Hybrid Control Approach to Multi-AUV System in a Surveillance Mission

Bychkov Igor, Davydov Artem, Nagul Nadezhda, and Ul'yanov Sergey
Matrosov Institute for System Dynamics and Control Theory, Siberian Branch of the Russian Academy of Science, Irkutsk, Russia

***Abstract*—Surveillance missions for multiple autonomous underwater vehicle (AUV) system suggest the use of different modes of operation including organizing and keeping a predefined formation, avoiding obstacles, reaching static and tracking dynamic targets. While exploiting a leader-follower strategy to formation control and the vector Lyapunov function method to controller design, we use discrete-event approach and supervisory control theory to switch between operational modes.**

Keywords—underwater vehicle; formation control; vector Lyapunov function; discrete-event system

I. INTRODUCTION

Nowadays autonomous underwater vehicles (AUV) have become the main tool for environmental monitoring of water space, seafloor mapping, and surveillance, or scanning, operations. It is traditional for scanning missions that a human operator generates a lawn-mowing pattern that covers an area under survey and AUV moves along the generated path at a constant altitude making forth and back movements as if it is mowing a lawn. The pattern is represented as a sequence of waypoints or as a smooth curve passing through them.

The application of coordinated groups (formations) in underwater works may significantly decrease mission duration and improve the operational reliability and robustness against unexpected events. However, the behavior of multi-AUV system is much more complicated and includes the number of elementary behaviors, or *modes of operations*. The following modes can be distinguished for surveillance missions: formation-keeping mode, formation-gathering mode, obstacle avoidance mode. In the formation-keeping mode AUVs try to maintain a desired configuration as accurately as possible. The formation-gathering mode is switched on when AUVs are comparatively far from each other and it is necessary to bring them together to start or continue survey in formation, and also when there is a need to change the formation structure, for example, after bypassing obstacles or reaching the border of the survey area, in order to continue the execution of the surveillance mission more efficiently. AUV activates the obstacle avoidance mode if it encounters an obstacle during the motion. The first two modes determine collective behavior of the group and they are initiated by the leader AUV while the obstacle avoidance mode is activated independently by each AUV.

When implementing the elementary behaviors of AUVs, the following basic problems arise: path generation problem [1] including real-time path correction [2], path-following problem [3], [4], and formation control problem [5], [6]. In this paper, on the basis of a 6-DOF nonlinear dynamic model of AUV, we design a path-following controller based on the conception of virtual target [3] moving along the path to be followed and a formation-keeping controller based on the leader-follower approach [7], [8]. The design of controllers is performed with the use of the numerical technology for analysis and synthesis of nonlinear control systems based on the reduction method [9] and sublinear vector Lyapunov functions [10]. Unlike most of the control design methods exploited in references, this technique allows one to build more practical sampled-data controllers taking into account uncertainties of the AUV's parameters, measurement errors, and constraints on the control actions (control force and torque). The designed controllers form the low level of the designed hybrid control system.

As far as continuous dynamics of the leader and follower AUVs defines their predefined modes of movement, switching between different modes of operation may be described in terms of a discrete-event model. Widely used, discrete-event systems (DES) represent systems evolution by considering the occurrence of some *event* sequences. Supervisory control theory (SCT), developed in 1980s to regulate DES behavior, nowadays becomes powerful instrument in many real life applications including robotics. Recent implementations in this area concern single robot [11], robot groups [12], [13] and robots formation control [14], swarm robotics [15], [16], robot fights [17], etc. However, most of the listed works employ the simplest supervisors constructed on the base of the finite-state automaton model of a system to be controlled, by eliminating unwanted transitions. A popular way of supervisory control design consists in constructing automatons for turning on and off elementary behaviors, building parallel composition of these automatons and analyzing states of the resulting automaton, which represent all possible combinations of elementary behaviors. Transitions that lead to undesired combinations are the subject of disablement by a supervisor. Being rather effective, this approach only partly use the results of SCT, developed in the first place to deal with formal languages describing a system behavior and constraints on it. Important properties of DES, such as controllability of a specification language and non-blockness of a supervisor, are not usually discussed, as well as specifications are often not explicitly defined. We propose a discrete-event model for the leader AUV operational modes switching as a reaction on environment changes, previous and current modes, and designa supervisor providing language-based specification on an AUVs formation movement.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

