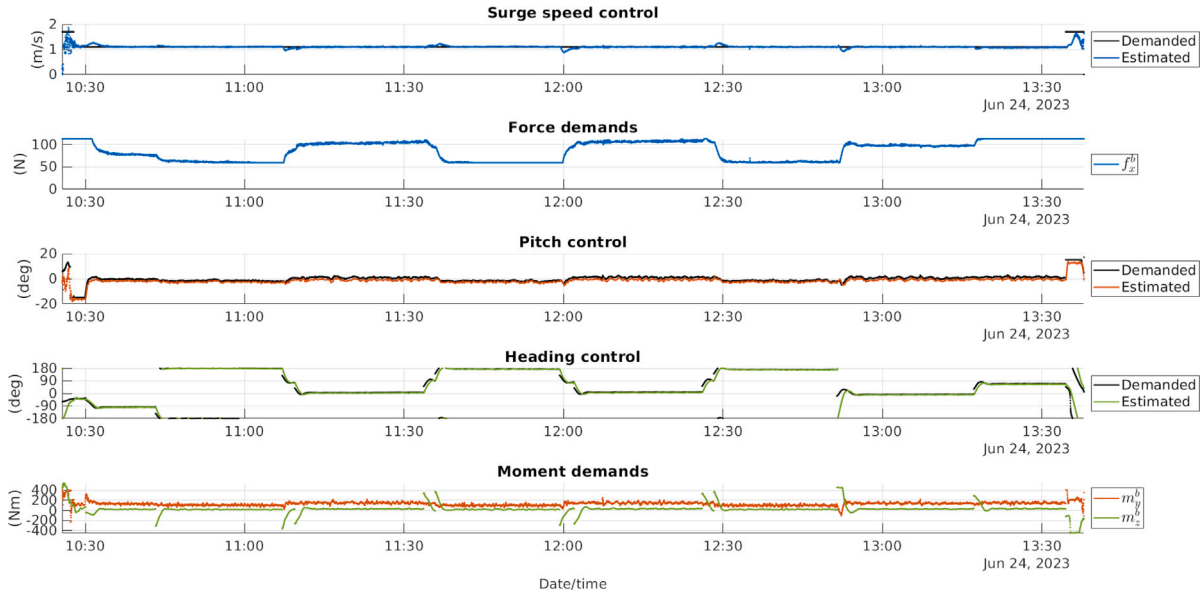


Fig. 15. Autosub5, mission AS5M062: motion controller. First two plots: linear velocity regulator. Last three plots: orientation regulator. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and maximum values for the pitch angle (pitch limits). Mission goal and constraints are given as input to the guidance system, that was instructed to run the "flight-style depth control" mode for the vertical component and the "track-following" algorithm for the horizontal component. The frontal component was instead set to inactive, leaving the roll DOF to evolve freely, with no active control.

The depth regulator on Autosub5 is configured as a PID controller, with an integral time of 135 s and a derivative time of 0.74 s. The pitch angle demand (Fig. 13, second plot) is the output of the depth regulator saturated between the desired pitch limits (dashed red lines), set to ±15 deg for the entire mission except for the surfacing track, where the lower limit was set to 0 deg.

The track-following algorithm was configured to use a look-ahead distance $l = 50$ m. The track-following outputs are the heading demand shown in Fig. 14 (bottom) and the surge speed demand, equal to the desired cruise speed that was set to 1.1 m/s for the entire mission. The switch from one track to the next was performed by the mission executive as the vehicle reached the acceptance radius defined for the final waypoint of each track, set to a constant value of 50 m for each segment of the lawn-mower. The time instants where the switches happen are visible in Fig. 14, in correspondence of the jumps in either the range to go (top) or the heading demand. It is worth mentioning that the cross-track correction given by (9) was disabled for this particular mission; in fact, the cross-track error shown in Fig. 14 (middle) exhibits a non-zero value for most of the mission. Fig. 12(b) reports the estimated navigation path followed by the vehicle on the North-East plane, giving a complete view of the result given by the track-following algorithm.

The surge, pitch and heading demand calculated by the guidance system are then used to feed the motion controller. In case of Autosub5,

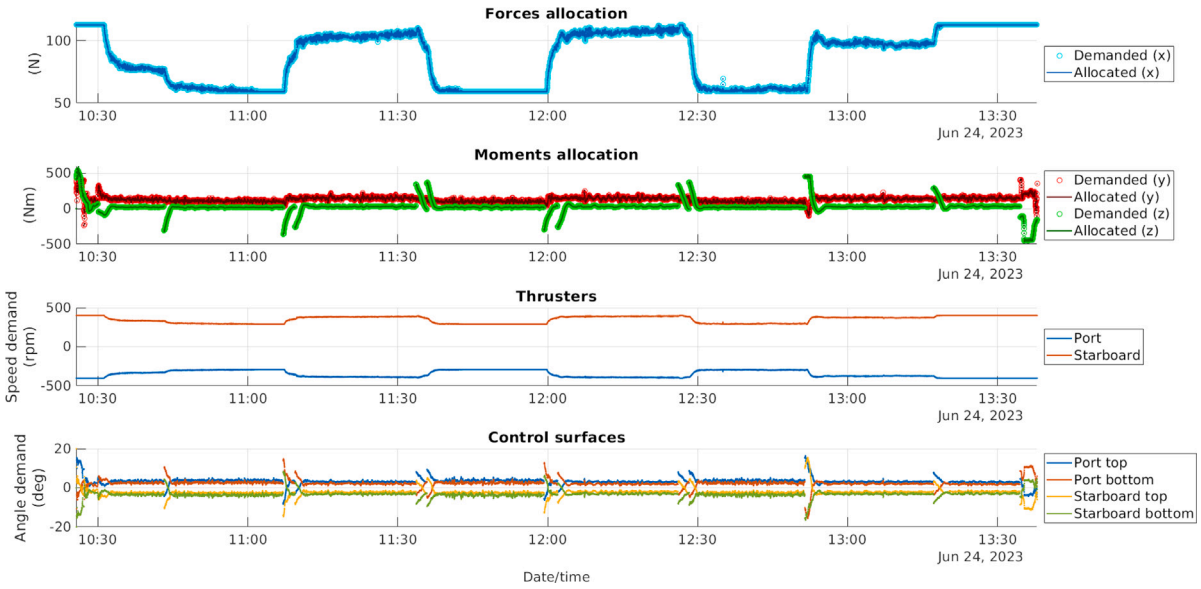


Fig. 16. Autosub5, mission AS5M062: control allocation system. First two plots: comparison between demanded and allocated forces/moments. Last two plots: input commands for the actuators output by the control allocation system.

the linear velocity regulator uses the default implementation (13) configured as a PI ($\mathbf{K}_{d,v} = \mathbf{0}$), with the sway and heave loops set to be inactive. Similarly, the orientation regulator is the default controller in (14) configured as a PI where the roll control loop is disabled and the gain matrices $\mathbf{K}_{p,\Theta}$ and $\mathbf{K}_{i,\Theta}$ are diagonal. It is worth noting that with this setup, the depth control loop results effectively equivalent to a cascaded PID controller, where the outer loop is represented by the depth regulator, and the inner loop by the pitch regulator that produces the final actuation demand.

