Intel Serv Robotics (2012) 5:133–146
DOI 10.1007/s11370-012-0108-0

ORIGINAL RESEARCH PAPER

A coordination architecture for UUV fleets

A. Freddi · S. Longhi · A. Monteriù

Received: 11 March 2011 / Accepted: 21 February 2012 / Published online: 6 March 2012

Abstract This paper presents a modular and expandable architecture, which includes diversified functions and can be applied to heterogeneous fleets of unmanned underwater vehicles (UUVs), to solve the problem of decentralized formation coordination. The architecture is modular and each module is built such that it can solve a precise task using one or more functions. Three functions among them play a key role for the whole architecture: localization, faultless formation control and fault tolerance. The localization function is performed by the use of an adaptive extended Kalman filter (A-EKF) algorithm; the fault-free formation control function is based on a nonlinear decentralized model predictive control (ND-MPC) algorithm; the fault tolerance function is based on a hierarchy graph theory. The novelty of the paper lies in the use of the above mentioned functions as the core of an architecture which is expandable, decentralized and can be applied to a wide range of vehicles.

Keywords Coordination architecture · Formation control · Underwater vehicles

1 Introduction

With the technological progress in unmanned underwater vehicles (UUVs), the idea of deploying multiple vehicles on coordinated missions has become reality. Fleets of UUVs have indeed enormous potential to successfully undertake different missions such as wide-area oceanographic research, scientific survey and sampling to the full depth of the ocean, seabed mapping, tactical surveillance, mine reconnaissance and many others [7,10,16,19,24,41].

The use of multiple vehicles in formation can increase speed, reliability and measurements quality. To control a multi-vehicle underwater formation in a coordinated fashion to achieve a set of goals, three main problems can be identified:

- localization,
- formation control,
- fault tolerance.

In this paper, an approach to solve the problem of coordinating UUV fleets is proposed based on a modular and expandable decentralized architecture. Each module is built such that it can solve a precise task using one or more functions. Three functions among them play a key role for the whole architecture: localization, faultless formation control and fault tolerance.

1.1 Localization

To successfully accomplish the coordination tasks, particular attention must be paid to the localization problem. Indeed, the most straightforward approach for performing formation control would be to measure the absolute position of each underwater vehicle and use that information to keep the relative distances at the desired values. This approach, however, can not be used for an underwater vehicle as the global positioning system (GPS) is able to receive data from the satellites

A. Freddi · S. Longhi (✉) · A. Monteriù
Dipartimento di Ingegneria dell'Informazione,
Via Brecce Bianche, 60131 Ancona, Italy
e-mail: s.longhi@univpm.it

A. Freddi
e-mail: freddi@diiga.univpm.it

A. Monteriù
e-mail: a.monteriu@univpm.it

only when the vehicle is on the sea surface, and thus a suitable localization procedure is necessary.

This problem is often solved in literature using baseline techniques, such as long baseline (LBL), short baseline (SBL) and ultra-short baseline (USBL) [22] which belong to the acoustic time-of-flight techniques. These approaches typically need the presence of a set of communication devices which must be placed into the see (e.g. anchored to the seabed or attached to the hull of a ship) at a known position and allow to calculate the linear and longitudinal displacements of underwater vehicles within the network, with an accuracy which is dependent on the extension of the sensor network.

Even if baseline techniques can provide GPS-like capabilities, they need additional infrastructure and are geographically limited. To provide a general approach to localization, it is better to rely only sensors that can be equipped directly on the vehicles and to adopt state estimators exploiting sensor fusion algorithms; in this way localization can be performed using a reduced set of measurements, but can always be improved whenever baseline techniques are adopted. In literature the available approaches usually belong to one of these four categories: terrain-based navigation [9,30], simultaneous localization and mapping (SLAM) [36], deterministic state estimators [21] and stochastic model-based estimators. This last category includes optimal unbiased estimators which can be based both on kinematic or dynamic models and typically use all the available sensor data to perform sensor fusion. Among them the most used are the nonlinear Kalman filters.

The localization algorithm adopted in this paper is an adaptive extended Kalman filter (A-EKF) which is a variant of the EKF, in which the error covariance matrices are adapted online to reduce the errors due to linearization and prevent the divergence of the filter.

1.2 Formation control

The vehicles are required to follow a desire path keeping a formation geometry in order to properly perform a task. To solve this problem, several control approaches have been proposed in literature, ranging from classical solutions, such as proportional derivative (PD) controllers [18], particle filters [31] or Lyapunov-based controller [39], to more specialized ones, such as virtual bodies and artificial potentials (VBAP) [3,11] and behavioral control [2]. The approach adopted in this article and described in Sect. 4 is based instead on the nonlinear decentralized model predictive control (ND-MPC). The choice of the control algorithm follows from the analysis of the requirements that a formation control scheme for unmanned underwater vehicles should meet.

First of all it is important that the adopted control scheme can limit the information data exchanged between the vehicles as much as possible. This is necessary since the networks which can be realized underwater, such as ad hoc acoustic networks [13,32], have a limited bandwidth. For this reason decentralized solutions, in which there is limited communication among the vehicles and information is inferred about the other vehicle objectives by measuring their actions through their sensors, are to be preferred.

The main advantage of a decentralized control architecture is the lack of dependency of the whole system on a central processing unit: this characteristic grants more robustness, reduces the computational effort on the single agents and scales well to team size [33]. Moreover, the malfunctioning of a single vehicles will not affect the whole architecture, and local functions with faultless working vehicles continue to perform properly.

Finally, a formation control scheme should also be able to take into account constraints: speed of underwater vehicles (such as gliders) are often very limited, control efforts must be minimized to reduce the vehicles consumption and collision among vehicles are to avoid.

1.3 Fault tolerance

Furthermore, for vehicle formation to be effective in long-duration applications requiring an high operational autonomy, the coordination architecture is requested to be reliable. This means that such architecture must be able to handle the various difficulties of malfunctions, such as communication and vehicle faults, and other unexpected events. In this context, the fault tolerant strategy plays a crucial role. The success of a multi-vehicle mission depends on each vehicle operating in a fault-free manner. Fault tolerant for multi-vehicle formations is discussed in [1,8]. Faults in one vehicle can propagate to others over communication links, and this can cause problems in maintaining the formation and its shape to execute the desired tasks [14,15]. Formation of vehicles are both more fault tolerant (through redundancy) and more efficient (through parallelism) than single vehicles, if the vehicles are well coordinated. In a decentralized architecture, each vehicle can exchange sensor data information among neighbors and thus has the possibility to use this piece of information to recover from its faulty situation. In this way, the malfunctioning of a single vehicles will not affect the whole architecture, and working vehicles can complete the assigned tasks.

1.4 The modular architecture

Individually each of these major aspects do not allow an efficient formation coordination in real applications. For this reason there is a need for a modular coordination architecture which permits a correct and efficient organization of the information flow.

Using the extensive experience gained on vehicle positioning, ranging from early work on single vehicle to those on multi-vehicles [12,27,29,37,38], the authors in this paper propose a modular architecture for a fleet of underwater vehicles. The coordination of the fleet is based on the dominant and successful design approach of the leader–follower strategy, where one of the vehicle is designated as the leader, with the rest of the vehicle designated as followers. The proposed architecture is modular and each module is built such that it can solve a precise task using one or more functions. Three functions among them play a key role for the whole architecture: localization, faultless formation control and fault tolerance. The localization function is performed by the use of an A-EKF algorithm; the fault-free formation control function is based on a ND-MPC algorithm; the fault tolerance function is based on a hierarchy graph theory. The UUV team is controlled in a leader–follower manner, and the leader UUV is assigned by the Supervisor, where the followers are positioning themselves with respect to the other UUVs in the network. This architecture allows to solve the problem of formation coordination, even in case of fault, for a class of UUVs satisfying the architecture requirements. As illustrative example, this architecture has been applied for coordination of a fleet of gliders, and preliminary results show how the proposed solution performs.

The paper is organized as follows: the proposed architecture is introduced in Sect. 2, while its modules and functions are described in Sect. 3. Section 4 details the key functions of the architecture. Finally in Sect. 5 preliminary results are presented through an illustrative example. The paper ends with some concluding remarks and future developments in Sect. 6.

2 An architecture for UUVs coordination

The objective of this section is to introduce an architecture for coordinating UUVs formations. The proposed architecture subsumes leader-following approach and is based on two fundamental concepts: module and function.

Definition 1 A *"function"* is defined as a mathematical/logical data processing unit which can perform a limited number of operations.

Definition 2 A *"module"* is an abstract mathematical/logical unit which can perform a complex task (i.e. several operations) by using one or more functions within its scope.

Once a coordination problem has been planned, it can be decomposed into several tasks, each one carried out by one module. Each task can be further decomposed into several subsets of operations, each one performed by one function. In this way, a complex problem can be decomposed into smaller problems using a modular approach. Whenever a new function is needed, it can be added within the module without the need of modifying the whole architecture. The same applies to modules, which can be added without the need to re-design the part of architecture which does not make use of that module.