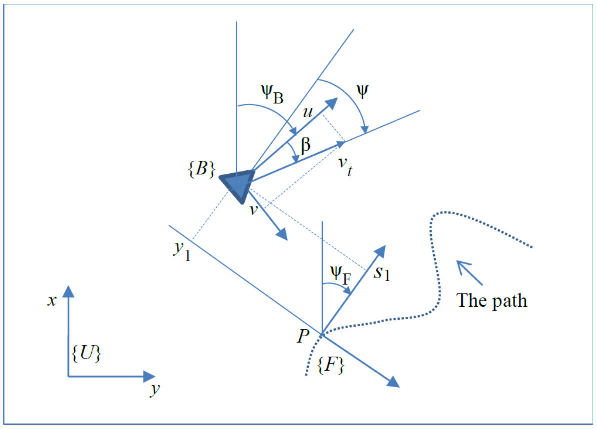

Fig. 1. Reference frames

The rest of the paper is organized as follows. Section II explains the development of low level controllers for AUVs. DES-based top level control algorithms are designed in Section III. Section IV provides simulation results for the designed hybrid control system. Section V contains the conclusions and explanations of future work.

II. Low Level Control Algorithms

A. AUV model

In this paper we use the dynamic model of an AUV borrowed from [3]. The kinematic and dynamics equations of the vehicle can be defined using a global coordinate frame $\{U\}$ and a body-fixed coordinate frame $\{B\}$ (Fig. 1).

The kinematic equation of the AUV can be written

$$\begin{aligned} \dot{x} &= u\cos(\psi_B) - v\sin(\psi_B), \\ \dot{y} &= u\sin(\psi_B) + v\cos(\psi_B), \\ \dot{\psi}_B &= r, \end{aligned} \tag{1}$$

where x, y are the coordinates of the center of mass of the vehicle, ψ_B denotes the yaw angle, u and v are the surge and sway velocities expressed in $\{B\}$, respectively, and r is the angular yaw rate.

Neglecting the equations in heave, roll and pitch, the equations for surge, sway and yaw can be presented as

$$\begin{aligned} \mathrm{F} &= m_u\dot{u} + d_u, \\ 0 &= m_v\dot{v} + m_{ur}ur + d_v, \\ \mathrm{G} &= m_r\dot{r} + d_r, \end{aligned} \tag{2}$$

where $m_u = m - X_{\dot{u}}$, $d_u = -X_{uu}u^2 - X_{vv}v^2$, $m_v = m - Y_{\dot{v}}$, $d_v = -Y_v uv - Y_{v|v|}v\,|\,v\,|$, $m_r = I_z - N_{\dot{r}}$, $d_r - N_v uv - N_{v|v|} - N_r ur$, $m_{ur} = m - Y_r$, m is the mass of the AUV, I_z is the moment of inertia around AUV's vertical axis, $X_{\{\cdot\}}$, $Y_{\{\cdot\}}$, $N_{\{\cdot\}}$ are classical hydrodynamic derivatives, and $[\mathrm{F}\ \mathrm{G}]^T$ is the vector of force and torque applied to AUV.

B. Path-following controller

To design path-following controller for a AUV, the conception of virtual target is exploited. Define the virtual target as a point P that moves along the path to be followed by the AUV. Associated with P, consider the corresponding Serret-Frenet frame $\{F\}$ (see Fig. 1). As shown in [3], the dynamics of the virtual target in $\{F\}$ can be described by

$$\begin{aligned} \dot{s}_1 &= v_t\cos\psi - \dot{s}_a + \dot{\psi}_F y_1, \\ \dot{y}_1 &= v_t\sin\psi - \dot{\psi}_F s_1, \\ \dot{\psi} &= r + \dot{\beta} - \dot{\psi}_F, \end{aligned}$$

where s_1, y_1 are the coordinates of the vehicle in $\{F\}$, s is the signed curvilinear abscissa of P along the path, $\beta = \arctan(v/u)$ is the side-slip angle, $v_t = (u^2+v^2)^{1/2}$ is the absolute value of the total velocity vector; ψ_F is an angle that defines the orientation of F with respect to U ($\dot{\psi}_F = c_c(s_a)\dot{s}_a$), c_c is the path curvature, $\psi \triangleq \psi_B + \beta - \psi_F$. We suppose that the virtual target moves along the path with a desired speed u_d and there is a restriction on the curvature of the path ($|\,c_c\,| \le \bar{c}_c$).

The path-following control problem can be formulated as follows. Given the AUV model (1) and a path to be followed, derive control laws for the force F and torque G that minimize the steady-state errors in variables y_1, s_1, and ψ.