Fig. 15 shows the result of the linear velocity controller (first two plots) and the orientation controller (last three plots). Comparing the surge force demand (second plot) and the vehicle's heading (fourth plot, green line), it is possible to notice that the surge force required during the tracks heading South (heading demand ± 180 deg) is lower than the force required for the tracks heading North (heading demand 0 deg). This was due to the presence of strong sea-currents running North to South in the mission area.

The demanded forces/moments calculated by the motion controller are received by the control allocation system. On Autosub5, the control allocation problem is restricted to the force along the body-x axis f_x^b, and the moments about the body-y and body-z axis m_y^b and m_z^b, using the selection matrix:

$$\mathbf{E}_{\text{DOF}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{23}$$

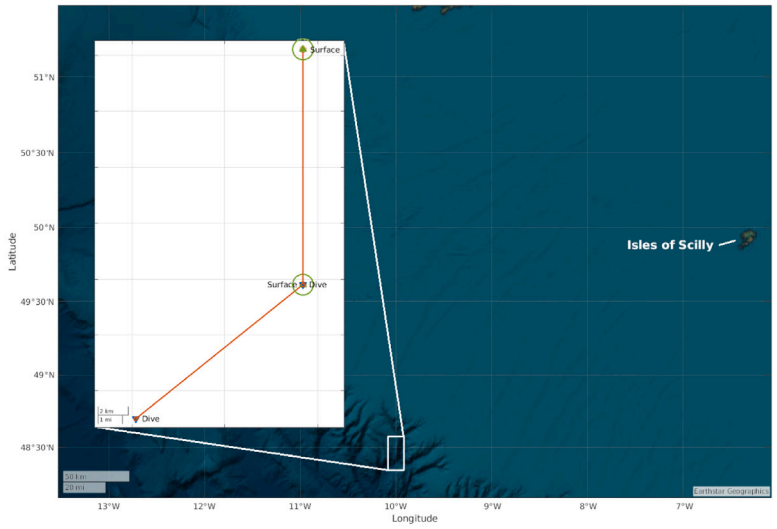
The first two plots at the top in Fig. 16 show the comparison between the demanded forces/moments and the allocated ones. The allocated forces and moments are calculated using (18), where the actuators' input vector $\boldsymbol{u}$ is filled with the actuators' input commands resulting from the solution of the control allocation problem. In case of Autosub5 the input commands are represented by the angular speed demand for the two counter-rotating propellers (Fig. 16, third plot) and by the angle demand for the four control planes (Fig. 16, fourth plot).

5.2. ALR

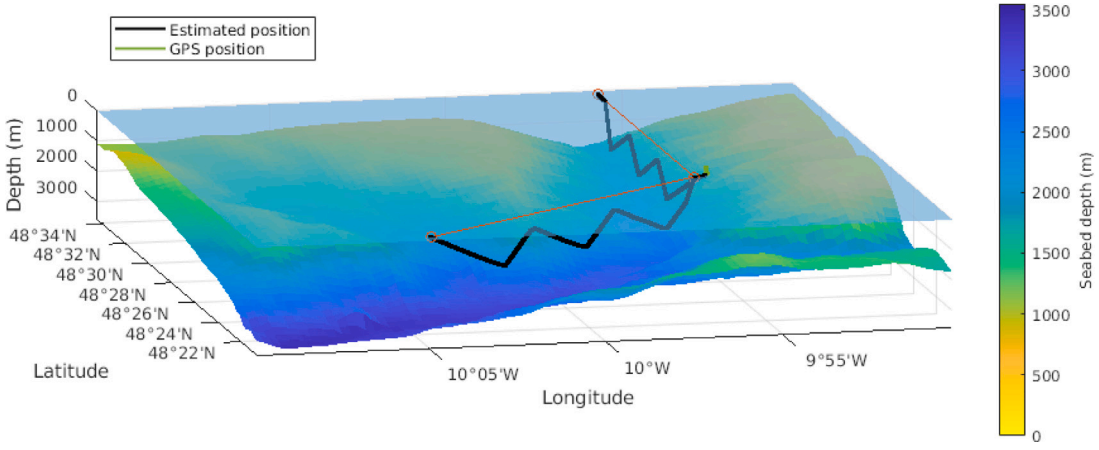
ALR-4 was also deployed on the expedition DY166, where it executed a CTD profiling mission (ALR4M130) up the slope of Whittard Canyon, between the Celtic Sea and the Atlantic Ocean. A profiling mission consists in repeated ascents and descents between two desired depths as the vehicle navigates forward, resulting in the typical saw-toothed profile visible in Fig. 17(b). The profiling mission ALR4M130 was executed in the mid-water column, between 500 and 1500 metres of depth, with no DVL bottom track. The path was split in two transects, interleaved with a surfacing to reset the navigation error through a GPS fix: the first part was a 14.2 km long track heading North-East, whereas the second one was a 15.6 km long track heading North (Fig. 17(a)).

The profiling behaviour is implemented at the mission executive level, so that the guidance system can be commanded using the same modalities selected for Autosub5's mission AS5M062: "flight-style depth control", "track-following" and "inactive" respectively for the vertical, horizontal and frontal components. After being achieved according to (2), the current depth set-point is maintained for a given *hold time*; once the hold time has elapsed, the helm switches the depth demand to the other value, and so on (Fig. 18, top). In mission ALR4M130 the depth tolerance was set to 50 m, and the hold time to 300 s for both the tracks. During the ascent and descent phases, the pitch limits are set to the same value (30 deg on the ascent and -30 deg on the descent in mission ALR4M130, see Fig. 18, middle-top), effectively overriding the output of the depth regulator. The intent of forcing the pitch angle demand to a constant value was to obtain a constant ascent/descent rate (Fig. 18, third plot).

The track-following algorithm was configured with a look-ahead distance $l = 50$ m and the cross-track error correction enabled. The effect of the latter is visible in the last plot at the bottom of Fig. 18, where the cross-track error for the profiling legs is reported. The configuration of the motion controller and the control allocation system was very similar to the one used on Autosub5 in mission AS5M062. The only difference was represented by the orientation regulator, that on ALR is configured as a decoupled PID regulator (e.g. equivalent to the form shown in (13)) with the roll DOF disabled. This configuration is effectively equivalent to the legacy control system of ALR, based on two decoupled control loops: pitch angle regulation via sternplanes actuation, and heading control via rudder actuation (McPhail, Templeton, Pebody, Roper, & Morrison, 2019). It is worth remarking that the capability of configuring the proposed GNC architecture to replicate well-established systems, extensively tested over several years of in water deployments is a major feature, made possible by its versatile and adaptable design. The configuration parameters of the two controllers have been determined from the tuning values of the legacy control system, leveraging the outcomes of years of field-experience. In particular,



(a) Planned mission.