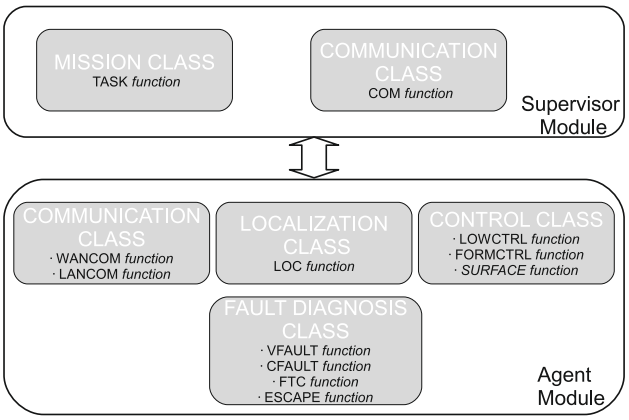


Fig. 1 Modules, classes and functions

Modules are typically associated with a specific type of vehicle and can only access functions designed for it. Functions can be found inside each module and are usually decomposed into classes according to their field of application.

The architecture presented in this paper introduces two modules: the "Supervisor Module", which can be both associated with a mobile marine base (such as a ship or boat equipped with all necessary instruments) or to a land control base, and the "Agent Module" which is associated with each unmanned underwater vehicle accomplishing the mission (see Fig. 1). Their relative functions are explained in details in Sects. 3 and 4.

Each underwater vehicle has the typical internal architecture of an underwater sensor node ([40], see Fig. 2) and it is supposed to be equipped with a typical sensor set for localization purposes, such as GPS, an attitude and heading reference system (AHRS), a depth sensor, sidescan and other sonars, magnetometers, and a different sensor set used for underwater exploration, such as thermistors and conductivity probes. Note that the specifications of these sensors do not affect the effectiveness of the proposed coordination architecture, as long as their performances can be considered reasonable according to the state of the art.

Node synchronization is much difficult to achieve in underwater wireless networks, and differ from terrestrial wireless networks since they have long and variable propagation delay and mobility. If the underwater node clocks are not synchronized, their transmission time will seem random and the propagation of data through the network will be slow. Several time synchronization algorithms have been

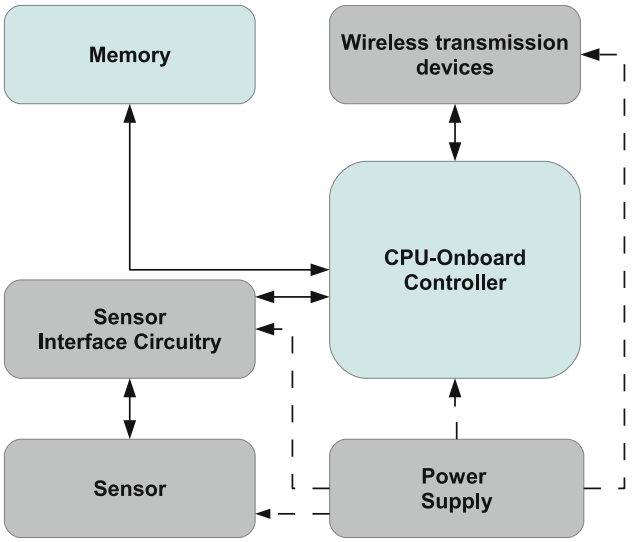


Fig. 2 Internal organization of an underwater sensor node

carried out to face this issue. In [23,26], time synchronization in underwater acoustic networks (UANs) has been considered. In [26], a three-dimensional, scalable UAN time synchronization scheme was proposed to achieve both horizontal and vertical clock synchronization to overcome the long propagation delay. In [28], a time synchronization method is presented. It not only compensates the main time delay in acoustic communication, but also decreases the cumulate errors in the multi-hop scene. The synchronization process is carried out while the node is broadcasting itself, so more energy is saved because of less control frame data exchange. Therefore, the time synchronization method has good performance in acoustic peer to peer networks, such as low power, high precision. The paper [26] also considers security issues about the time synchronization correlation test and statistical reputation trust model to detect outlier timestamps. In [23], effects of node movements on underwater time synchronization are considered. Because an underwater node can move out of and into another node's range frequently [23], no time synchronization is necessary if the timestamps of the received data packets are within the tolerance. In this way, the underwater network does not need to perform global time synchronizations periodically, which reduces the time used to synchronize clocks among sensor nodes. In [42] an energy efficient distributed time synchronization algorithm for underwater acoustic mobile sensor networks, called "E^2DTS" is presented. In the MU-Sync [6], the clock skew is estimated by performing the linear regression twice over a set of local time information gathered through message exchanges. The first linear regression enables the cluster head to offset the effect of long and varying propagation delay; the second regression in turn obtains the estimated skew and offset. With the help of MAC-level time stamping, nondeterministic errors that are commonly encountered by those synchronization algorithms that rely on message exchanges, are reduced.

In this paper, it is assumed that one of the time synchronization algorithm is established and fully working, such that clock drifts are compensated and the clock of each stationary or mobile underwater node results synchronized. The bandwidth requirements of the communication channel depends by the type of underwater transmission which has been chosen to adopt, such as acoustic waves, radio waves or optical waves. In any case, acoustic waves still remain the most promising mode for communicating underwater in application where tethering is unacceptable [22,25,34]. Typical bandwidths of the underwater acoustic channel for different ranges are reported in [40].

3 Modules and functions of the architecture

3.1 Supervisor module

In this section, the functions contained in the Supervisor Module are detailed. Please note that only the high level functions have been included in Fig. 1 and are explained in this section. All the low level functions, which implements the physical connection to sensors and the physical layers of the transmission networks, are omitted for brevity.

TASK function is the core function of the Supervisor Module. The *TASK* function allows the user to decide

- the leader trajectory,
- the relative displacement vectors (distance and relative headings among the vehicles in formation),
- additional information to be used for localization, formation control and fault tolerant procedure (e.g. velocity constraints, safe distance among vehicles, etc).

With this operation, the operator can choose among predefined settings or create a completely new formation using a graphical user interface. All the information is stored in a local database which can be accessed by the other functions in the supervisor module.
COM function can perform three operations:

- send the required mission information to the main leader agent module through the "Wide Area Communication" (see Fig. 3) using the wireless transmission device;
- receive data from the main leader to know the global position of the formation;
- receive data from a faulty vehicle in order to know its global position to recover it.

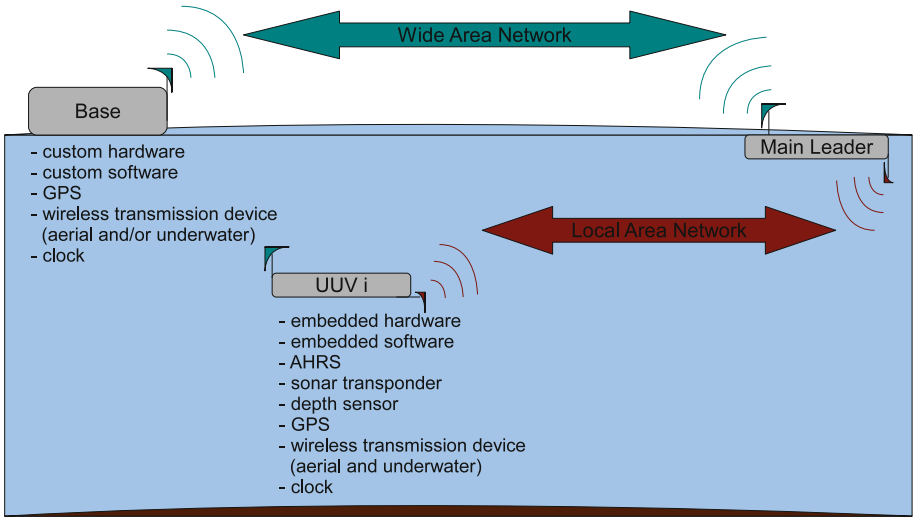


Fig. 3 Architecture layout

3.2 Agent module

In this section, the functions contained in the agent module are detailed. Please note that only the high level functions have been included in Fig. 1 and are explained in this section. Among the functions detailed in this section there are three functions that need a specific attention. These functions are the *LOC function*, which retrieve the localization information, the *FORMCTRL function*, which performs control formation when no fault is present and the *FTC function*, which allows to keep formation even in presence of fault. They represent the core of the agent module and are fully detailed in Sect. 4.

WANCOM function realizes the communication between the main leader agent module and the supervisor module. The wireless communication is typically performed using a wide area network (WAN) created by aerial radiofrequency devices. It can perform two operations:

- receive the required information sent by the supervisor module trough the WAN;
- send data to the supervisor module to communicate the updated global position of the formation.

This function is active when the vehicle is over the sea surface.

LANCOM function is fundamental to perform localization, formation control and fault tolerance maneuvers. It basically realizes the communication among the vehicles in formation: every time it is called by an agent module, it allows to synchronously exchange a limited set of data among all the other agent modules. The local network is typically realized using underwater acoustic devices (such as the Woods Hole Oceanographic Institution (WHOI) micro-modem [13]), according to protocols available in literature, such as [35], and is often referred as underwater wireless acoustic sensor network (UWASN) [4].

To improve the readability, in the rest of this paper, we use the term local area network (LAN) to indicate the local underwater network.

