Communication Quality in Anticipatory Vehicle Swarms: A Simulation-Based Model

Andrzej M. J. Skulimowski[1,2(✉)] and Arkadiusz Ćwik[1]

[1] Chair of Automatic Control and Biomedical Engineering,
Decision Science Laboratory, AGH University of Science and Technology,
30 Mickiewicza ave., 30-059 Kraków, Poland
ams@agh.edu.pl

[2] International Centre for Decision Sciences and Forecasting,
Progress & Business Foundation, 12B J. Lea Street, 30-048 Kraków, Poland

Abstract. Recent research indicates the important role anticipation plays in the planning and deployment of autonomous multi-vehicle systems. The present study is devoted to building a simulation model of a swarm of autonomous ground-based vehicles. It is assumed that the vehicles perform collaborative surveillance and threat mitigation activities under difficult environmental conditions. Their performance is evaluated as the efficiency of threat mitigation during a single operation cycle, the total damage sustained by the vehicles during the operation, as well as a factor related to operational costs. We will take into account the vulnerability of the LAN communication under the different circumstances that may occur during the swarm operation. Based on the simulation model implemented in Matlab, it has been shown that vehicles endowed with anticipatory decision algorithms and organized in an anticipatory network perform considerably better compared to the behavior that follows a natural swarm benchmark algorithm. The advantage of an anticipatory network organization is particularly salient in case of communication disturbances. In summary, a smooth operation of the swarm can be ensured either by a reliable communication between vehicles via a local network or by implementing an anticipatory self-organization algorithm. Specifically, the latter can compensate for permanent communication deficiencies and may be particularly useful in case of its temporary fallouts due to unexpected disturbances. AQ1

Keywords: Vehicle swarms · Anticipatory networks · Simulation
Agent-based models · Autonomous systems · Multicriteria optimization

1 Introduction

Efficient cooperation of multiple autonomous vehicles requires a deft combination of self-organization and supervision, which is reflected in numerous coordination approaches. Organizing vehicles into formation-like platoons etc. [2] decreases coordination requirements at the expense of decreased self-organization capabilities. This reduces the general flexibility of task fulfilment as well. Vehicle swarms may have a looser internal structure, which is believed to be robust due to the high substitutability of swarm units. The latter enhances communication reliability via multiple parallel

S.-L. Peng et al. (Eds.): IOV 2017, LNCS 10689, pp. 1–16, 2017.
https://doi.org/10.1007/978-3-319-72329-7_11

information transfer channels associated with every pair of units within a vehicle neighborhood [3]. Furthermore, a classical swarm consists of a large number of homogeneous units, with a simple rule-based communication.

The above principles impose restrictions on the organization of swarms composed of a small number of complex intelligent units with a high level of autonomy or freewill [10] and advanced communication capabilities. Studies of such systems, including groups of humans and human-driven vehicles, indicate that applying classical swarm principles may ensure efficient performance and emphasize the role of communication inside the swarm [7]. This is why different vehicle swarm self-organization processes are studied, leading ultimately to a better exploitation of individual unit intelligence, simultaneously preserving the advantages of swarm-type collaboration [14, 15].

Anticipation is a fundamental concept attributed to intelligent systems, both natural and artificial. A formal theory of anticipatory systems was provided by Rosen [8] within the framework of systems biology. By definition, an anticipatory system makes decisions based on a forecasting model of itself and its environment. As such, anticipation describes and explains numerous principles of artificial autonomous decision systems (AADS, [11]), such as traffic management and coordination problems [5, 6]. Moreover, anticipation plays an important role in the control of autonomous vehicles [2] as well as in the computing of transport networks [6] and traffic equilibria. Based on an efficient and extensive information exchange, each individual AADS (here: a vehicle) builds a model of other vehicles and simulates their future behavior before making its own decision.