(b) Estimated path.

Fig. 17. ALR-4, mission ALR4M130.

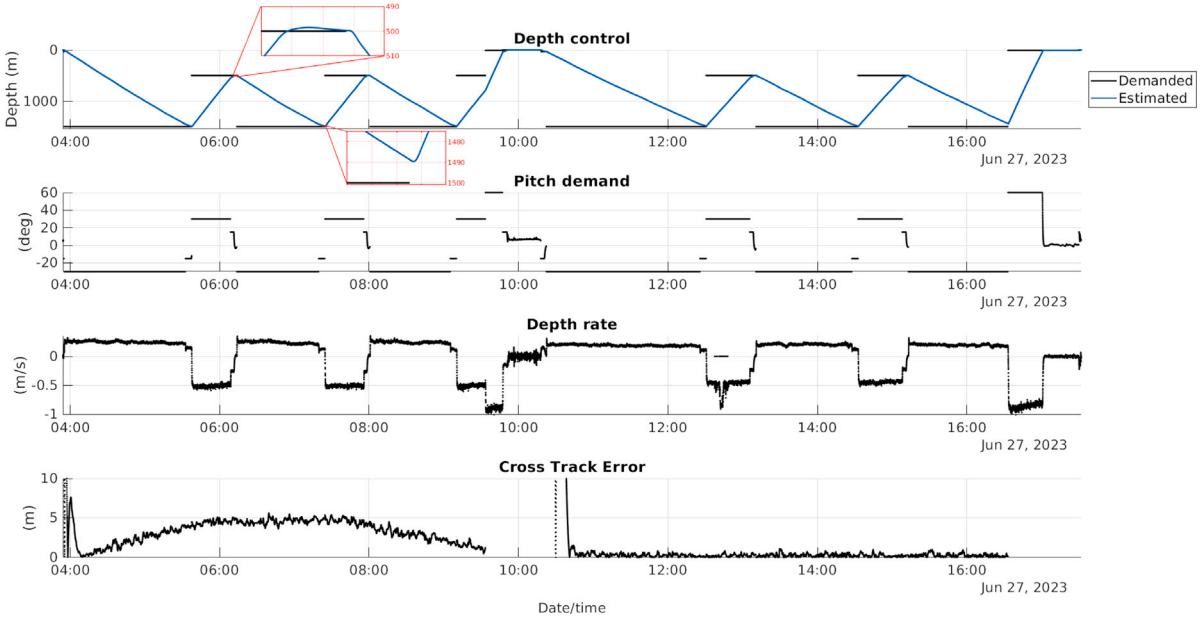
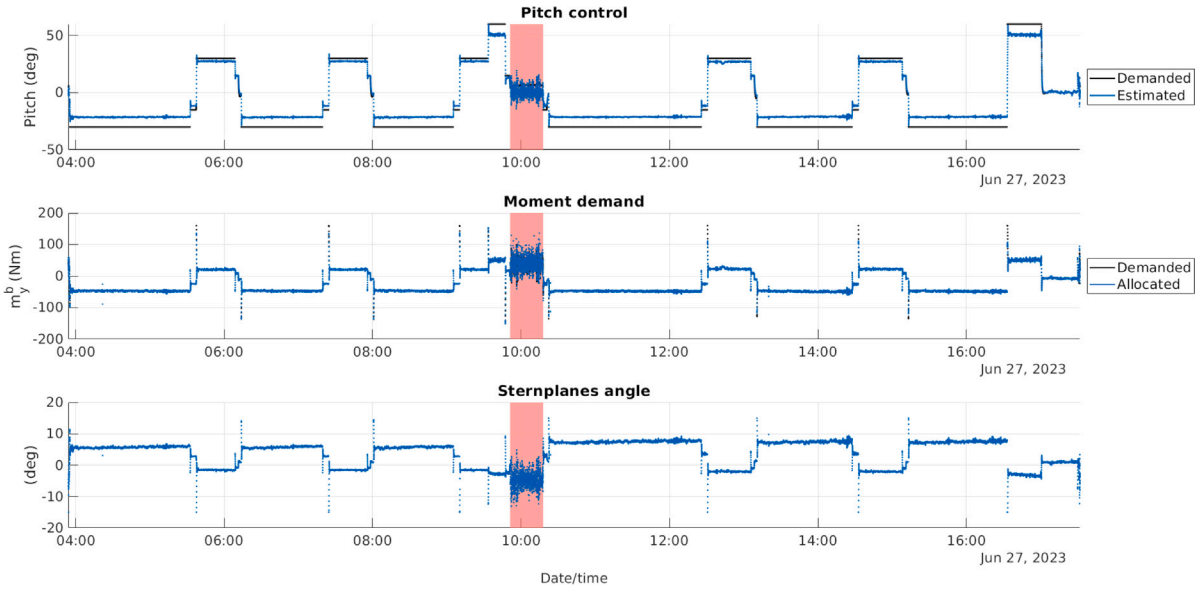
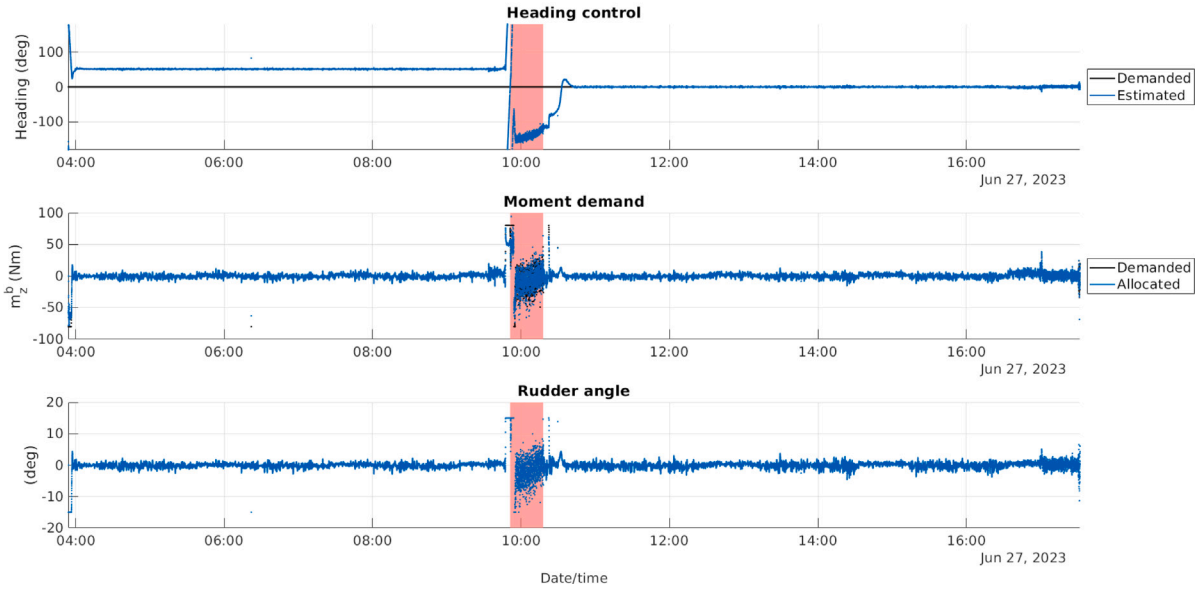


Fig. 18. ALR-4, mission ALR4M130: guidance system. Top: vertical component, depth control. Middle-top: vertical component, pitch demand resulting from the depth regulator. Middle-bottom: vertical component, depth rate (ascent/descent). Bottom: horizontal component, cross track error.