To solve the problem, the sampled-data control law is proposed as:

$$\begin{aligned} \mathrm{F}(t) = \mathrm{F}_c + \mathrm{F}_s, \quad \mathrm{G}(t) &= \mathrm{G}_c + \mathrm{G}_s, \quad t \in T_k \equiv [t_k, t_{k+1}), \\ \mathrm{F}_c = d_u, \quad \mathrm{G}_c &= d_r + m_r(\dot{c}_c\dot{s}_a + c_c\ddot{s}_a - \hat{\ddot{\beta}}), \\ \mathrm{F}_s &= \mathrm{sat}(k_1\hat{s}_{1k} + k_2\Delta\hat{u}_k, \bar{\mathrm{F}}_s), \\ \mathrm{G}_s &= \mathrm{sat}(k_3\hat{y}_{1k} + k_4\hat{\psi}_k + k_5\Delta\hat{r}_k, \bar{\mathrm{G}}_s), \end{aligned} \tag{4}$$

where $t_k = kh$, $k = 0,1,2\ldots$, h is the control step; F_c, G_c are feedforward control terms aimed to cancel terms d_u, d_r in equations (2) and terms $\dot{c}_c\dot{s}_a$, $c_c\ddot{s}_a$, $\ddot{\beta}$ in the equation for variable $\Delta r \triangleq r + \dot{\beta} - \dot{\psi}_F$; $\hat{\ddot{\beta}}$ is an estimate of acceleration $\ddot{\beta}$ obtained using the dynamic model of the AUV (see [3] for details), $\ddot{\psi}_F = \dot{c}_c\dot{s}_a + c_c\ddot{s}_a$; F_s, G_s are feedback control terms, $\bar{\mathrm{F}}_s$, $\bar{\mathrm{G}}_s$ are the shares of maximum control force and torque reserved for stabilization, $\hat{s}_{1k}$, $\hat{y}_{1k}$, $\hat{\psi}_k$, $\Delta\hat{u}_k$, $\Delta\hat{r}_k$ are measurements of variables s_1, y_1, ψ, $\Delta u \triangleq u - u_d$, Δr sampled at time moment t_k with some additive bounded errors; $\mathrm{sat}(\sigma,\bar{\sigma}) = \mathrm{sign}(\sigma)\min(|\sigma|,\bar{\sigma})$ is the saturation function; k_i are feedback coefficients ($i = \overline{1,5}$).

C. Formation controller

Formation control algorithm employed in the scanning mode is based on the leader-follower approach which suggests that each vehicle as a follower tries to maintain a desired position with respect to its leader.

Assume that each AUV is equipped with sensors capable of measuring the relative distance $s = \sqrt{(x_l - x_f)^2 + (y_l - y_f)^2}$ and the bearing angle $\theta = \psi_{Wf} - \arctan\dfrac{y_f - y_l}{x_f - x_l}$ [1]. Dynamics of the leader-follower pair in terms of these variables can be described by

$$\dot{s} = v_{tl}\cos(\psi_{Wl} - \psi_{Wf} + \theta) - v_{tf}\cos\theta,$$
$$s\dot{\theta} = s\omega_f + v_{tf}\sin\theta - v_{tl}\sin(\psi_{Wl} - \psi_{Wf} + \theta).$$

Let a desired relative position of a follower AUV be defined by constants s^* and θ^*. To achieve posture stabilization for the follower AUV, we use the following control law:

$$\mathrm{F}(t) = \mathrm{F}_s + d_u, \quad \mathrm{G}(t) = \mathrm{G}_s + d_r, \quad t \in T_k,$$
$$\mathrm{F}_s = \mathrm{sat}(k_1\Delta\hat{s}_k + k_2 z_k, \overline{\mathrm{F}}_s), \quad \mathrm{G}_s = \mathrm{sat}(k_3\Delta\hat{\theta}_k + k_4 z_k^{\theta}, \overline{\mathrm{G}}_s), \tag{4}$$

where $\Delta\hat{s}_k$, $\Delta\hat{\theta}_k$ are estimates of stabilization errors in distance $\Delta s = s - s^*$ and in the bearing angle $\Delta\theta = \theta - \theta^*$ respectively computed at t_k with the use of discrete filters as

$$\Delta\hat{s}_k = \sum_{\nu=1}^{\mu}\lambda_\nu(\bar{s}_{k-1,\nu} - s^*), \quad \hat{\theta}_k = \sum_{\nu=1}^{\mu}\lambda_\nu^{\theta}(\bar{\theta}_{k-1,\nu} - \theta^*),$$

$\bar{s}_{k-1,\nu} \cong s(t_{k-1,\nu}) + \tilde{s}(t_{k-1,\nu})$, $\bar{\theta}_{k-1,\nu} \cong \theta(t_{k-1,\nu}) + \tilde{\theta}(t_{k-1,\nu})$ are measurements of the distance and bearing angle sampled at $t_{k-1,\nu} = t_{k-1} + \tau_\nu$, $0 \le \tau_\nu \le h$, $\nu = \overline{1,\mu}$ (μ is the memory depth) with some bounded errors $\tilde{s}$, $\tilde{\theta}$; λ_ν, λ_ν^{θ} are parameters of the filters; z_k, z_k^{ϕ} are the outputs of discrete observers of variables $\Delta\dot{s}$ and $\Delta\dot{\theta}$ given by difference equations

$$z_{k+1} = \sum_{\nu=1}^{\mu}a_\nu(\bar{s}_{k,\nu} - s^*) + bz_k \quad z_0 \equiv z(t_0) = 0,$$

$$z_{k+1}^{\theta} = \sum_{\nu=1}^{\mu}a_\nu^{\theta}(\bar{\theta}_{k,\nu} - \theta^*) + b^{\theta}z_k^{\theta} \quad z_0^{\theta} \equiv z_0^{\theta}(t_0) = 0.$$