FORMCTRL function is one of the three core functions of the Agent Module as it represents the second step needed in order to solve the formation control problem: faultless formation control. This function requires the relative positions among the UUVs, the desired formation and the trajectory to follow to work correctly. When this information is available, then it returns the values of linear and angular speeds, in the horizontal plane, that each underwater vehicle must have. The adopted approach is based on network decentralized model predictive control (ND-MPC) and is detailed in Sect. 4.

LOWCTRL function implements the low level controller for each UUV. The proposed formation control algorithm returns the values of linear and angular speeds, in the horizontal plane, for each underwater vehicle, thus the low level controller must be capable to adjust the actuation forces for tracking those desired speeds. Moreover the low-level controller has the task to regulate the depth of the vehicle to a desired value (usually constant during navigation until the *SURFACE function* is called). The low level controller must also be robust to compensate drifts in case of unknown currents: this is typically achieved integrating a current estimator in the control loop [17]. The low level controller implementation depends on the type of UUV inside the formation and its implementation details are out of the scope of this paper.

SURFACE function checks the time elapsed from the last emersion. If this time exceeds a predetermined threshold then the depth of the main leader is set to zero. This value

is used by the *LOWCTRL function* to actually make the leader reach the surface.

LOC function is one of the three core functions of the Agent Module as it represents the first step needed in order to solve the formation control problem: localization. As said before, the formation control algorithm needs the position of the vehicles inside the formation in order to work properly. UUVs, however, can not be continuously localized using a GPS sensor since it is able to receive data from the satellites only when the vehicle is on the sea surface. Since absolute positions are not directly available, several sensors are employed to derive those measurements (see Fig. 3). To reduce the contribution of noise readings an A-EKF is implemented on the main leader and the information on localization are made available to all the vehicles along the mission by the use of the LAN. Each control agent has thus the information about position that can be used inside the ND-MPC algorithm (see Sect. 4).

VFAULT function implements a fault detection and isolation (FDI) algorithm for actuators and sensors of each vehicle. The *CFAULT function* implements a FDI algorithm for the communication system of each vehicle. In case of no fault, this function does not require any further action, however when a fault is detected and isolated the information is sent by the agent module of the faulty vehicle to the agent modules of the fault-free vehicles to call the *FTC function*, which performs a formation re-configuration according to the type of fault detected. The FDI system implementation depends on the type of UUV used inside the formation and on the faults which need to be detected and isolated.

FTC function is one of the three core functions of the agent module as it represents the third step needed to solve the formation control problem: formation re-configuration in case of fault. The fault-tolerance mechanism presented here is essentially based on the hierarchy graph theory of leader–follower formation. In case of unrecoverable fault on a leader vehicle, the local diagnostic system detects the fault and, when possible, the faulty vehicle broadcasts its faulty situation. After recognizing the faulty situation in the multi-vehicle formation, each follower reconfigures the controller for formation rearrangement. Faults in the network connections are recovered by changing the leaders and rerouting the information flow (for details, see Sect. 4). Finally, the faulty vehicle agent module executes the ESCAPE function to safely abandon the mission and reach the surface in order to be collected.

3.3 Workflow

This subsection describes the workflow of the presented architecture (see Fig. 4), where each block is labeled as *"ShortModuleName.Function"* and a short description of each function is provided.

Please note that each module executes functions with different sample times. The greatest sample time is that used by the EMERGE function which represents the time interval between surfacings (typically several minutes). The smallest sample time is that of *LOWCTRL function* (typically comparable to that of sensor readings) and all other sample times must be a multiple of this value. Only exception is made by the *TASK function* which is asynchronous: however this function can be actually performed only when the main leader is on the surface.

From the base station, the user chooses the coordination problem to solve and sets it using the *TASK function* of the supervisor module. This information is sent by the supervisor module to the main leader agent module using the *COM function*. The main leader agent module receives the signal with the *WANCOM function*, and communicates with all the other Agent modules using the *LANCOM function*: this establishes a continuous communication which is used by all the vehicles for all the duration of the mission.

When a vehicle is below the sea surface it measures at each sample time its depth and the relative distance to its leader. These measurements, together with the control efforts elaborated by the *FORMCTRL functions* of each Agent Module, are sent to the main leader using the LAN (this is done by the *LOC function* of each follower agent module). The main leader collects these data and uses them to update the predicted state and obtain a new estimate. This new estimate is used together with the control efforts to elaborate the predicted state at the next sample time, which is sent back to the follower agent modules using the LAN. The *FORMCTRL function* of each agent module uses the predicted state to calculate the new control efforts. These efforts are used by the *LOWCTRL function* to set the desired speeds.

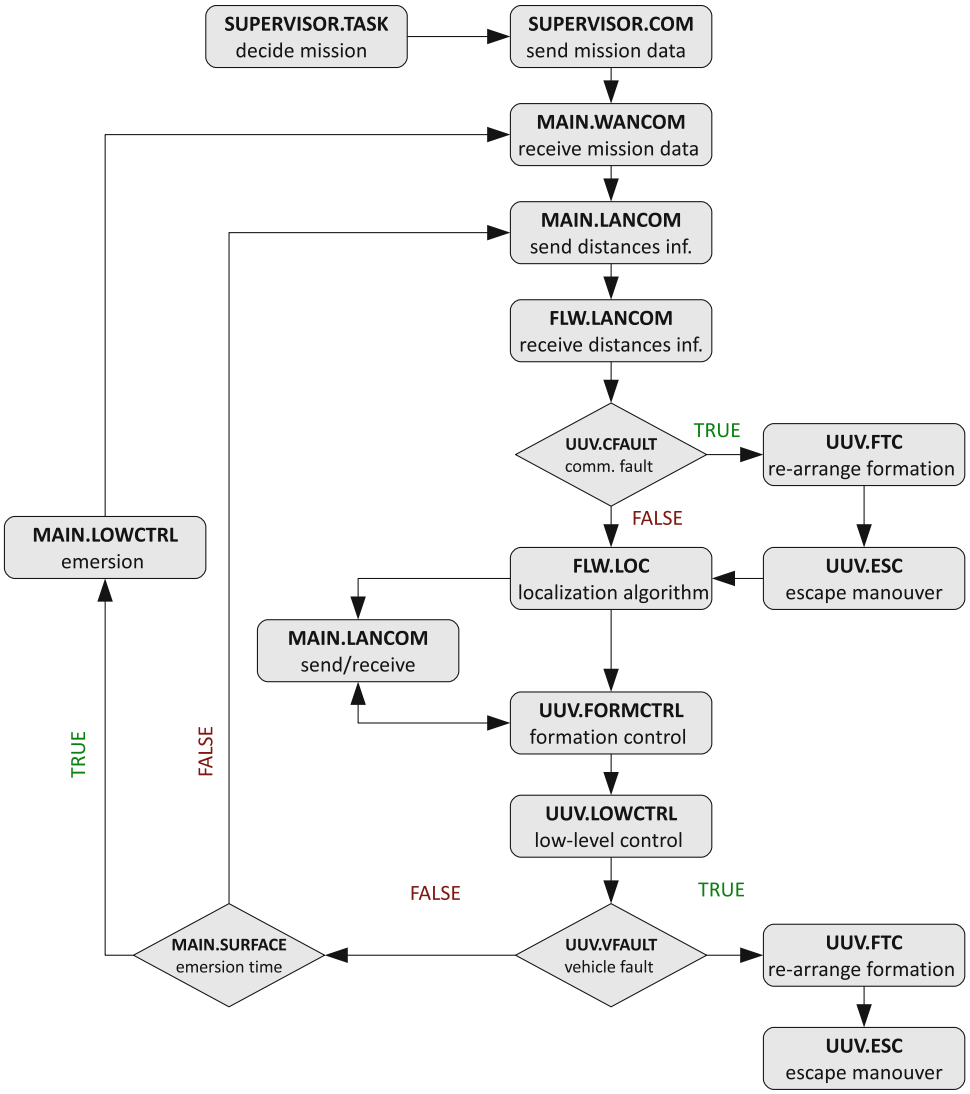
The architecture is realized such that the coordination can be formulated on demand: the main leader agent module executes the *SURFACE function* to set the desired depth of the formation to zero in order to reach the surface. When this happens the workflow can start from the beginning and a new coordination problem can be formulated. When the *SURFACE function* is not called, the mission proceeds as decided before and the loop is iterated from the MAIN.LANCOM block.

Finally, the UUV agent modules executes periodically the two fault detection functions: *CFAULT* and *VFAULT*. When a fault is detected the formation is re-arranged according to the algorithm described in the following Sect. 4.

4 Framework key functions

Before describing in detail the key functions of the proposed architecture, a formation vector model is introduced.

Fig. 4 Work flow chart



Afterwards, a full description of each of the key functions, namely localization function, control formation function and fault tolerant function, is provided.

Let consider a set of N underwater vehicles $\mathcal{V}^i$, $i = 1, \ldots, N$, that should accomplish the considered formation keeping task: the position of each leader vehicle with respect to the following vehicles should be kept equal to a desired value. Assume that at time t a low level controller imposes the desired surge, sway and yaw (angular) speeds $v^i(t), s^i(t)$ and $w^i(t)$ on the horizontal plane: in this way the high level controller has only the task to define the optimal speeds v, s, w that allow to keep the desired formation with the minimum possible efforts. Assume to sample the continuous-time variables with sampling interval T_s and define the sampled variables $v_k^i \triangleq T_s v^i(kT_s)$, $s_k^i \triangleq T_s s^i(kT_s)$, $w_k^i \triangleq T_s w^i(kT_s)$ that represents finite movements within each sampling interval T_s. These movements can also be seen as velocities normalized with respect to the sampling interval T_s and, in the following, they will be referred to as velocities.