(a) Pitch control loop. Top: pitch control performance. Middle: comparison between demanded and allocated moment about the body-y axis. Bottom: angle demand for the sternplanes output by the control allocation system.



(b) Heading control loop. Top: heading control performance. Middle: comparison between demanded and allocated moment about the body-z axis. Bottom: angle demand for the rudder output by the control allocation system.

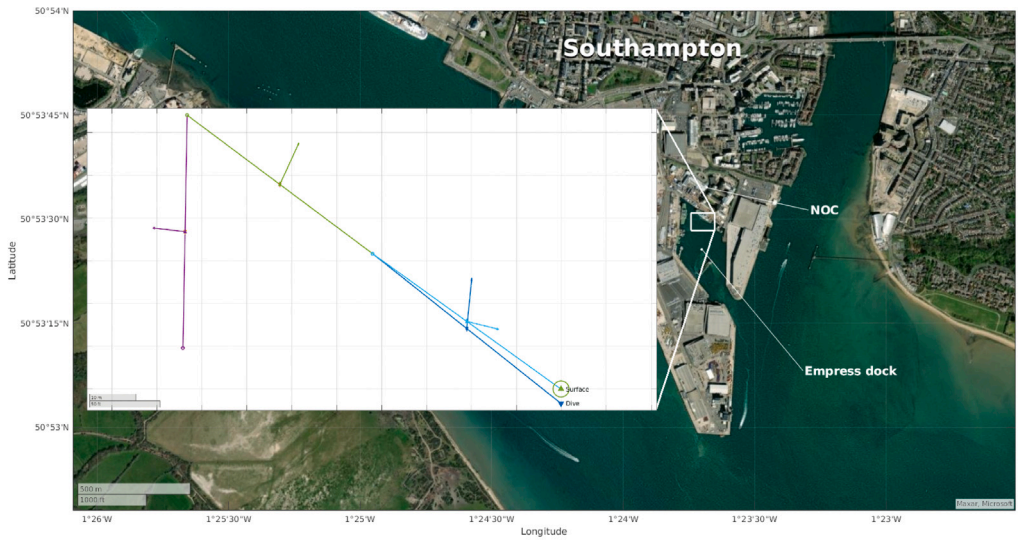
Fig. 19. ALR-4, mission ALR4M130: sternplanes and rudder actuation control. The red bands on the background corresponds to the period in which the vehicle was on surface. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the pitch regulator was configured as a PI controller with an integral time of 125 s, whereas the heading regulator was a simple proportional regulator.

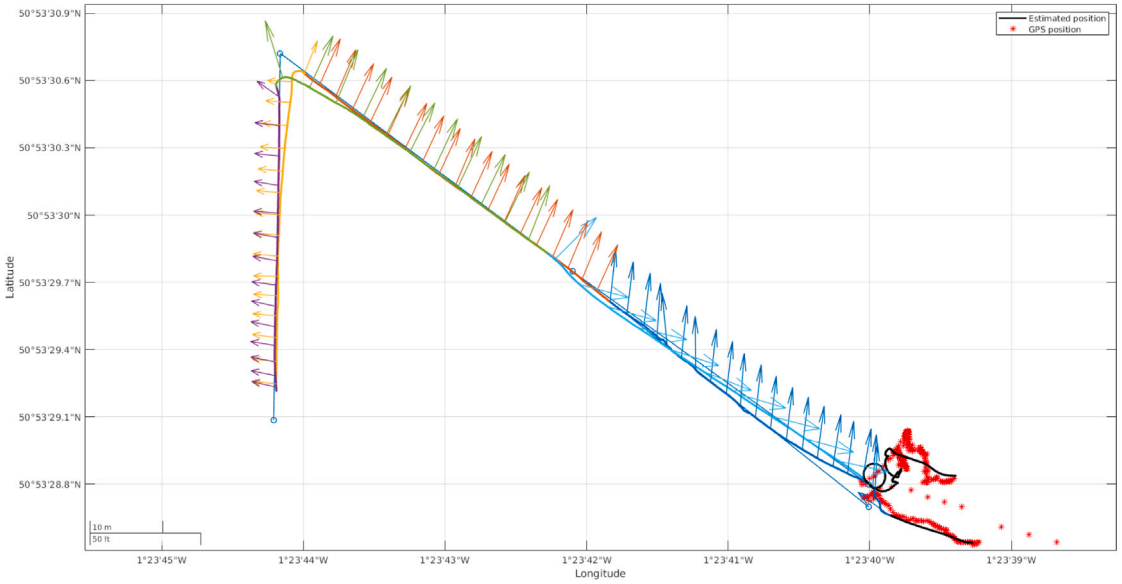
The performance of the two control loops in mission ALR4M130 is reported in Fig. 19. It is possible to note that the pitch angle exhibits an asymmetric steady-state error (Fig. 19(a), top), despite of the integral correction included in the controller. This is due to a non-optimal tuning of the anti-windup strategy for the specific parameters of the selected mission. The asymmetry is caused by a combination of several hydrodynamics effects resulting in a positive pitching moment, hence aiding/contrasting the control action on the ascents/descents. Although a finer tuning of the controller would help compensating those issues and improving the overall behaviour, a complete characterisation of the control performances goes beyond the scope of this paper.

5.3. AH-1

The mission selected for AH-1 is referred to as AH1M007, and it was executed in Empress Dock, the water basin in front of the NOC facilities in Southampton, UK, as part of the preliminary experimental tests of this prototype. AH1M007 is composed of a sequence of six tracks at 2 m depth to be travelled with different slant headings. Fig. 20(a) shows the tracks with different colours; the arrows associated with the tracks indicate the desired heading for the corresponding segment of the mission. Note that the violet and the green tracks are both superimposed on another leg reciprocal to them, as visible in Fig. 20(b), where the estimated path followed by the vehicle is reported. The arrows indicate the heading angle estimated by the vehicle during the mission, demonstrating that the vehicle executed the desired behaviour.



(a) Planned mission.



(b) Estimated path.