For modes of operation that require keeping accurately a desired geometrical shape (when scanning), the controller design problem consists in finding the parameters of the control algorithm (feedback and observer's coefficients) that provide robust dissipativity of the formation [18] and minimize its steady state error in variables s and θ. For transient modes (the gathering mode), the parameters of the controller have to be synthesized in such a way that they provide decreasing some given errors in distance s and bearing angle θ in oder to hit the admissible set of initial states of the accurate stabilization modes. The conception of practical stability [18] describes the desired behavior of the formation in this case.

D. Controllers for obstacle avoidance

Assume that, using data obtained from range sensors, AUV can generate a path that allows it to bypass obstacles encountered during the scanning. The problem of real-time path generation in unstructured environment is not considered here and we refer the reader to [19, 2]. Under the assumption above, the problem of bypassing obstacles by the leader AUV can be solved using controller (4). The AUV activates the obstacle avoidance mode when it detects an obstacle. Once the AUV reaches the scanning path (tack), it switches the obstacle avoidance mode to the gathering mode. After bypassing maneuver, the leader of the group can also send control commands to other vehicles to rearrange formation in order to reduce energy consumption (for reasons of efficiency).

The followers of the team keeps a rigid formation with the leader by using control law (4). But once obstacles are encountered it is not possible to meet all of the formation constraints at the same time. Hence we require the follower to keep only a desired distance from the leader when bypassing obstacles. A controller for obstacle avoidance with formation is designed as in (4) except for

$$\mathrm{F}_s = \mathrm{sat}(k_1\hat{s}_{1k} + k_2\Delta\hat{u}_k + k_6\Delta\hat{s}_k, \overline{\mathrm{F}}_s). \tag{5}$$

E. Control design method

The parameters of the proposed sampled-data control algorithms (4) and (5) are synthesized with the use of a technique for rigorous analysis and design of nonlinear control systems based on sublinear vector Lyapunov functions (see, e.g., [20, 10]). When designing controllers, we take into account uncertainties of the AUV's parameters, measurement errors, constraints on the control force and torque.

III. Top Level Control Algorithms

A. Discrete-event systems

Considered as discrete-event system, system functioning is described with sequences of events, or words of some formal language. Let $G = (Q, \Sigma, \delta, q_0, Q_m)$ be a discrete event system

[1]Subscript index l (f) refers to the leader (follower).

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

modeled as a generator of a formal language [19]. Here Q is the set of states q; Σ the set of events; $\delta : \Sigma \times Q \to Q$ the transition function; $q_0 \in Q$ the initial state; $Q_m \subset Q$ the set of marker states. Unlike finite state automaton, which recognizes a formal language, i.e. whether or not a word belongs to the language, generator produces words of some language. As usual, let Σ^* denote the set of all strings over Σ, including the empty string ε. The *closure* of L is the set of all strings that are prefixes of words of L, i.e. $\overline{L} = \{s \mid s \in \Sigma^* \text{ and } \exists t \in \Sigma^* : s \cdot t \in L\}$. Symbol $\cdot$ denotes string concatenation and is often omitted. A language L is *closed* if $L = \overline{L}$. If G is any generator then $L(G)$ is closed.

Language generated by G is $L(G) = \{w : w \in \Sigma^*$ and $\delta(w, q_0)$ is defined$\}$, while language marked by G is $L_m(G) = \{w : w \in L(G)$ and $\delta(w, q_0) \in Q_m\}$. Marked words may be interpreted as completed tasks performed by the system, for example, a finished sequence of actions, which AUV should perform to inspect an objective of interest.

In this paper we suppose that G is fully observable, although SCT for partially observed DES is an interesting and challenging theory, which is indispensable in study of real life systems. Dealing with partial observation in considered AUV group control problem is left for further investigations.