Owing to physical limits surge, sway and angular velocities of the vehicle v_k^i, s_k^i, w_k^i are constrained and their limits depend on lower level controller and on the dynamic behavior of the vehicle. However, for the sake of simplicity, fixed constraints are assumed in the following:

$$\underline{v} \le v_k^i \le \overline{v}, \qquad \underline{s} \le s_k^i \le \overline{s}, \qquad |w_k^i| \le \overline{w}, \tag{1a}$$

$$|\Delta v_k^i| \le \overline{\Delta v}, \qquad |\Delta s_k^i| \le \overline{\Delta s}, \qquad |\Delta w_k^i| \le \overline{\Delta w}. \tag{1b}$$

where $(\underline{v}, \bar{v})$ are the linear velocity limits, the angular velocity is bounded by $\overline{w}$, while linear and angular velocity variations are limited to $\overline{\Delta v}$ and $\overline{\Delta w}$.

Defining rotation matrix $\mathbf{T}(\alpha) \triangleq \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$, the absolute configuration of vehicle i, on the horizontal plane and with respect to the earth frame, $\mathbf{q}_k^i \triangleq [q_{x,k}^i \; q_{y,k}^i \; q_{\theta,k}^i]^T$ is determined by integrating the control action $\mathbf{u}_k^i \triangleq [v_k^i \; s_k^i \; w_k^i]^T$ by the following discrete-time kinematic model:

$$\mathbf{q}_{k+1}^i = \mathbf{q}_k^i + \mathbf{T}^{-1}(q_{\theta,k}^i)\mathbf{u}_k^i. \tag{2}$$

Referred to the frame fixed to vehicle $\mathcal{V}^i$, the relative displacement of vehicle $\mathcal{V}^j$, $\mathbf{d}_k^{ji} \triangleq [x_k^{ji} \;\; y_k^{ji} \;\; \theta_k^{ji}]^T = \mathbf{T}(q_{\theta,k}^i)(\mathbf{q}_k^j - \mathbf{q}_k^i)$ gives the following discrete-time formation vector model (see [37])

$$\mathbf{d}_{k+1}^{ji} = \mathbf{A}_k^i \mathbf{d}_k^{ji} + \mathbf{B}_k^i \mathbf{u}_k^i + \mathbf{E}_k^{ji} \mathbf{u}_k^j, \tag{3}$$

where

$$\mathbf{A}_k^i \triangleq \mathbf{T}(w_k^i),\;\; \mathbf{B}_k^i \triangleq -\mathbf{T}(w_k^i),\;\; \mathbf{E}_k^{ji} \triangleq \mathbf{T}(w_k^i)\mathbf{T}^{-1}(\theta_k^{ji}). \tag{4}$$

4.1 *LOC function*: algorithm description

To perform formation control the most straightforward approach would be to measure the absolute position of each underwater vehicle and use that information to keep the relative distances at the desired values. This approach, however, can not be used for an UUV as the GPS sensor is able to receive data from the satellites only when it is on the sea surface, and thus a localization procedure is necessary. Even if baseline techniques can provide GPS-like capabilities, they need additional infrastructure, thus an A-EKF which uses the measurements of heading and distance between vehicles is preferred. The A-EKF presented here is a variant of the EKF in which the error covariance matrices are adapted on-line. Since the filter is applied to a nonlinear system an adaptive version is adopted in order to reduce the errors due to linearization and limit the divergence.

4.1.1 Prediction

Consider the kinematic model described by (2) and assume that an additive, zero-mean gaussian white noise is affecting the system. Defining

$$\boldsymbol{L}_k \triangleq \begin{bmatrix} \boldsymbol{L}_k^1 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & & 0 \\ 0 & \dots & 0 & \boldsymbol{L}_k^n \end{bmatrix},$$

with

$$\boldsymbol{L}_k^i \triangleq \begin{bmatrix} T_s \cos \hat{q}_{\theta,k|k}^i & -T_s \sin \hat{q}_{\theta,k|k}^i & -\frac{T_s^2}{2} v_{k-1}^i \sin \hat{q}_{\theta,k|k}^i - \frac{T_s^2}{2} s_{k-1}^i \cos \hat{q}_{\theta,k|k}^i \\ T_s \sin \hat{q}_{\theta,k|k}^i & T_s \sin \hat{q}_{\theta,k|k}^i & \frac{T_s^2}{2} v_{k-1}^i \cos \hat{q}_{\theta,k|k}^i - \frac{T_s^2}{2} s_{k-1}^i \sin \hat{q}_{\theta,k|k}^i \\ 0 & 0 & T_s \end{bmatrix},$$

the state prediction equation for the entire formation, linearized around the working points $\boldsymbol{q}_{0,k} = \hat{\boldsymbol{q}}_{k|k}$ and $\boldsymbol{u}_{0,k} = \boldsymbol{u}_{k-1}$, is

$$\hat{\boldsymbol{q}}_{k+1|k} = \hat{\boldsymbol{q}}_{k|k} + \boldsymbol{L}_k \boldsymbol{u}_k. \tag{5}$$

4.1.2 Update

Consider the measurement equation described by

$$\boldsymbol{z}_{k+1} = \boldsymbol{G}(\boldsymbol{q}_{k+1}) + \boldsymbol{V}_{k+1}, \tag{6}$$

where $\boldsymbol{q}_{k+1}$ and $\boldsymbol{z}_{k+1}$ represent the state vector and the measurement vector at time $(k+1)T_s$, $\boldsymbol{V}_{k+1}$ is the noise vector affecting measurements which is white, zero-mean gaussian and with a covariance matrix $\boldsymbol{R}_{k+1}$ which depends on the sensor used.

Let assume that vector $\boldsymbol{G}_f$ describes the formation geometry, i.e.

$$\boldsymbol{G}_f = \big[l_1, l_2, \dots, l_n\big]^T, \qquad l_1 = 0,$$

where l_i indicates that $\mathcal{V}^i$ has l_i as its leader. l_1 is set to 0 since the main leader has the virtual vehicle $\mathcal{V}^0$ as leader. With this notation the measurement equation can be written as

$$\boldsymbol{G}(\boldsymbol{q}_{k+1}) = \Big[g_{k+1}^{l_2,2}, g_{k+1}^{l_3,3}, \dots, g_{k+1}^{l_n,n}, q_{\theta,k+1}^1, q_{\theta,k+1}^2, \dots, q_{\theta,k+1}^n\Big]^T,$$

with

$$g_{k+1}^{l_i,i} = \sqrt{\big(q_{x,k+1}^{l_i} - q_{x,k+1}^i\big)^2 + \big(q_{y,k+1}^{l_i} - q_{y,k+1}^i\big)^2} \qquad i = 2, \dots, n.$$

Linearizing the measurement equation around the working point $\boldsymbol{q}_{k+1} = \hat{\boldsymbol{q}}_{k+1|k}$ and defining

$$\boldsymbol{A}_{d,k} \triangleq \begin{bmatrix} \boldsymbol{A}_{d,k}^1 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & & 0 \\ 0 & \dots & 0 & \boldsymbol{A}_{d,k}^n \end{bmatrix},$$

with

$$\boldsymbol{A}_{d,k}^i \triangleq \begin{bmatrix} 1 & 0 & -T v_{k-1}^i \sin \hat{q}_{\theta,k|k}^i - T s_{k-1}^i \cos \hat{q}_{\theta,k|k}^i \\ 0 & 1 & T v_{k-1}^i \cos \hat{q}_{\theta,k|k}^i - T s_{k-1}^i \sin \hat{q}_{\theta,k|k}^i \\ 0 & 0 & 1 \end{bmatrix},$$

the update equation can be described by

$$\hat{\boldsymbol{q}}_{k+1|k+1} = \hat{\boldsymbol{q}}_{k+1|k} + \boldsymbol{K}_{k+1}\Big[\boldsymbol{z}_{k+1} - \boldsymbol{G}\big(\hat{\boldsymbol{q}}_{k+1|k}\big)\Big] \tag{7}$$

The gain matrix and prediction matrix are defined as

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1|k}\boldsymbol{C}_{k+1}^T\big[\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1|k}\boldsymbol{C}_{k+1}^T + \boldsymbol{R}_{k+1}\big]^{-1} \tag{8}$$

$$\boldsymbol{P}_{k+1|k+1} = \big[\boldsymbol{I} - \boldsymbol{K}_{k+1}\boldsymbol{C}_{k+1}\big]\boldsymbol{P}_{k+1|k} \tag{9}$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{A}_{d,k}\boldsymbol{P}_{k|k}\boldsymbol{A}_{d,k}^T + \boldsymbol{Q}_{d,k} \tag{10}$$

where $\boldsymbol{Q}_{d,k}$ is the covariance matrix of noise affecting the states and $\boldsymbol{C}_{k+1}$ is defined by

$$\boldsymbol{C}_{k+1} = \left.\frac{\partial \boldsymbol{G}\big(\boldsymbol{q}(k+1)\big)}{\partial \boldsymbol{q}(k+1)}\right|_{\boldsymbol{q}(k+1)=\hat{\boldsymbol{q}}(k+1,k)}. \tag{11}$$

4.1.3 Adaptive estimation of $Q_{d,k}$ and R_{k+1}

The adaptive procedure presented here can be used whenever matrices $\boldsymbol{Q}_{d,k}$ and $\boldsymbol{R}_{k+1}$ have the form

$$\boldsymbol{Q}_{d,k} = \sigma_{\eta,k}^2 \boldsymbol{Q}_k \tag{12}$$

$$\boldsymbol{R}_{k+1} = \text{diag}[\sigma_{v,i,k+1}^2] \quad i = 1, \ldots, p \tag{13}$$

where $\sigma_{v,i}$ and σ_η are the parameters to be adapted. Equation (12) is valid whenever the kinematic model describes the state dynamics with statistically independent errors. Equation (13) is valid whenever the measurements are statistically independent.