Fig. 20. AH-1, mission AH1M007. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

For every track, the guidance system was commanded to use the "track-following strafe" algorithm for the horizontal component and the "hover-capable depth control" mode for the vertical component. Moreover, the frontal component was set to active, asking a constant roll demand of 0 deg to keep the vehicle levelled. Fig. 21 illustrates the outputs of the track-following strafe algorithm. Given the high manoeuvrability of AH-1, the look-ahead distance was set to a much smaller value ($l = 5$ m) compared to the one used for Autosub5 and ALR. The desired cruise speed on the horizontal plane (top plot) gets split into the surge and sway speed demands (middle-top and middle-bottom plots, respectively) according to (12). Moreover, the heading demand (bottom plot) is set to the desired value associated with each track.

The depth controller produces the heave demand shown in the bottom plot of Fig. 22, as well as a constant pitch demand of 0 deg. The top plot of Fig. 22 illustrates the result of the depth control, obtained through the joint action on the roll, pitch, and heave DOFs.

On AH-1, the motion controller is configured to actively control all the six DOFs. More specifically, both the linear velocity and the orientation regulator use the default implementation given in (13) and (14), respectively, with the gain matrices all diagonal. The results of the motion controller are reported in Fig. 23. Overall, the actions produced by the motion controller are effective to execute the desired mission. Performances tend to degrade as the mission demands approach the operative limits of the vehicle, and further optimisations can be done with a finer tuning of the controllers' parameters. For instance, roll control becomes more challenging the higher the sway speed is (see between 09:58 and 10:12). However, as previously mentioned, the analysis presented in this work aims at demonstrating how the proposed system can be adapted and deployed on very different marine robotic platforms, and a thorough discussion on the control performances goes beyond the scope of this paper.

The control allocation system on AH-1 computes the full six DOFs solution. The comparison between the demanded and the allocated forces and moments are shown in Fig. 24.

6. Conclusions

This paper presents the modular Guidance, Navigation and Control architecture developed for the Autosub Family of AUVs. The current

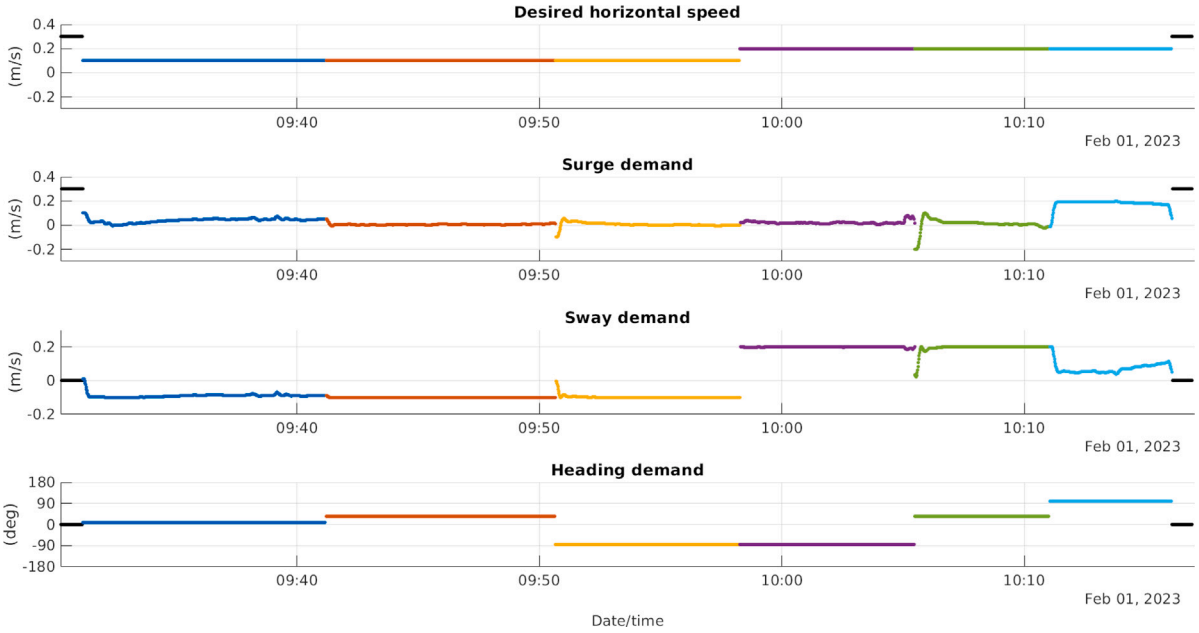


Fig. 21. AH-1, mission AH1M007: results of the guidance component for the horizontal plane. The colour code corresponds to the one used in Fig. 20.

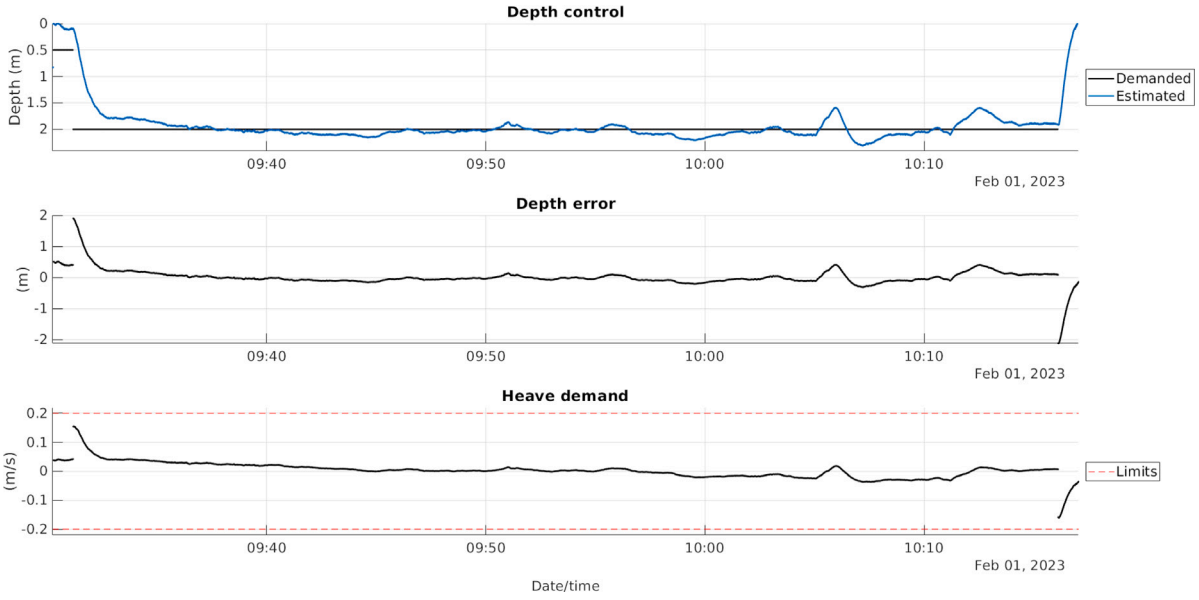
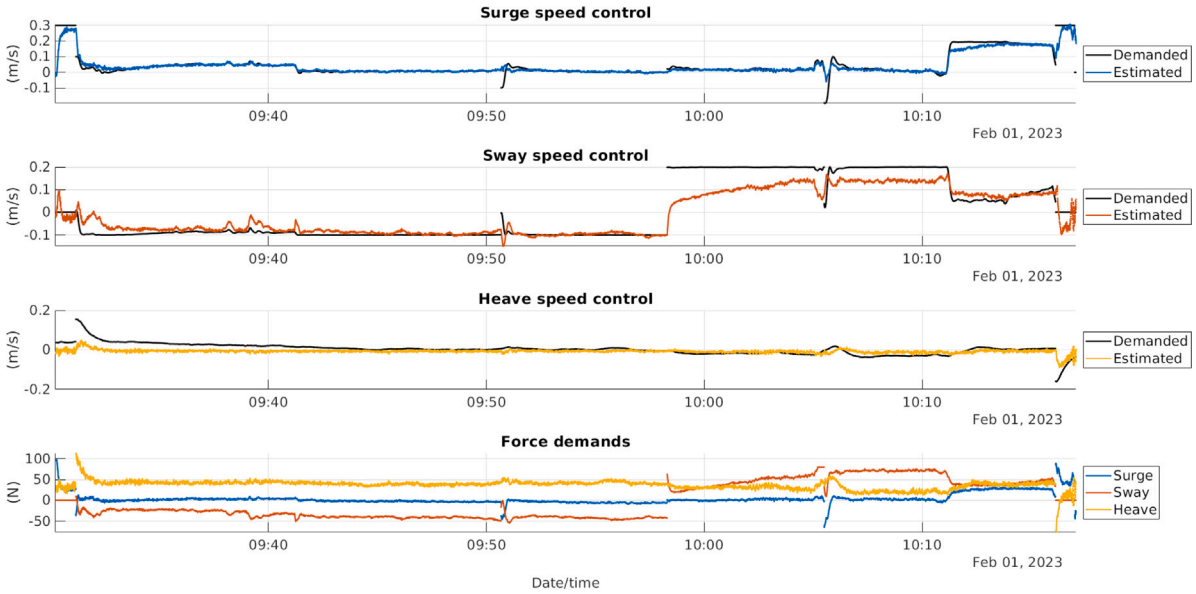