Classical anticipatory system theory provides an indication as regards the interaction of an individual anticipatory AADS with its environment but is less suitable in explaining the structure of systems composed of multiple AADSs and their dynamics. More light on the rational behavior of multiple anticipatory systems and their self-organization was shed by the theory of *anticipatory networks* introduced in [12, 13]. Each node in an anticipatory network is an anticipatory AADS while the edges model different relations and impacts between them. The decision-making process is analyzed in the context of mutual impacts and information exchanges between nodes, including the crucial concept of *anticipatory feedback* [12]. This is explained in more detail in the next section.

A recent paper [14] presents an application of the anticipatory network theory to planning the operation of vehicle swarms. It has been assumed that swarm members share at least one common goal, while each vehicle also takes into account individual goals. In addition, the vehicles begin their operation as a swarm without a clear internal structure or they may lose it due to the impact of various environmental events. However, the vehicles are capable of organizing themselves into efficient teams thanks to their anticipatory capabilities. Each team can be modelled as an anticipatory network with an ad hoc structure. The organization of teams is driven by supervisory discrete-event control principles. Teams are dissolved after a particular task has been executed or when the final common goal of the swarm has been reached. The above principles have proved useful as a base mechanism for the organization of mining inspection and emergency vehicles, as shown in the above cited paper [14].

This paper presents a simulation environment for a swarm of standardized autonomous vehicles that jointly perform inspection and threat mitigation tasks. In the next section, we provide an outline of the anticipatory network theory, focusing on timed networks and their applicability to swarm robotics. The vehicle simulation model presented in Sect. 3 has been implemented in Matlab and can be run in an independent Matlab Runtime™ environment. In Sect. 5, we present the results of an eight-vehicle swarm simulation. These results will show that vehicles endowed with an anticipatory decision algorithm and organized in an anticipatory network perform considerably better than if the swarm behavior followed a natural benchmark algorithm. The advantage of anticipatory network organization is particularly salient in the case of communication disturbances. It also outperforms the behavior of a swarm of anticipatory vehicles that do not form anticipatory networks. The conclusion section summarizes and discusses these findings. AQ2

2 Timed Anticipatory Networks of Autonomous Vehicles

Anticipatory networks (abbreviated as AN, [12, 13]) on one hand generalize anticipatory models of consequences in multicriteria optimization problems; on the other hand, they can be regarded as an extension of the anticipatory systems of Rosen [8]. According to the basic assumption of this theory, decisions made at a node O in a network can influence algorithms and the scope of decisions made later at another node. This influence defines a causal relation that is described by an acyclic digraph, the first component of an AN. However, there may be more than one causal dependence, so in general the above component is an acyclic multidigraph, where nodes represent decision problems and vertices correspond to causal relations. It will also be assumed that the above causal vertices comply with the time order. The next fundamental assumption states that all decisions in an AN are made to ensure the satisfaction of some additional preference requirements concerning the selection of future decisions. These requirements are termed *anticipatory feedbacks*. They indicate the desired properties of future decisions from the point of view of key decision makers in the network. The anticipatory feedback relation adds an additional component when building an AN as a multidigraph.

The relevant multicriteria optimality principles implemented by decision makers modelled in an AN imply that the decisions admitted should

(a) fulfill the immediate preferences of decision makers,
(b) ensure maximum satisfaction of their wishes concerning the outcomes of those future problems that are starting nodes of the anticipatory feedback relation.

The latter condition can be accomplished by manipulating the causal influences invoked by the choice of decisions. The above construction should be supplemented by:

- a partial order defined on nodes which indicates the relevance of the corresponding decision maker's anticipatory feedback preferences, and/or
- a partial order defined on the vertices of the anticipatory feedback relation which indicates the relevance of each individual relation between two nodes.