B. The notion of controlled DES

The supervisory control theory (SCT) assumes that some events of G may be prevented from occurring and there exists a means of control presented by a *supervisor* [19]. Let Σ_c be a controllable event set, $\Sigma_{uc} = \Sigma \setminus \Sigma_c$, $\Sigma_c \cap \Sigma_{uc} = \varnothing$. The supervisor switches control patterns so that the supervised DES achieve a control objective described by some regular language K called a *specification* on DES behavior. Formally, a supervisor is a pair $J = (S, \phi)$ where $S = (X, \Sigma, \xi, x_0, X_m)$ is a deterministic automaton with input alphabet Σ. S is considered to be driven externally by the stream of event symbols (words) generated by G (i.e. words from $L(G)$), while $\phi : X \to \Gamma$ is a function that maps supervisor states x into control patterns $\gamma \in 2^\Sigma$. If S is in state x, the events $\sigma \in \Sigma_c$ of G are subject to control by $\phi(x)$. If $\sigma \in \phi(x)$, then σ is enabled, while if $\sigma \notin \phi(x)$ then σ is disabled (prohibited from occurring). Note that, unlike DES models with forced events, enabled events should not necessary occur. It is obvious that ϕ is the state feedback map. Because uncontrollable events cannot be disabled, it is required $\Sigma_{uc} \subseteq \gamma = \phi(x)$. The function δ is now extended to the function $\delta_c : \Gamma \times \Sigma \times Q \to Q$ accounting control patterns as

$$\delta_c(\gamma, \sigma, q) = \begin{cases} \delta(\sigma, q), \ \textit{if} \ \delta(\sigma, q) \ \textit{is defined and} \ \sigma \in \gamma; \\ \textit{undefined}, \ \textit{otherwise.} \end{cases}$$

Construct the function $\xi \times \delta_c : \Sigma \times X \times Q \to X \times Q$, where $(\xi \times \delta_c)(\sigma, x, q) = (\xi(\sigma, x), \delta_c(\phi(x), \sigma, q))$ is defined iff $\delta(\sigma, q)$ is defined, $\sigma \in \phi(x)$ and $\xi(\sigma, x)$ is defined. Denote $L(J / G)$ a language generated by the closed-looped behavior of the plant and the supervisor: $L(J / G) = \{w : w \in \Sigma^*$ and $(\xi \times \delta_c)(w, x, q)$ is defined$\}$. Let $L_m(J / G)$ denote the language marked by the supervisor: $L_m(J / G) = \{w : w \in L(J / G) \text{ and } (\xi \times \delta_c)(w, x_0, q_0) \in X_m \times Q_m\}$. The main goal of supervisory control is to construct such supervisor that $L_m(J / G) = K$. The notion of controllable language is essential in solving this problem.

Definition 1 [21]. *K is controllable (with respect to* $L(G)$ *and* Σ_{uc}*) if* $\overline{K}\Sigma_{uc} \cap L(G) \subseteq \overline{K}$.

Thinking that K is the admissible behavior of the system, it is controllable if occurring of any uncontrolled event after prefix of the word from K leads to a word from K, i.e. still admissible. Checking for controllability is a necessary stage of a supervisor design. For this the product $H \times G$ should be constructed where H is a recognizer of the specification language. Next for all $(q_H, q_G) \in Q_{H \times G}$ inclusion $E(q_G) \cap E_{uc} \subseteq E(q_H, q_G)$ is checked, where $E(q)$ denotes a set of events that are possible is the state q.

Definition 2 [21]. K *is* $L_m(G) - closed$ *if* $K = \overline{K} \cap L_m(G)$.

Supervisor existence criterion sounds as follows: given $K \subseteq L(G)$, there exists supervisor J such that $K = L_m(J / G)$ iff K is controllable and $L_m(G) - closed$ w.r.t. $L(G)$.

C. Discrete-event model for AUVs formation control

From the point of view of the scanning width and maneuverability of the group, it is reasonable for surveillance missions to use in-line formations where the follower is shifted backward with respect to its leader along the driving direction. For reasons of efficiency, it also makes sense to provide possibility of changing positions of the leader in the line formation during the mission, thus distinguishing two types of formations: with the follower on the left from the leader (left formation) and with the follower on the right (right formation).