Assume that $\gamma_{i,k+1} = z_{i,k+1} - G_i(\hat{q}_{k+1|k})$: similarly to the linear case, it represents the components of the innovation process at time $k+1$. Using the procedure described in [20], we can write

$$\hat{\sigma}_{\eta,i,k}^2 = \left(\boldsymbol{c}_i(k+1)\boldsymbol{Q}_k\boldsymbol{c}_i^T(k+1)\right)^{-1} \max\left\{\gamma_{i,k+1}^2 - \bar{\hat{\sigma}}_{v,i,k+1}^2 - \boldsymbol{c}_i(k+1)\boldsymbol{A}_{d,k}\boldsymbol{P}_{k|k}\boldsymbol{A}_{d,k}^T\boldsymbol{c}_i^T(k+1), 0\right\} \tag{14}$$

and

$$\hat{\sigma}_{v,i,k+1}^2 = \max\left\{\gamma_{i,k+1}^2 - \left[\boldsymbol{c}_i(k+1)\boldsymbol{A}_{d,k}\boldsymbol{P}_{k|k}\boldsymbol{A}_{d,k}^T\boldsymbol{c}_i^T(k+1) + \boldsymbol{c}_i(k+1)\bar{\hat{\sigma}}_{\eta,i,k}^2\boldsymbol{Q}_k\boldsymbol{c}_i^T(k+1)\right], 0\right\} \tag{15}$$

where $\boldsymbol{c}_i(\cdot)$ is the ith row of matrix $\boldsymbol{C}(\cdot)$ and $\bar{\hat{\sigma}}_{\eta,i}$ and $\bar{\hat{\sigma}}_{v,i}$ represent the smoothed estimates calculated as the average value among the last l_η and l_v values (with l_η and l_v the last number of measurements of $\hat{\sigma}_\eta^2(\cdot)$ and $\hat{\sigma}_{v,i}^2(\cdot)$ respectively, to be chosen a priori), i.e.

$$\bar{\hat{\sigma}}_\eta^2(k) = \frac{1}{(l_\eta+1)p}\sum_{j=0}^{l_\eta}\sum_{i=1}^{p}\hat{\sigma}_{\eta,i}^2(k-j), \tag{16}$$

and

$$\bar{\hat{\sigma}}_{v,i}^2(k+1) = \frac{1}{l_v+1}\sum_{j=0}^{l_v}\hat{\sigma}_{v,i}^2(k+1-j). \tag{17}$$

The adaptive algorithm reduces the probability of divergence. One easy way to prevent divergence at all is to keep track of the difference between the real distances among vehicles and the desired ones: when these values exceed a desired threshold, then it is necessary to make the fleet emerge to initialize again the algorithm.

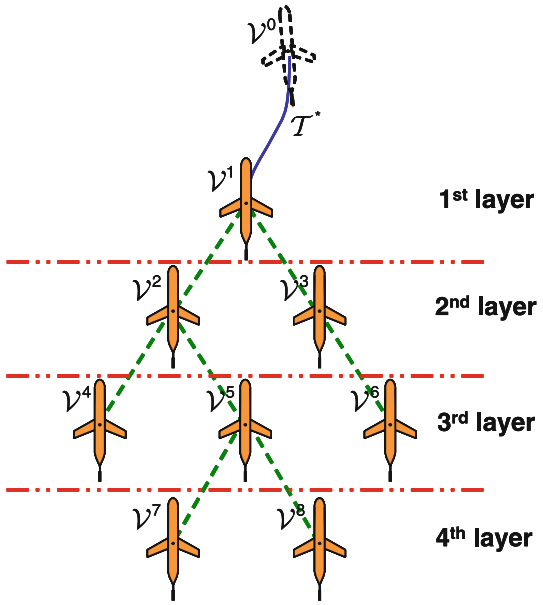


Fig. 5 The considered leader–follower architecture

4.2 *FORMCTRL function*: algorithm description

The *FORMCTRL function* is devoted to the formation control of the underwater fleet. In this context, the formation control problem is formulated as a cascaded leader–follower problem (see Fig. 5) where

- The reference trajectory $\mathcal{T}^*$ is generated by a virtual reference vehicle $\mathcal{V}^0$ which moves according to the considered unicycle model.
- Each vehicle $\mathcal{V}^i$ follows one and only one leader $\mathcal{V}^j$, $j \neq i$; $\mathcal{V}^1$ follows virtual vehicle $\mathcal{V}^0$ which exactly tracks the reference trajectory $\mathcal{T}^*$.
- Each vehicle $\mathcal{V}^i$ should keep the reference formation pattern $\bar{\mathbf{d}}^{ji}$, from its leader $\mathcal{V}^j$.

In this framework, the multi-vehicle formation can be seen as a directed tree that can be formally expressed by the notation of the directed graphs. Let us denote with the tuple $\mathcal{G} \triangleq \{\mathcal{V}, \mathcal{E}\}$ the digraph with nodes $\mathcal{V} \triangleq \{\mathcal{V}^1, \ldots, \mathcal{V}^n\}$ and edges $\mathcal{E} \triangleq \{\mathcal{E}^1, \ldots, \mathcal{E}^m\}$, in which each edge is an ordered pair of nodes $\mathcal{E}^l = (\mathcal{V}^j, \mathcal{V}^i)$ establishing a link from $\mathcal{V}^j$ to $\mathcal{V}^i$. Each node corresponds to a real vehicle and the direction of the edge goes from the leader to the corresponding follower, in agreement with the information flow. The hierarchy of the formation and dependencies among vehicles are represented as a hierarchy graph.

In a general leader–follower formation, it is possible to distinguish between leaders and followers, but the leader of the whole formation is unique. The formation can be decomposed into several layers, depending on the member's level in

the formation, as shown in Fig. 5. In this way, any member of the formation is a node in the hierarchy graph. The first layer has only one node; namely the father node, which represents the leader of the whole formation and the second layer could have one or different local nodes which are child nodes of the father node, and so on. The direct connections occur among father and local nodes, or local and child nodes. Note that there are not connections among nodes of the same layer, but connections are possible only from nodes of different layers. This analysis is valid also in the event of a different formation shape. In effect, all connections are logical, indicating that the formation is not fixed, and nodes in the same layer need not be parallel.

The adjacency matrix $\mathbf{A}$ of digraph $\mathcal{G}$ has $n \times n$ entries $\mathbf{a}_{ij} = 1$ iff $(\mathcal{V}^i, \mathcal{V}^j) \in \mathcal{E}$ and $\mathbf{a}_{ij} = 0$ otherwise. Distance matrix $\mathbf{D}$ of digraph $\mathcal{G}$ has $n \times n$ entries $\mathbf{d}_{ij} = \bar{d}^{ji}$ iff $(\mathcal{V}^i, \mathcal{V}^j) \in \mathcal{E}$ and $\mathcal{V}^i$ has to keep distance $\bar{d}^{ji}$ with respect to $\mathcal{V}^j$, while $\mathbf{d}_{ij} = 0$ otherwise. The incidence matrix $\mathbf{C}$ of digraph $\mathcal{G}$ has $n \times m$ entries $\mathbf{c}_{il} = 1$ iff edge $\mathcal{E}^l$ exits from node $\mathcal{V}^i$, $\mathbf{c}_{il} = -1$ iff edge $\mathcal{E}^l$ enters in node $\mathcal{V}^i$ and $\mathbf{c}_{il} = 0$ otherwise. Formation structure can be completely described by the graph, its adjacency matrix and its incidence matrix whereas formation geometry is described by distance matrices $\mathbf{D}_x, \mathbf{D}_y, \mathbf{D}_\theta$.

The formation control problem is decomposed into an inner-loop dynamic task [12], which consists of making the vehicle's velocity track a reference one (performed by the *LOWCTRL function* and not reported here), and an outer-loop kinematic task, which assigns the reference speed for tracking a desired trajectory.