The above two additional ordering relations determine the way the AN-solving algorithms [12] are performed. For simulation purposes, we implemented a version of solution algorithm based on the assumption that the earlier the decision is to be made, the more relevant are the preferences of the corresponding decision maker. Furthermore, we assumed that all decisions made by autonomous vehicles in an anticipatory network are rational and cooperative; therefore, the nodes can be called *optimizers*. By definition, an *optimizer O* is a multivalued function that assigns to a set of feasible decisions U and the preference structure P a subset $O(F,U,P)$ of the set $\Pi(U,F)$ of nondominated decisions with respect to multiple optimization criteria $F = (F_1,\ldots,F_N)$ defined at the node O. It is assumed that the preference relation $\leq_P$ associated to P fulfills the condition $x \leq_F y \Rightarrow x \leq_P y$, where "$\leq_F$" is a partial order in U induced by the ordering of the criteria values in IR^N, i.e. $x \leq_F y \Leftrightarrow F(x) \leq F(y)$. Thus, $O(F,U,P)$ contains decisions which were selected from U taking into account both, the order "$\leq_F$" related to the criteria F and the additional preferences P, i.e.

$$O(F,U,P) \subset \Pi(U,F,P) := \{u \in U : [\forall v \in \Pi(U,F) \subset U : F(v) \leq_P F(u) \Rightarrow v = u]\}. \tag{1}$$

The criteria F and the preference structure P (cf. [13] for a discussion of preference structures and models) can be applied to optimize a variety of simultaneous tasks and goals which can be performed by mobile multi-functional vehicles. The swarm of vehicles may have its own goals pre-defined as a subset of criteria F, say $G_1 := (G_1,\ldots, G_{1k})$, while each individual vehicle V_i may additionally optimize its own criterion G_{2i}. The final choice of a nondominated solution is accomplished according to the anticipatory preference structure implied by the requirement (b) above [12].

Now, let us provide formal definitions of the above outlined notions:

Definition 1. Suppose that A is a causal network with nodes corresponding to decision-making units. If a node V_i in A precedes another one, V_j in the causal order r then the *anticipatory feedback* $f_{j,i}$. between V_j and V_i is a specification which outputs from V_j are solicited by V_i. This information is taken into account by V_i when making a decision to influence the choice to be made at V_j, so that a solicited decision could be selected from the subset $\Lambda_{j,i} \subset U_i$ defined by $f_{i,j}$ or be as close as possible to $\Lambda_{j,i}$. ■

An anticipatory feedback makes sense if V_j is causally dependent on V_i or if they are both able to influence another node V_k, which is relevant to V_j. The latter situation is termed *induced anticipatory feedback* [13] and may occur in some special situations of disturbed communication. Now, we can formulate the following definition of an AN.

Definition 2. An *anticipatory network* (AN) is a finite multidigraph with nodes corresponding to anticipatory decision problems, comprising at least one acyclic causal relation and an anticipatory feedback between at least two causally-dependent nodes. ■

An AN node without any causal predecessor will be termed an *initial node*. The simulated vehicle networks will always have one initial node termed *supervisor*. The *anticipatory problem solution* in a vehicle swarm is a collection of solutions to all sequential decision problems solved by the vehicles in order to optimize the common goal G_1, provided that the values of individual vehicle criteria G_2 are nondominated.

The partial values of G_1 resulting from single vehicle decisions are aggregated, first to the momentary values G_{1t} which encompass problems solved by all vehicles until the moment t from a discrete time interval $\{t_1,\ldots,t_{fin}\}$. Thus, the values of G_{1t} aggregate the assessment of all tasks, usually additively, for all $t \in \{t_1,\ldots,t_{fin}\}$.

Let us observe that the solution of an *anticipatory problem* related to a certain task of the swarm need not be immediate; it may extend over certain period and require forming multiple anticipatory networks or changing their structure during the solution process. This leads us [14] to define *timed anticipatory networks* (*TAN*), denoted by $A(t)$, where the anticipatory multigraph may vary for $t \in T$. This network evolution is a major part of the vehicle swarm simulation.