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

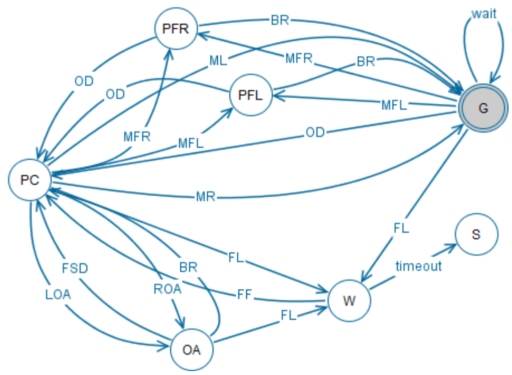

Fig. 2. Leader' generator G_l

To implement SCT for AUVs formation control, first we construct a generator describing switching of a leader's operational modes. Let the set of leader's generator states be $Q_l = \{$ *PFR (path following in right formation), PFL (path following in left formation), OA (obstacle avoidance mode), W (waiting), S (surfacing), PC (path computing), G (formation-gathering mode)* $\}$, $q_{0,l} = Q_{m,l} = G$, and the set of leader events be $\Sigma_l = \{$ *MFR (make right formation), MFL (make left formation), OD (obstacle detected), LOA (obstacle avoidance on the right), LOA (obstacle avoidance on the left), FSD (free space detected), BR (border reached), FL (follower lost), FF (follower found), ML/MR (send message to form left/right formation), timeout, wait* $\}$. Function δ is defined according to Fig. 2.

Assume $\Sigma_{l,uc} = \{OD, BR, FSD, FF\}$. The model does not claim to be exhaustive but presents key points of AUV operation as a leader in scanning mission. Note that treating FL as a controllable event allows one to manage leader's behavior aspects relative to the followers. Indeed, being enabled, this event makes the leader AUV to wait for the lost followers and in case of their absence for a certain time period (*timeout* event), surface. This may be important to get to know a human operator where the lost followers may be found. Recall that enabled events should not necessary occur so enabling controllable *LOA* and *ROA* does not imply their concurrent occurring. Choosing between *LOA* and *ROA* is made on board of AUV according to the obstacle detected since a supervisor is just the means of restriction of system functioning due to some constraints. Let a specification on the leader AUV actions is as described by the language

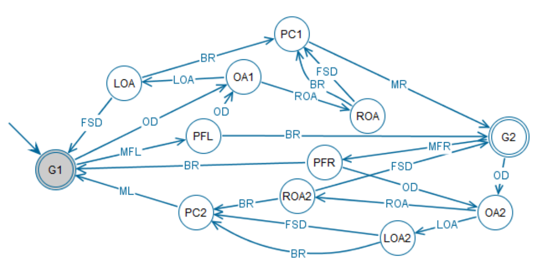

Fig. 3. Automaton H for specification on leader's behavior

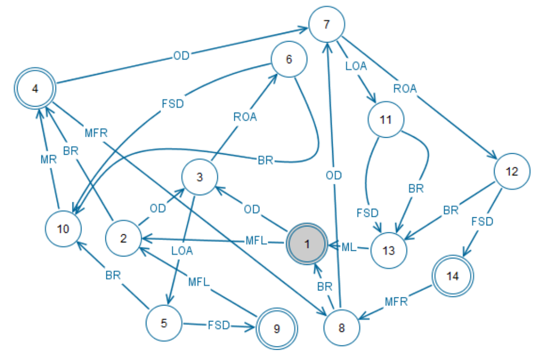

Fig. 4. The automaton S of the supervisor for $L(G_l)$ and K

$K = L(H)$, where H is the automaton on Fig. 3.

This specification implies that AUV group starts scanning mission in the left formation, and after reaching a border of the scanned area, change the formation to the right one. Also, bypassing an obstacle on the left (right) while moving in the left (right) formation, formation is not changed. But after bypassing an obstacle on the left (right) while moving in the right (left) formation, there is not only the need to change a formation but also to compute a new path and gather following AUVs to achieve a new formation. The specification does not suppose awaiting in gathering mode and forbid waiting for the lost followers AUVs. Since event FL is controllable, these do not affect controllability of K, which is easily checked. States of H are named in the the similar way to G_l states for convenience only, for there may be no coincidence in the names of these states. K may be described by any automaton since we are interested only in the language itself but not the way it generated. By marking the states *G1* and *G2* $L_m(G)$ - closeness of K is achieved.

Fig. 4 shows the automaton of a supervisor $J = (S, \phi)$, such that $L_m(J/G) = K$, constructed with the help of DESUMA2 software, and its mapping ϕ is presented in Table 1. In the table enabling of an event corresponds to 1

TABLE I. MAPPING ϕ:X→Γ

	MFL	*MFR*	*FL*	*wait*	*ROA*	*LOA*	*ML*	*MR*
1	1	0	0	0	-	-	-	-
2	-	-	-	-	-	-	-	-
3	0	0	0	-	1	1	0	0
4	0	1	0	0	–	–	–	–
5	–	–	0	–	–	–	–	–
6	–	–	0	–	–	–	–	–
7	0	0	0	–	1	1	0	0
8	–	–	–	–	–	–	–	–
9	1	0	0	–	0	0	0	0
10	0	0	0	–	0	0	0	1
11	–	–	0	–	–	–	–	–
12	–	–	0	–	–	–	–	–
13	0	0	0	–	0	0	1	0
14	0	1	0	–	0	0	0	0

while disabling of an event corresponds to 0. Dashes mean

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

that it does not matter if an event enabled or disabled. Uncontrolled events are not included in the table since they are always enabled. The event *timeout* is not included as well because FL is never enabled and the system never reaches the state W.