The *FORMCTRL function* is based on a networked decentralized MPC algorithm. In the developed navigation system, each underwater vehicle $\mathcal{V}^i$ is equipped with an independent control agent $\mathcal{A}^i$ which collects local and remote information (i.e. the position vector provided by the A-EKF) and iteratively performs a nonlinear optimization for computing the local control action. As previously stated, each vehicle $\mathcal{V}^i$ tracks a leader $\mathcal{V}^j$ with a defined displacement. The set of all displacements defines the formation (Fig. 5).

In the implementation of the proposed navigation system the following assumptions are made

- Each control agent $\mathcal{A}^i$ communicates with its neighboring agents by a LAN only once within a sampling interval.
- The communication network introduces a delay $\tau = 1$.
- The agents are synchronous.
- Each control agent is able to measure the relative configurations of the neighbors.

The drifts of the different clocks are very slow and each agent $\mathcal{A}^i$ synchronizes its own clock with the clock of its leader agent $\mathcal{A}^j$. Each leader also broadcasts to its followers the predictions about its future behavior. Finally, the main leader broadcasts the information relative to the position vectors of the UUVs provided by the centralized A-EKF.

The following scalar is considered here as a measure of the performance for control agent $\mathcal{A}^i$:

$$\langle \mathbf{d}_k^{ji} - \bar{\mathbf{d}}^{ji} \rangle^2 \triangleq \rho_x (x_k^{ji} - \bar{x}^{ji})^2 + \rho_y (y_k^{ji} - \bar{y}^{ji})^2 + \rho_\theta \sin^2 \frac{\theta_k^{ji} - \bar{\theta}^{ji}}{2} \tag{18}$$

where $\bar{\mathbf{d}}^{ji}$ is the constant desired displacement and ρ_x, ρ_y, ρ_θ are arbitrary weights. The cost function to be minimized is:

$$J_k^i = \sum_{h=1}^{p} \langle \hat{\mathbf{d}}_{k+h|k}^{ji} - \bar{\mathbf{d}}^{ji} \rangle^2 + \mu |\hat{\mathbf{u}}_{k+h-1|k}^i|^2 + \sigma |\Delta \hat{\mathbf{u}}_{k+h-1|k}^i|^2 + \eta \sum_{h=1}^{p-1} |\hat{\mathbf{u}}_{k+h-1|k}^i - \hat{\mathbf{u}}_{k+h-1|k-1}^i|^2 \tag{19}$$

where μ, σ and η are weights whose meaning is described in [27], $\hat{\boldsymbol{u}}_{k|h}^j$ is the jth control agent predicted control effort at time h for time k and $\hat{\boldsymbol{d}}_k^{ji}$ is the displacement vector calculated using the positions provided by the A-EKF. The predicted control efforts are necessary since the decentralized solution here proposed implies that the interaction vector $\mathbf{u}_k^j$ is unknown to the local control agent $\mathcal{A}^i$. This information is shared using the LAN.

The above-nonlinear constrained optimization problem is iteratively set-up and solved at each sample time by proper minimization algorithms and allow to compute the control efforts $\boldsymbol{u}_k$.

The solution here described grants both stability and collision-free properties:

Stability. Consider the set $\mathcal{V}$ of all vehicles $\{\mathcal{V}^i, i = \ldots, N\}$ with structure (3). If for each vehicle $\mathcal{V}^i$ with leader $\mathcal{V}^{\mathcal{L}_i}$, its independent agent $\mathcal{A}^i$ minimizes the cost function (19) under the following constraint

$$\langle \hat{\mathbf{d}}_{k+p|k}^{\mathcal{L}_i,i}(\mathbf{u}_{\cdot|k-1}^{i*}, \hat{\mathbf{u}}_{k+p-1|k}^i) - \bar{\mathbf{d}}^{\mathcal{L}_i,i} \rangle^2 + \mu |\hat{\mathbf{u}}_{k+p-1|k}^i|^2 + \sigma |\Delta \hat{\mathbf{u}}_{k+p-1|k}^i|^2 \leq r_k^i \tag{20}$$

where r_k^i is known at time k and defined as

$$r_k^i \triangleq \langle \mathbf{d}_k^{\mathcal{L}_i,i} - \bar{\mathbf{d}}^{\mathcal{L}_i,i} \rangle^2 + \mu |\mathbf{u}_{k-1}^i|^2 + \sigma |\Delta \mathbf{u}_{k-1}^i|^2 - \sum_{h=1}^{p} \Big[\langle \hat{\mathbf{d}}_{k+h-1|k}^{\mathcal{L}_i,i}(\mathbf{u}_{\cdot|k-1}^{i*}) - \bar{\mathbf{d}}^{\mathcal{L}_i,i} \rangle^2 - \langle \hat{\mathbf{d}}_{k+h-1|k-1}^{\mathcal{L}_i,i}(\mathbf{u}_{\cdot|k-1}^{i*}) - \bar{\mathbf{d}}^{\mathcal{L}_i,i} \rangle^2 \Big] \tag{21}$$

than the set $\mathcal{A}$ of control agents $\mathcal{A}^i$, $i = 1, \ldots, N$, guarantees the local stability of the equilibrium point $\bar{\mathbf{d}} \triangleq [(\bar{\mathbf{d}}^{\mathcal{L}_1,1})^T, \ldots, (\bar{\mathbf{d}}^{\mathcal{L}_N,N})^T]^T$ for the whole closed-loop system. Further details and proof are provided in [37].

Collision-free. If the following constraints are satisfied at each sample time k:

$$|\boldsymbol{N}\left(\boldsymbol{A}_k^i \boldsymbol{d}_k^{li} + \boldsymbol{B}_k^i \boldsymbol{u}_k^i\right)| \geq \underline{d} + \sqrt{2}\max\left(|\underline{v}|, |\underline{s}|, |\overline{v}|, |\overline{s}|\right), \quad l = 1, \ldots, n \quad l \neq i. \tag{22}$$

than the collision-free property is guaranteed.

4.3 *FTC function*: algorithm description

The decentralized FDI enables the set $\mathcal{V}$ of vehicles to continue to complete given tasks by reorganizing their formation, when some faults occur. Two kinds of faults are possible: *"communication faults"* and *"vehicle faults"* [5].

When a *communication fault* occurs in a vehicle, the leader or the follower of that vehicle will lose connectivity and no information is exchanged. The leader or follower of that vehicle will try to reconnect the faulty vehicle and, after a fixed time delay, if the vehicle does not reply, the other vehicles know that such vehicle is into communication fault. The maximum time delay is fixed in the prediction horizon p, and until the faulty vehicle is not reconnected, the predictions on its future behavior are obtained from the previous data. If vehicle $\mathcal{V}^j$ does not reply for a time delay $1 \leq \tau \leq p-1$, the prediction of the interaction is used by the follower vehicle to predict its future behavior.

In a *vehicle fault*, the vehicle FDI and control system will try to recover the fault. In this situation, the vehicle broadcasts its current position and faulty information to the others vehicles of the fleet. In such situation, all other vehicles of the fleet know which vehicle is into vehicle fault.

The fault tolerance algorithm for a fleet of underwater vehicles can be described as follows. After a communication/vehicle fault has been detected, the FDI system of the faulty vehicle tries to recover the fault (this operations are performed by the *VFAULT* and *CFAULT functions*). If this is possible, the vehicles rearrange the fleet in the original shape. In case of unrecoverable fault, the faulty vehicle moves away from the formation through an escape maneuver (ESC function), and the remaining vehicles substitute the faulty vehicle and rearrange the fleet obeying to some pre-assigned formation shape, depending on the number of remaining vehicles (this is actually performed by the *FTC function*). The escape maneuver is always possible in open sea where available space is not an issue: the collision avoidance policy included in the ND-MPC control law allows the fault-free vehicles to avoid impact with the faulty vehicle, thus the maneuver which makes the faulty vehicle emerge is achievable.

The substitute of the faulty vehicle must be one of its child nodes in the fleet hierarchy graph. Fault-free vehicle which takes the place of the faulty one is determined by two priority rules [5]. For each node, the following priority rules hold:

1. the number of total follower vehicle nodes is larger, while the priority is lower;
2. the pre-assigned sequence number in the hierarchy is smaller, while the priority is higher.

In this way, if by the first rule it is impossible to determine the substitute, than this is univocally chosen through the second rule. Note that the rank of the first rule is higher than that of the second rule.

5 Preliminary results

The proposed architecture can be applied to a wide class of underwater vehicles and in this section it is successfully used to solve a formation coordination problem for an underwater glider fleet. The gliders are underactuated marine vehicles that are difficult to maneuver, and for this reason they represent a challenging research topic. On the other hand, underwater gliders are extremely energy efficient and have already demonstrated high endurance, making them very attractive for oceanographic surveys requiring long-term deployment and autonomous operation. For this reason, there is a big interest and necessity to control and coordinate a fleet of them. The formation control law operates using the kinematic model, and the dynamic regulation of each single vehicle is demanded to the low level controller. In this case, we suppose that the glider low level controller is capable of tracking the desired velocities on the plane of motion (horizontal plane), thus the formation control only needs to provide the suitable velocities to reach and maintain the desired formation.