Definition 3 [14]. A *timed anticipatory network* $A(t)$ is a multidigraph-valued time series defined for $t \in T: = \{t_0, t_1, \ldots, t_{fin}\}$, $t_{i-1} < t_i$, where $i = 1,\ldots,fin$, such that

(a) For each $t \in T$ $A(t)$ is an anticipatory network where each decision is to be made within a prescribed time interval $[t, t + \tau(t)]$ ending at the *decision horizon* at t.
(b) For each i, $i = 1,\ldots,fin-1$, $t_{i+1} \geq t_i + \tau(t_i)$ [i.e. the internal network solution processes do not interfere with the network $A(t)$ evolution driven by the time index t].
(c) The decisions made and solutions implemented in the network $A(t_i)$ by all vehicles until the time t_{i+1} comply with the structure of the network $A(t_{i+1})$.
(d) The initial node and at least one other node in $A(t_{i+1})$ inherit the multidigraph structure from $A(t_i)$. ■

Definition 4. Suppose that $A(t)$ is a TAN. The smallest finite digraph $S(A(t))$ with no cycles such that for all $t \in T$ the causal subgraph of $A(t)$ can be embedded as subgraph of $S(A(t))$ will be termed the *structure graph* for A. ■

The following principles will be applied to simulate TANs modelling inspection vehicles in a harsh environment (cf. the next section and [14], p. 69):

- TANs emerge to solve a threat problem and dissolve spontaneously after the threat is mitigated.
- Anticipatory decision problems modelled by $A(t)$, $t = t_0,\ldots,t_n$, are solved independently from each other, but the performance criteria values achieved at each step are merged recursively.
- The time t which occurs in Definition 3 is merely an ordering index for $A(t)$ and cannot be identified with the simulated real time τ. All real-life vehicle operations, such as moving or mitigating threats are performed with respect to τ.
- The time index t_i switches to t_{i+1} after the initial node's decision is made, if a new vehicle is admitted to the network or if another one leaves this TAN. Following the assumption (b) above we can assume that $t_i: = i$, for $i = 1,\ldots,n$.
- The functions of a TAN's vehicle are determined by its position in the structure graph $S(A)$ assigned by the coordinator of this TAN and may vary from t to $t + 1$.

As already mentioned, the swarm performance criteria G are split into two groups: the superordinated vector criterion $G_1 = (G_{1,1},\ldots,G_{1,n})$ is optimized on the set of admissible decisions of the overall $A(t)$, for each $t \in T$, yielding an aggregation of its nondominated values for all $t \in T$. Vehicles performing activities that lead directly to reaching goal G_1 have the right-of-way and priority access to common resources.

The second group is formed by individual performance indicators of V_i, $G_2 := (G_{2i,1}, \ldots, G_{2i,m})$. Their nondominated values will be also combined for all $t \in T$.

The simulation model presented in the next section will touch upon a swarm of identical, autonomous and anonymous *vehicles* $\Xi := \{V_1, .., V_N\}$ that share a common goal G_1 and at the same time optimize individual goals $G_{21}, \ldots, G_{2N}$. The overall swarm performance is described by the vector criterion $G = (G_1, G_2)$, where G_2 is composed of criteria corresponding to the degrees that the individual vehicle goals have been reached. The vehicles perform a given task jointly either acting according to individual swarm-member algorithms (the benchmark case) or they may create a formation following the anticipatory network scheme presented in the previous section. The sets of allowed actions of each vehicle V_i will be denoted by $U_i(t)$. The activity of the swarm while performing a joint task in an anticipatory network formation is shown in Fig. 1.