Fig. 22. AH-1, mission AH1M007: results of the guidance component for the vertical plane. The heave demand (bottom) is the result of the depth regulator.

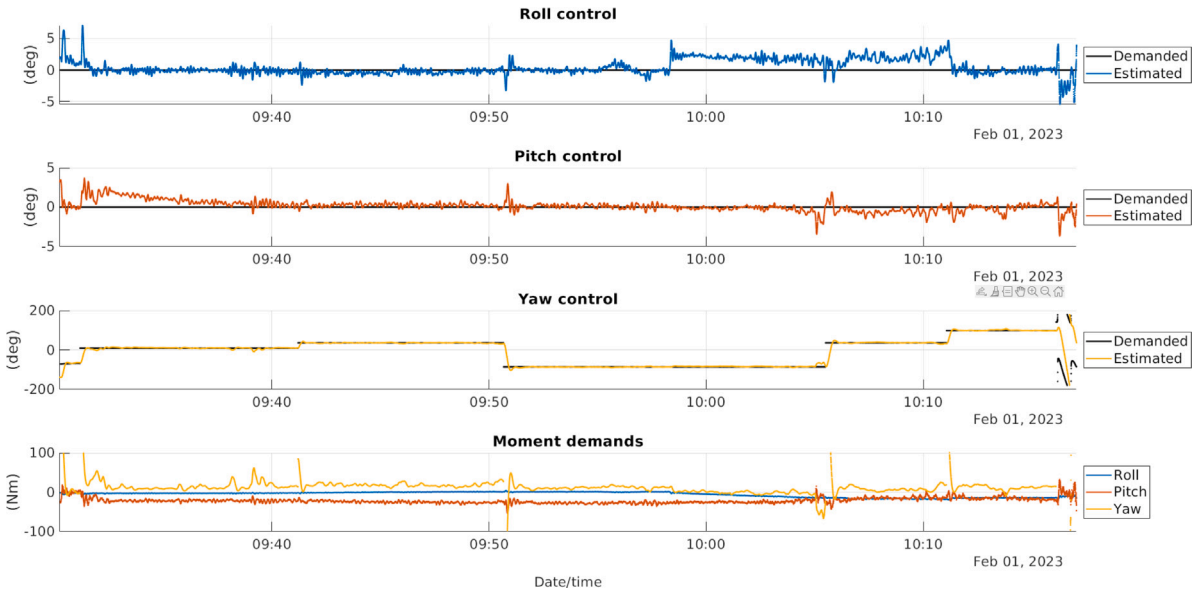
generation of Autosub vehicles includes: the under-actuated flight style Autosub Long Range, Autosub5 a flight-style AUV with redundancy and the hover style AUV AH-1. To support vehicles with such diverse actuation and address the peculiar set of requirements defined for the Autosubs, a robust architecture has been adopted with generic interfaces between highly configurable modules. The ability of the proposed approach to adapt to these different classes of AUVs operating in real world conditions is illustrated via examples from recent campaigns. Although not optimised, default algorithms implemented for each component of the GNC enable the execution of a broad range of typical marine surveys. When improved performance are required for specific types of missions, the flexible structure of the proposed GNC allows the replacement of default algorithms with more refined versions. The development and testing of more advanced techniques will be subject of future investigations driven by the requirements of new emerging applications involving the Autosub platforms. Future extensions to the proposed approach also include: active management of actuator faults and inclusion of buoyancy engines within the control allocation framework.

CRediT authorship contribution statement

Davide Fenucci: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Francesco Fanelli:** Writing – review & editing, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Alberto Consensi:** Writing – review & editing, Visualization, Validation, Supervision,



(a) Linear velocity regulator.



(b) Orientation regulator.

Fig. 23. AH-1, mission AH1M007: results of the motion controller.

Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Georgios Salavasidis:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Miles Pebody:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Conceptualization. **Alexander B. Phillips:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Development of Autosub5 and ALR was funded by the UK's Industrial Strategy Challenge Fund and Natural Environment Research Council's Oceanids Programme and successor National Underpinning Autonomous Capability funding. Development of AH-1 was funded by Innovate UK (Grant Numbers: 104822 and 10011940) and the Defence and Security Accelerator grant "Unmanned Underwater Vehicles in Coastal Waters".

The Authors wish to thank the past and present members of the Marine Autonomous and Robotic Systems (MARS) Group at the National Oceanography Centre who contributed to the development of the Autosub vehicles, as well as the Captain, Crew and Technical Party on-board the RRS Discovery for expedition DY166.