Due to the space limitaion we just mention that the follower AUV's generator states set would be $Q_f = \{$ *KRF (keeping right formation), KLF (keeping left formation), OFA (obstacle avoiding in formation), OA (obstacle avoiding), W (waiting), S (surfacing), G (formation gathering)* $\}$ and the set of follower events would be $\Sigma_f = \{$ *MFR (make right formation), MFL (make left formation), OD (obstacle detected), FSD (free space detected), LL (leader lost), LF (leader found), ML/MR (receive message to form left/right formation), timeout* $\}$. Thus the leader and the follower AUVs share events *MFR*, *MFL*, *ML*, and *MR*, and this fact will be used to construct decentralized supervisor in our future work.

IV. Simulation

To demonstrate the effectiveness of the developed approach to hybrid control of multi-AUV systems, numerical computations and simulations were conducted for a group of three identical large-sized AUV with mass $m \approx 2200$ kg and moment of inertia $I_z \approx 2000$ N m^2. These and other parameters of the AUV are borrowed from [3]. For each follower we take $s^* = 11.66$ m and $\delta^* = -1.03$ rad in the left formation $s^* = 11.66$ m and $\delta^* = 1.03$ rad in the right formation. Cubic splines are used to represent predefined and real time generated curved paths for different operational modes.

When designing path-following and formation-keeping controllers and carrying out simulations, we set h = 0.2 s (sampling period, common for all AUVs), $\overline{F}_s$ = 320 N and $\overline{G}_s$ = 160 N m (maximum force and torque reserved for stabilization), $\overline{c}_c$ = 0.12 (constraints on the path curvature). It is worth mentioning that controllers are synthesized off-line and resulting control algorithms can be implemented in AUVs with law computational capacities.

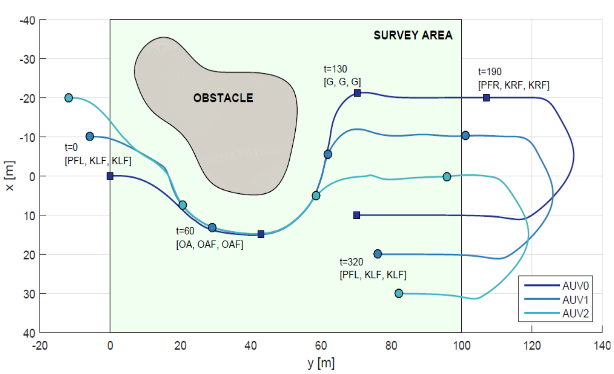

Fig. 5. Simulation results

Fig. 5 shows the trajectory of the group in the following simulation scenario. AUVs start moving along the first line in the left formation, where AUV0 is the leader for AUV1 and AUV1 is the leader for AUV2, until they encounter an obstacle and asynchronously activate the obstacle avoidance modes. After bypassing the obstacle, the group tries to reorganize into the right formation and continue scanning. Once the leader AUV reaches the border of the survey area, it makes U-turn and gathers the group in the left formation for the next line. In square brackets in Fig. 5 we list operation modes of all AUVs at five different time instants.

V. Conclusion

In this paper we generally focused on the continuous dynamics of the AUV formations and briefly discussed the event-driven top-level control. Some results were omitted and lots of questions are left for future work, including the partial observability of system functioning, modular approach to DES construction, and decentralized supervision. The case when a robot in the group shares the functions of a follower for one robot and a leader for another is our immediate research line. Results from [22] will be used to provide supervisor properties.

During surveillance or other complex missions, there may be a situation when several variants of further actions are possible. For example, if a vehicle detects an obstacle, it needs to choose the side to bypass it. The problem of choice cannot be solved using DES framework and therefore it is desirable to have a subsystem, which is responsible for making strategic decision and planning actions using knowledge about underwater environment and AUV's state. In future studies, we are planning to use original logical calculus of positively constructed formulas (PCF) and PCF-based automated theorem proving method [23] for representing and processing this knowledge.

Acknowledgment

The research is supported by the Russian Science Foundation (project no. 16-11-00053).

References

[1] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans,and D. Lane, "Path planning for autonomous underwater vehicles," *IEEE Transactions on Robotics*, vol. 23, no 2, pp. 331–341, April 2007.

[2] A. Bagnitckii, A. Inzartsev, and A. Pavin, "Planning and correction of the auv coverage path in real time," in *2017 IEEE Underwater Technology (UT)*, Feb 2017, pp. 1–6.