The developed strategy is here tested on the formation control of a fleet composed by $N = 5$ gliders initially positioned as follows:

$$\mathbf{q}_0^1 = \begin{bmatrix} 23 \\ 2 \\ \frac{\pi}{2} \end{bmatrix} \mathbf{q}_0^2 = \begin{bmatrix} 19 \\ -\frac{1}{2} \\ \frac{3\pi}{4} \end{bmatrix} \mathbf{q}_0^3 = \begin{bmatrix} 24 \\ -1 \\ \frac{\pi}{4} \end{bmatrix} \mathbf{q}_0^4 = \begin{bmatrix} 22 \\ 0 \\ \frac{4\pi}{3} \end{bmatrix} \mathbf{q}_0^5 = \begin{bmatrix} 21 \\ -2 \\ \frac{\pi}{3} \end{bmatrix} \tag{23}$$

Simulation has been implemented for 121 samples for virtual vehicle $\mathcal{V}^0$ that follows an 'S' shaped path and tuning parameters as in Table 1. Three faults have been introduced into the simulation:

Table 1 Simulation parameters

Parameter	Value	Parameter	Value
p	5	$\underline{v}$	0
$\overline{w}$	$\pi/3$	$\overline{v}$	1
$\overline{\Delta w}$	$2\pi/9$	$\overline{\Delta v}$	7/20
ρ_x	1	σ	1
ρ_y	1	η	2/5
ρ_θ	1	μ	2/5

f_1 : the first fault affects the communication system of the vehicle $\mathcal{V}^2$ at time $k = 15$ for 5 samples, thus $\mathcal{V}^2$ does not broadcast predictions for 5 samples;

f_2 : the second fault affects low level control of the vehicle $\mathcal{V}^1$ at time $k = 55$ for 25 samples, thus $\mathcal{V}^1$ does not move anymore and broadcasts faulty state.

f_3 : the third fault affects low level control of the vehicle $\mathcal{V}^3$ in unrecoverable manner, $\mathcal{V}^3$ does not move anymore and broadcast faulty state.

The obtained results are reported in Fig. 6. The vehicles and their performed trajectories are drawn with different colors and the fleet configurations have been frozen at the most significant conditions.

Fault (f_1) implies that vehicle $\mathcal{V}^2$ does not broadcast predictions for 5 samples and vehicle $\mathcal{V}^4$ uses the last received predictions to track it: its effect is not visible and this shows the high tolerance of the proposed strategy to communication faults which stay within prediction horizon.

Fault (f_2) involves the low level controller of vehicle $\mathcal{V}^1$ that stops at time $k = 55$ for a number of samples greater than $p = 5$: in this case formation is rearranged at sample $k = 60$. Based on priority rule (1), vehicles $\mathcal{V}^2$ and $\mathcal{V}^3$ are the highest in the hierarchy but priority rule (2) establishes that vehicle $\mathcal{V}^2$ has the highest priority, therefore $\mathcal{V}^2$ takes place of the main leader $\mathcal{V}^1$. Graph matrices are then transformed into:

$$\mathbf{A} = \begin{bmatrix} 0&0&0&0&0\\ 0&0&1&1&0\\ 0&0&0&0&1\\ 1&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0&0&-1&0\\ 1&1&0&0\\ 0&-1&0&1\\ -1&0&1&0\\ 0&0&0&-1 \end{bmatrix} \tag{24a}$$

$$\mathbf{D}_x = \begin{bmatrix} 0&0&0&0&0\\ 0&0&-1&-1&0\\ 0&0&0&0&-1\\ -1&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix}, \quad \mathbf{D}_y = \begin{bmatrix} 0&0&0&0&0\\ 0&0&-1&1&0\\ 0&0&0&0&-1\\ 1&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix}. \tag{24b}$$

When $\mathcal{V}^1$ recovers, it automatically restarts to keep the formation with respect to its new leader $\mathcal{V}^4$.

Unrecoverable fault (f_3) damages low level controller of vehicle $\mathcal{V}^3$ that stops at time $k = 95$ for a number of samples greater than $p = 5$: in this case formation is rearranged at sample $k = 100$. Based on first priority rule (i.), vehicles $\mathcal{V}^5$ has the highest priority and it takes place of its leader $\mathcal{V}^3$. Graph matrices are then transformed into:

$$\mathbf{A} = \begin{bmatrix} 0&0&0&0&0\\ 0&0&0&1&1\\ 0&0&0&0&0\\ 1&0&0&0&0\\ 0&0&1&0&0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0&0&-1&0\\ 1&1&0&0\\ 0&0&0&-1\\ -1&0&1&0\\ 0&-1&0&1 \end{bmatrix} \tag{25a}$$

$$\mathbf{D}_x = \begin{bmatrix} 0&0&0&0&0\\ 0&0&0&-1&-1\\ 0&0&0&0&0\\ -1&0&0&0&0\\ 0&0&-1&0&0 \end{bmatrix}, \mathbf{D}_y = \begin{bmatrix} 0&0&0&0&0\\ 0&0&0&1&-1\\ 0&0&0&0&0\\ 1&0&0&0&0\\ 0&0&-1&0&0 \end{bmatrix}. \tag{25b}$$

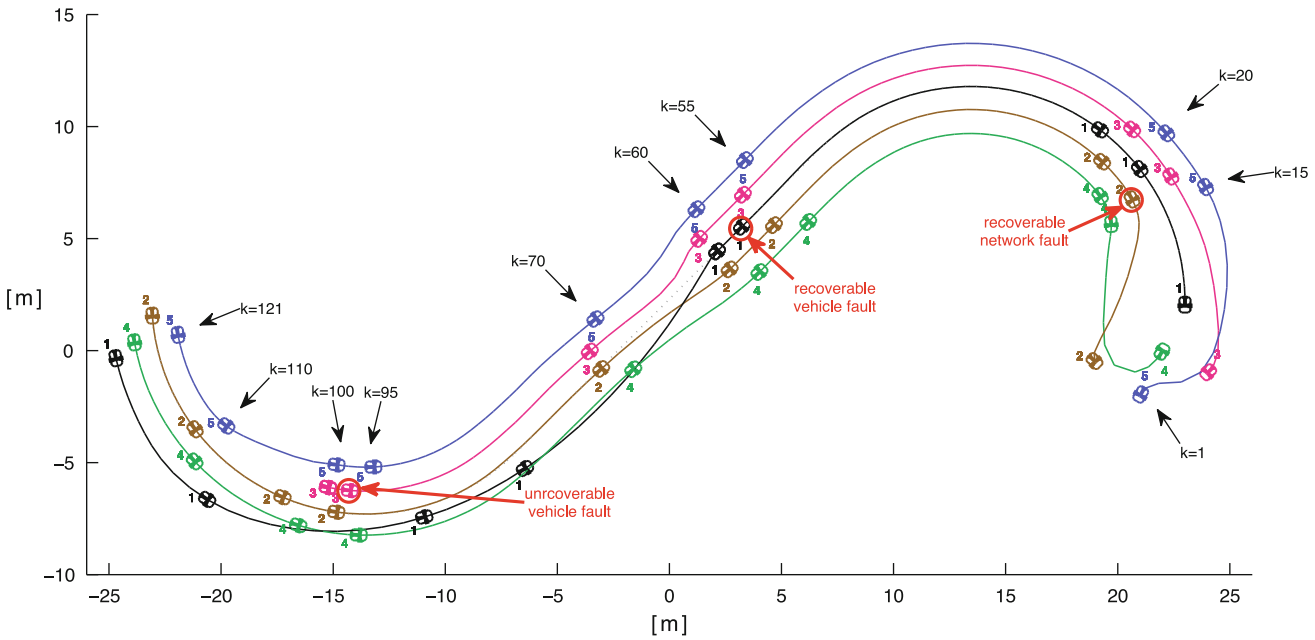


Fig. 6 Trajectories followed by the five vehicles, where each vehicle is identified by a different color

It can be seen that the fleet always reaches the desired formation in 10–15 samples after each formation rearrangement, and the task is accomplished even in the case of several faults.

The described algorithm has been implemented in Matlab®. The optimization has been performed by `fmincon` function that implements a sequential quadratic programming algorithm that iteratively solves QP sub-problems by means of the active set strategy and a positive quasi-Newton approximation of the Hessian of the Lagrangian. By compiling and optimizing the Matlab code and limiting the maximum number of iterations, the computational requirements can be significantly reduced thus making the algorithm online implementable.

6 Conclusions and future works

In this paper, an approach to solve the problem of coordinating UUV fleets is proposed based on a modular and expandable decentralized architecture. The problem is decomposed into tasks which can be solved by the functions introduced in the modules of the proposed architecture. Among the functions detailed in the paper, three of them play a key role for the whole architecture: the localization function, the formation control function and the fault tolerant function. They actually realize a decentralized fault tolerant formation control whenever all the modules and functions of the proposed architecture are working. Simulation results show the effectiveness of the proposed solution for a special class of underwater vehicles.

The main advantages of the presented solution are modularity and its wide applicability. Modularity allows to increase the number of tasks that the architecture can cope with, without modifying it from the core, and this is possible thanks to the decentralization policy. Wide applicability, instead, is due to the fact that the algorithms implemented by the key functions are general and can be easily adapted to any UUV fleet. Moreover, the proposed decentralized solution permits to accomplish the assigned mission also in case of communication or vehicle faults. Formation of vehicles are both more fault tolerant (through redundancy) and more efficient (through parallelism) than single vehicles, if the vehicles are well coordinated, as shown in the presented simulation results. However, in real applications, attention must be paid to the implementation of the communication functions (*COM*, *LANCOM* and *WANCOM*) and of those operating on the single vehicle dynamics (*LOWCTRL* and *C,VFAULT*) since they depend, respectively, on the implementation of the communication channel and the type of vehicle adopted.