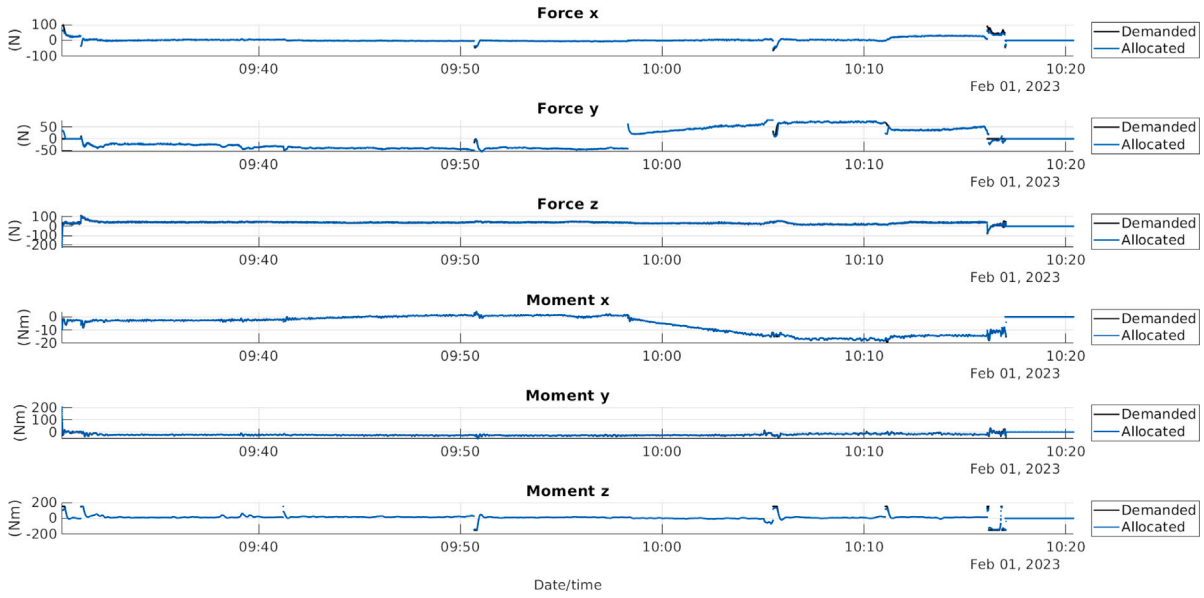


Fig. 24. AH-1, mission AH1M007: comparison between demanded and allocated forces and moments.

References

A Diverse team. (2016). https://ardupilot.org/.

Antonelli, G. (2006). *Underwater Robots* (pp. 15–44). Berlin, Heidelberg: Springer Berlin Heidelberg.

Åström, K. J., & Hägglund, T. (2006). *Advanced PID Control*. ISA - The Instrumentation, Systems and Automation Society.

Azkarate, M., Gerdes, L., Joudrier, L., & Pérez-del Pulgar, C. J. (2020). A GNC architecture for planetary rovers with autonomous navigation. In *2020 IEEE international conference on robotics and automation* (pp. 3003–3009). http://dx.doi.org/10.1109/ICRA40945.2020.9197122.

Benjamin, M. R., Schmidt, H., Newman, P. M., & Leonard, J. J. (2010). Nested autonomy for unmanned marine vehicles with MOOS-ivp. *Journal of Field Robotics*, *27*(6), 834–875. http://dx.doi.org/10.1002/rob.20370.

Caffaz, A., Caiti, A., Casalino, G., & Turetta, A. (2010). The hybrid glider/AUV folaga. *IEEE Robotics & Automation Magazine*, *17*(1), 31–44.

Carreras, M., Hernández, J. D., Vidal, E., Palomeras, N., Ribas, D., & Ridao, P. (2018). Sparus II AUV—A hovering vehicle for seabed inspection. *IEEE Journal of Oceanic Engineering*, *43*(2), 344–355. http://dx.doi.org/10.1109/JOE.2018.2792278.

Connelly, D. P., Copley, J. T., Murton, B. J., Stansfield, K., Tyler, P. A., German, C. R., et al. (2012). Hydrothermal vent fields and chemosynthetic biota on the world's deepest seafloor spreading centre. *Nature Communications*, *3*(1), 1–9.

Consensi, A., Kingsland, M., Linton, N., Bowring, L., Roper, D. T., Austin-Berry, R., et al. (2022). Autosub5: Preparing for science. In *2022 IEEE/OES autonomous underwater vehicles symposium* (pp. 1–6). IEEE.

Costanzi, R., Fenucci, D., Manzari, V., Micheli, M., Morlando, L., Terracciano, D., et al. (2020). Interoperability among unmanned maritime vehicles: Review and first in-field experimentation. *Frontiers in Robotics and AI*, *7*, http://dx.doi.org/10.3389/frobt.2020.00091.

D'Angelo, V., Folino, P., Lupia, M., Gagliardi, G., Cario, G., Gaccio, F. C., et al. (2022). A ROS-based GNC architecture for autonomous surface vehicle based on a new multimission management paradigm. *Drones*, *6*(12), http://dx.doi.org/10.3390/drones6120382.

Eickstedt, D. P., & Sideleau, S. R. (2010). The backseat control architecture for autonomous robotic vehicles: A case study with the Iver2 AUV. *Marine Technology Society Journal*, *44*(4), 42–54.

Fanelli, F., Fenucci, D., Marlow, R., Pebody, M., & Phillips, A. B. (2020). Development of a multi-platform obstacle avoidance system for autonomous underwater vehicles. In *2020 IEEE/OES autonomous underwater vehicles symposium* (pp. 1–6). http://dx.doi.org/10.1109/AUV50043.2020.9267942.

Fantoni, I., & Lozano, R. (2001). *Non-linear Control for Underactuated Mechanical Systems*. Springer London, http://dx.doi.org/10.1007/978-1-4471-0177-2.

Fossen, T. I. (2002). *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Trondheim: Marine Cybernetics AS.

Graham, A. G., Dutrieux, P., Vaughan, D. G., Nitsche, F. O., Gyllencreutz, R., Greenwood, S. L., et al. (2013). Seabed corrugations beneath an antarctic ice shelf revealed by autonomous underwater vehicle survey: origin and implications for the history of pine island glacier. *Journal of Geophysical Research: Earth Surface*, *118*(3), 1356–1366.

Griffiths, G. (2012). Steps towards autonomy: From current measurements to underwater vehicles. *Methods in Oceanography*, *1*, 22–48.

Hagen, P. (2001). AUV/UUV mission planning and real time control with the HUGIN operator system. In *MTS/IEEE oceans 2001. an ocean odyssey. conference proceedings (IEEE cat. no.01CH37295), vol. 1* (pp. 468–473 vol.1). http://dx.doi.org/10.1109/OCEANS.2001.968769.

Harris, C., Lorenzo, A., Jones, O., Buck, J., Kokkinaki, A., Loch, S., et al. (2020). Oceanids C2: An integrated command, control and data infrastructure for the over-the-horizon operation of marine autonomous systems. *Frontiers in Marine Science*, *submitted*.

Hussain, R., & Zeadally, S. (2019). Autonomous cars: Research results, issues, and future challenges. *IEEE Communications Surveys & Tutorials*, *21*(2), 1275–1313. http://dx.doi.org/10.1109/COMST.2018.2869360.