[3] L. Lapierre and D. Soetanto, "Nonlinear path-following control of an AUV," *Ocean Engineering*, vol. 34, no. 1112, pp. 1734–1744, 2007.

[4] D. W. Kim, "Tracking of remus autonomous underwater vehicles with actuator saturations," *Automatica*, vol. 58, pp. 15–21, 2015.

[5] X. Li, D. Zhu, and Y. Qiun, "A survey on formation control algorithms for multi-auv system," *Unmanned Systems*, vol. 2, no. 4, pp. 351–359, 2014.

[6] B. Das, B. Subudhi, and B. B. Pati, "Cooperative formation control of autonomous underwater vehicles: An overview," *International Journal of Automation and Computing*, vol. 13, no. 3, pp. 199–225, 2016.

[7] R. Cui, S. S. Ge, B. V. E. How, and Y. S. Choo, "Leader-follower formation control of underactuated autonomous underwater vehicles," *Ocean Eng.*, vol. 37, pp. 1491–1502, 2010.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

[8] P. Millan, L. Orihuela, I. Jurado, and F. R. Rubio, “Formation control of autonomous underwater vehicles subject to communication delays,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 770–777, 2014.

[9] S. N. Vassilyev, “Method of reduction and qualitative analysis of dynamic systems: I, II,” *Journal of Computer and Systems Sciences International*, vol. 45, no. 1, pp. 17–25; no. 2, pp. 167–179, 2006.

[10] I. V. Bychkov, V. A. Voronov, E. I. Druzhinin, R. I. Kozlov, S. A. Ul’yanov, B. B. Belyaev, P. P. Telepnev, and A. I. Ul’yashin, “Synthesis of a combined system for precise stabilization of the spektr-uf observatory: II,” *Cosmic Research*, vol. 52, no. 2, pp. 145–152, 2014.

[11] A. Jayasiri, G.K. Mann, and R.G. Gosine, “Behavior coordination of mobile robotics using supervisory control of fuzzy discrete event systems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 5, pp. 1224–1238, 2011.

[12] X. Dai, L. Jiang, and Y. Zhao, “Cooperative exploration based on supervisory control of multi-robot systems,” *Applied Intelligence*, vol. 45, no. 1, pp. 18–29, 2016.

[13] A. Tsalatsanis, A. Yalcin, and K. P. Valavanis, “Dynamic task allocation in cooperative robot teams,” *Robotica*, vol. 30, no. 5, pp. 721–730, 2012.

[14] G. W. Gamage, G. K. I. Mann, and R. G. Gosine, “Discrete event systems based formation control framework to coordinate multiple nonholonomic mobile robots,” in *Proceedings of the 2009 IEEE RSJ International Conference on Intelligent Robots and Systems*, IROS 09, 2009, pp. 4831–4836.

[15] Y. K. Lopes, S. M. Trenkwalder, A. B. Leal, T. J. Dodd, and R. Groß, “Supervisory control theory applied to swarm robotics,” *Swarm Intelligence*, vol. 10, no. 1, pp. 65–97, 2016.

[16] F. J. Mendiburu, M. R. A. Morais, and A.M.N. Lima, “Behavior coordination in multi-robot systems,” *IEEE International Conference on Automatica (ICA-ACCA)*, Oct. 2016, no. 32, pp. 1–7, 2016.

[17] C. R. Torrico, A. B. Leal, and A. T. Watanabe, “Modeling and supervisory control of mobile robots: A case of a sumo robot,” *IFAC-Papers OnLine*, vol. 49, no. 32, pp. 240–245, 2016.

[18] S. N. Vassilyev, R. I. Kozlov, and S. A. Ul’yanov, “Multimode formation stability,” *Doklady Mathematics*, vol. 89, no. 2, pp. 257–262, 2014.

[19] E. Galceran, R. Campos, N. Palomeras, M. Carreras, and P. Ridao, “Coverage path planning with realtime replanning for inspection of 3d underwater structures,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6586–6591, May 2014.

[20] S. V. Burnosov and R. I. Kozlov. “Investigation of the dynamics of nonlinear systems with uncertainty and perturbations on the basis of the vector lyapunov functions II,” *Journal of Computer and Systems Sciences International*, vol. 53, pp. 82–90, 1996.

[21] P. J. Ramadge and W. M. Wonham, “Supervisory control of class of discrete event processes,” *SIAM J. Control and Optimisation*, vol. 25, no. 1, pp. 206–230, 1987.

[22] N. V. Nagul, “Generating conditions for preserving the properties of controlled discrete event systems,” *Autom. Remote Control*, vol. 77, no. 4, 2016.

[23] A. Larionov, A. Davydov, and E. Cherkashin, “The calculus of positively constructed formulas, its features, strategies and implementation,” in *2013 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1023–1028, May 2013.