The next step of the proposed activity is to implement the introduced architecture on a fleet of reduced scale unmanned underwater vehicles to improve the non-key functions and to fully test the key functions in a real application scenario.

References

1. Antonelli G, Chiaverini S (2004) Fault tolerant kinematic control of platoons of autonomous vehicles. In: Proceedings of the IEEE international conference on robotics and automation (ICRA'04), vol 4
2. Arrichiello F, Chiaverini S, Fossen TI (2006) Formation control of underactuated surface vessels using the null-space-based behavioral control. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, pp 5942–5947
3. Bhatta P, Fiorelli E, Lekien F, Leonard NE, Paley DA, Zhang F, Bachmayer R, Davis RE, Fratantoni DM, Sepulchre R (2005) Coordination of an underwater glider fleet for adaptive ocean sampling. In: Proceedings of the international workshop on underwater robotics
4. Casari P, Stojanovic M, Zorzi M (2007) Exploiting the bandwidth–distance relationship in underwater acoustic networks. In: Proceedings of the MTS/IEEE Oceans conference (OCEANS'07), vol 1, pp 1–6
5. Cheng L, Wang YJ (2004) Fault tolerance for communication-based multirobot formation. In: Proceedings of the international conference on machine learning and cybernetics, vol. 1
6. Chirdchoo N, Soh WS, Chua KC (2008) Mu-sync: a time synchronization protocol for underwater mobile networks. In: Proceedings of the 3rd ACM international workshop on underwater networks. ACM, New York, pp 35–42
7. Curtin TB, Bellingham JG, Catipovic J, Webb D (1993) Autonomous oceanographic sampling networks. Oceanography 6(3):86–94
8. Daigle MJ, Koutsoukos XD, Biswas G (2007) Distributed diagnosis in formations of mobile robots. IEEE Trans Robot 23(2):353–369
9. Di Massa DE, Stewart WK Jr (1997) Terrain-relative navigation for autonomous underwater vehicles. In: Proceedings of the MTS/IEEE Oceans conference (OCEANS'97), vol 1, pp 541–546
10. Encarnaçao P, Pascoal A (2001) Combined trajectory tracking and path following: an application to the coordinated control of autonomous marine craft. In: Proceedings of the 40th IEEE conference on decision and control, vol 1, pp 964–969, Orlando, FL, USA
11. Fiorelli E, Leonard NE, Bhatta P, Paley DA, Bachmayer R, Fratantoni DM (2006) Multi-auv control and adaptive sampling in monterey bay. IEEE J Ocean Eng 31(4):935–948
12. Freddi A, Longhi S, Monteriù A, Vaccarini M (2010) Fault tolerant decentralized nonlinear MPC for fleets of unmanned marine vehicles. In: Proceedings of 8th IFAC conference on control applications in marine systems, Rostock, Germany
13. Freitag L, Grund M, Singh S, Partan J, Koski P, Ball K (2005) The whoi micro-modem: an acoustic communications and navigation system for multiple platforms. In: Proceedings of the MTS/IEEE Oceans conference (OCEANS'05), pp 1086–1092
14. Ghabcheloo R, Aguiar AP, Pascoal A, Silvestre C (2006) Coordinated path-following control of multiple auvs in the presence of communication failures and time delays. In: Proceedings of the 7th IFAC conference on maneuvering and control of marine craft (MCMC'06), Lisbon, Portugal, vol 26, pp 27–32
15. Ghabcheloo R, Aguiar AP, Pascoal A, Silvestre C, Kaminer I, Hespanha J (2006) Coordinated path-following control of multiple underactuated autonomous vehicles in the presence of communication failures. In: Proceedings of the 45th IEEE conference on

decision and control (CDC'06), San Diego, CA, USA, pp 4345–4350
16. Ghabcheloo R, Aguiar AP, Pascoal A, Silvestre C, Kaminer I, Hespanha J (2009) Coordinated path-following in the presence of communication losses and time delays. Naval Postgraduate School Monterey CA Department of Mechanical, and Astronautical Engineering
17. Hegrenaes O, Hallingstad O (2011) Model-aided ins with sea current estimation for robust underwater navigation IEEE J Ocean Eng 36(2):.316–337
18. Hou SP, Cheah Chien Chern (2011) Can a simple control scheme work for a formation control of multiple autonomous underwater vehicles?. IEEE Trans Control Syst Technol 19(5):1090–1101
19. Ihle IAF (2007) Coordinated control of marine craft, vol 18. Faculty of Information Technology, Mathematics and Electrical Engineering, Norwegian University of Science and Technology, Norway
20. Jazwinski AH (1970) Stochastic processes and filtering theory. Academic Press, New York
21. Kinsey JC (2007) Advances in precision navigation of oceanographic submersibles, vol. 67(11). The Johns Hopkins University Press, Baltimore
22. Kinsey JC, Eustice RM, Whitcomb LL (2006) A survey of underwater vehicle navigation: Recent advances and new challenges. In: Proceedings of the 7th IFAC conference on maneuvering and control of marine craft (MCMC'06), Lisbon, Portugal
23. Kopetz H, Schwabl W (1989) Global time in distributed real-time systems. Inst für Techn Informatik Univ, Austria
24. Kyrkjebø E (2007) Motion coordination of mechanical systems: leader–follower synchronization of Euler–Lagrange systems using output feedback control. Norwegian University of Science and Technology, Norway
25. Ling J, Zhao K, Li J, Nordenvaad ML (2011) Multi-input multi-output underwater communications over sparse and frequency modulated acoustic channels. J Acoust Soc Am 130
26. Liu L, Xiao Y, Zhang J (2009) A linear time synchronization algorithm for underwater wireless sensor networks. In: Proceedings of the IEEE international conference on communications (ICC'09), pp 1–5
27. Longhi S, Monteriu A, Vaccarini M (2008) Cooperative control of underwater glider fleets by fault tolerant decentralized MPC. In: Proceedings of the 17th IFAC World Congress, Coex, South Korea
28. Lu C, Wang S, Tan M (2009) A time synchronization method for underwater wireless sensor networks. In: Proceedings of the 48th IEEE conference on decision and control (CDC'09), Shanghai, China, pp 4305–4310
29. Monteriu A, Asthana P, Valavanis K, Longhi S (2007) Model-based sensor fault detection and isolation system for unmanned ground vehicles: theoretical aspects (part I–II). In: Proceedings of the IEEE International conference on robotics and automation (ICRA'07), Rome, Italy, pp 2736–2751
30. Newman P, Durrant-Whyte H (1998) Using sonar in terrain-aided underwater navigation. In: Proceedings of the IEEE international conference on robotics and automation, Leuven, Belgium, vol 1, pp 440–445
31. Paley DA, Zhang F, Leonard NE (2008) Cooperative control for ocean sampling: the glider coordinated control system. IEEE Trans Control Syst Technol 16(4):735–744
32. Pompili D, Akyildiz I (2009) Overview of networking protocols for underwater wireless communications. IEEE Commun Mag 47(1):97–102
33. Scattolini R (2009) Architectures for distributed and hierarchical model predictive control—a review. J Process Control 19(5):723–731
34. Stojanovic M (1996) Recent advances in high-speed underwater acoustic communications. IEEE J Ocean Eng 21(2):125–136
35. Stojanovic M, Freitag L, Leonard J, Newman P (2002) A network protocol for multiple AUV localization. In: Proceedings of the MTS/IEEE Oceans conference (OCEANS'02), Biloxi, Mississippi, USA, vol 1, pp 604–611
36. Tardós JD, Neira J, Newman PM, Leonard JJ (2002) Robust mapping and localization in indoor environments using sonar data. Int J Robot Res 21(4):311
37. Vaccarini M, Longhi S (2007) Networked decentralized MPC for formation control of underwater glider fleets. In: Proceedings of the 7th IFAC conference on control applications in marine systems (CAMS'07), vol 7(1), Bol, Croatia
38. Vaccarini M, Longhi S (2007) Networked decentralized MPC for unicycle vehicles formation. In: Proceedings of the 7th IFAC symposium on nonlinear control systems, Pretoria, South Africa
39. Vanni F, Aguiar AP, Pascoal A (2007) Nonlinear motion control of multiple autonomous underwater vehicles. In: Proceedings of the 7th IFAC conference on control applications in marine systems (CAMS'07), vol 7(1), pp 75–80
40. Xiao Y (2009) Underwater acoustic sensor networks. Auerbach Publications, Boca Raton
41. Zhang F, Fratantoni DM, Paley DA, Lund JM, Leonard NE (2007) Control of coordinated patterns for ocean sampling. Int J Control 80(7):1186
42. Zhengbao L, Feng H, Lu H, Zhongwen G (2011) E2dts: an energy efficiency distributed time synchronization algorithm for underwater acoustic mobile sensor networks. Elsevier J Ad Hoc Netw