Jenkins, A., Dutrieux, P., Jacobs, S. S., McPhail, S. D., Perrett, J. R., Webb, A. T., et al. (2010). Observations beneath pine island glacier in west antarctica and implications for its retreat. *Nature Geoscience*, *3*(7), 468–472.

Johansen, T. A., & Fossen, T. I. (2013). Control allocation—A survey. *Automatica*, *49*(5), 1087–1103. http://dx.doi.org/10.1016/j.automatica.2013.01.035.

Kendoul, F. (2012). Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, *29*(2), 315–378. http://dx.doi.org/10.1002/rob.20414.

Manhães, M. M. M., Scherer, S. A., Voss, M., Douat, L. R., & Rauschenbach, T. (2016). UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In *OCEANS 2016 MTS/IEEE Monterey* (pp. 1–8). http://dx.doi.org/10.1109/OCEANS.2016.7761080.

McPhail, S. D. (2009). Autosub6000: A deep diving long range AUV. *Journal of Bionic Engineering*, *6*(1), 55–62.

McPhail, S. D., Furlong, M. E., Pebody, M., Perrett, J., Stevenson, P., Webb, A., et al. (2009). Exploring beneath the PIG ice shelf with the Autosub3 AUV. In *Oceans 2009-Europe* (pp. 1–8). IEEE.

McPhail, S. D., & Pebody, M. (1998). Navigation and control of an autonomous underwater vehicle using a distributed, networked, control architecture. *Underwater Technology*, *23*(1), 19–30. http://dx.doi.org/10.3723/175605498783259975.

McPhail, S. D., Templeton, R., Pebody, M., Roper, D., & Morrison, R. (2019). Autosub long range AUV missions under the filchner and ronne ice shelves in the weddell sea, antarctica - an engineering perspective. In *OCEANS 2019 - marseille* (pp. 1–8). http://dx.doi.org/10.1109/OCEANSE.2019.8867206.

Munafo, A., Pebody, M., Consensi, A., Fanelli, F., Fenucci, D., Fox, P., et al. (2019). The NOCS on-board control system. In *OCEANS 2019-marseille* (pp. 1–8). IEEE.

Nađ, Đ., Mišković, N., & Mandić, F. (2015). Navigation, guidance and control of an overactuated marine surface vehicle. *Annual Reviews in Control*, *40*, 172–181. http://dx.doi.org/10.1016/j.arcontrol.2015.08.005.

Palomeras, N., El-Fakdi, A., Carreras, M., & Ridao, P. (2012). COLA2: A control architecture for AUVs. *IEEE Journal of Oceanic Engineering*, *37*(4), 695–716. http://dx.doi.org/10.1109/JOE.2012.2205638.

Phillips, A. B., Kingsland, M., Linton, N., Baker, W., Bowring, L., Soper, S., et al. (2020). Autosub 2000 under ice: Design of a new work class AUV for under ice exploration. In *2020 IEEE/OES autonomous underwater vehicles symposium* (pp. 1–8). IEEE.

Phillips, A. B., Templeton, R., Roper, D., Morrison, R., Pebody, M., Bagley, P. M., et al. (2023). Autosub long range 1500: A continuous 2000 km field trial. *Ocean Engineering*, *280*, Article 114626. http://dx.doi.org/10.1016/j.oceaneng.2023.114626.

Pinto, J., Dias, P. S., Martins, R., Fortuna, J., Marques, E., & Sousa, J. (2013). The LSTS toolchain for networked vehicle systems. In *2013 MTS/IEEE OCEANS - bergen* (pp. 1–9). http://dx.doi.org/10.1109/OCEANS-Bergen.2013.6608148.

Pollini, L., Antonelli, G., Arrichiello, F., Caiti, A., Casalino, G., De Palma, D., et al. (2020). AUV navigation, guidance, and control for geoseismic data acquisition. In *Autonomous underwater vehicles: design and practice* (pp. 469–491). IET, Editor: Frank Ehlers.

Pollini, L., Razzanelli, M., Pinna, F., Indiveri, G., Simetti, E., Alibani, M., et al. (2018). Development of the guidance navigation and control system of the folaga AUV for autonomous acoustic surveys in the WiMUST project. In *OCEANS 2018 MTS/IEEE charleston* (pp. 1–6). http://dx.doi.org/10.1109/OCEANS.2018.8604632.

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, no. 3.2* (p. 5). Kobe, Japan.

Ribas, D., Palomeras, N., Ridao, P., Carreras, M., & Mallios, A. (2012). Girona 500 AUV: From survey to intervention. *IEEE/ASME Transactions on Mechatronics*, *17*(1), 46–53. http://dx.doi.org/10.1109/TMECH.2011.2174065.

Roper, D., Harris, C. A., Salavasidis, G., Pebody, M., Templeton, R., Prampart, T., et al. (2021). Autosub long range 6000: a multiple-month endurance AUV for deep-ocean monitoring and survey. *IEEE Journal of Oceanic Engineering*, *46*(4), 1179–1191.

Rudnick, D. L., Davis, R. E., Eriksen, C. C., Fratantoni, D. M., & Perry, M. J. (2004). Underwater gliders for ocean research. *Marine Technology Society Journal*, *38*(2), 73–84.

Silvestrini, S., Cassinis, L. P., Hinz, R., Gonzalez-Arjona, D., Tipaldi, M., Visconti, P., et al. (2023). Chapter fifteen - modern spacecraft GNC. In V. Pesce, A. Colagrossi, S. Silvestrini (Eds.), *Modern spacecraft guidance, navigation, and control* (pp. 819–981). Elsevier, http://dx.doi.org/10.1016/B978-0-323-90916-7.00015-9.

SNAME, TheSocietyofNavalArchitectureandMarineEngineers (1950). Nomenclature for treating the motion of a submerged body through a fluid. *The Society of Naval Architects and Marine Engineers, Technical and Research Bulletin*, (1950), 1–5.

Sprague, C. I., Özkahraman, Ö., Munafò, A., Marlow, R., Phillips, A., & Ögren, P. (2018). Improving the modularity of AUV control systems using behaviour trees. In *2018 IEEE/OES autonomous underwater vehicle workshop* (pp. 1–6). http://dx.doi.org/10.1109/AUV.2018.8729810.

Tanakitkorn, K., Wilson, P. A., Turnock, S. R., & Phillips, A. B. (2017). Depth control for an over-actuated, hover-capable autonomous underwater vehicle with experimental verification. *Mechatronics*, *41*, 67–81.

Testor, P., DeYoung, B., Rudnick, D. L., Glenn, S., Hayes, D., Lee, C., et al. (2019). Oceangliders: a component of the integrated GOOS. *Frontiers in Marine Science*, *6*, 422.

Wynn, R. B., Huvenne, V. A., Le Bas, T. P., Murton, B. J., Connelly, D. P., Bett, B. J., et al. (2014). Autonomous underwater vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, *352*, 451–468.