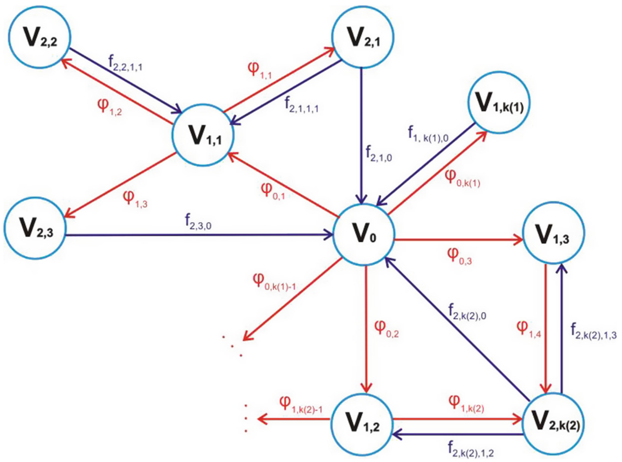

Fig. 1. An example of an anticipatory network formation of a vehicle swarm. V_0 is the coordinating vehicle (virtual supervisor, 0th layer of the AN) that is also an initial element in the network. $V_{1,p}$, $p = 1, \ldots, k(1)$, are coordinated vehicles (1st layer), $V_{2,q}$ $q = 1, \ldots, k(2)$, are monitored vehicles (2nd layer), $\varphi_{i,r}$ are causal influence relations starting from the elements of the i-th layer, $f_{i,j,k,l}$ are anticipatory feedback relations between the units $V_{i,j}$ and $V_{k,l}$ (with $V_{0,0} := V_0$).

A formal statement of the above swarm optimization problem [14] is given below:

$$[\text{Vehicle}\, V_0] \qquad (G_1 : U_0 \to IR^N) \to min, (G_{2,0} : U_0 \to IR^{m(0)}) \to min \tag{2a}$$

$$[V_{1,i}] \qquad (G_1 : U_{1,i} \to IR^N) \to min, (G_{2,i} : \varphi_{0,i}(u_0) \cap U_i \to IR^{m(i)}) \to min, i = 1, \ldots, k(1) \tag{2b}$$

$$[V_{2,j}]\, (G_{2,j} : \varphi_{0,q(j)}(u_0) \cap \varphi_{1,r1(j)}(u_0) \cap \ldots \cap \varphi_{1,r_p(j)}(u_0) \cap U_j \to IR^{m(j)}) \to min, j = 1, \ldots, k(2) \tag{2c}$$

where the notation is the same as that explained above. The anticipatory feedbacks $f_{.,.,k,l}$ are defined as the requirements imposed on the choice of decisions $u_{1,i}$ and $u_{2,j}$:

$$[f_{1,i,0,0}] : u_{1,i} \in G_i^{-1}(\{x \in \Pi_i : x \le g_i\}) \text{ and } \varphi_{0,i} : U_0 \to 2^{U_{1,i}} \text{ exists, for } i \in [1 : n(1)] \quad (3a)$$

$$[f_{2,j,k,l}] : u_{2,j} \in W_{2,k,l} \text{ and } \varphi_{k,p} : U_{k,l} \to 2^{U_{2,j}} \text{ exists, for } j \in [1 : n(2)], k \in \{0,1\}, \quad (3b)$$

where g_i is a target reference level for $G_i := (G_1, G_{1,i})$ – a desired or a satisfactory value of this criterion [9], $\Pi_i := G_i(\Pi(U_{1,i}, G_i, IR_+^{N+m(i)})$, $W_{2,k,l}$ are sets of potential actions of the vehicle $V_{2,j}$ such as the communication between $V_{2,j}$ and $V_{k,l}$ is preserved after the next action of $V_{2,j}$, as anticipated from the perspective of $V_{k,l}$.

The dynamics of the vehicle network $A(t)$ and the assignment of vehicle status and tasks is driven by a discrete-event control system that was presented in [14]. The activity of the supervisor as well as the numerical computations yielding the solution of the anticipatory decision problems based on Algorithms 1 and 2 in [12] proceed in the background while the next two sections focus on the presentation of the overall software architecture and simulation results under different assumptions on the swarm Ξ.

3 Simulation of Anticipatory Vehicle Swarms

Anticipatory system simulation is burdened by a fundamental feature of anticipation, namely that current actions of simulated agents depend on their expectations concerning future states. This means that the results of another simulation looking into a more distant future should be taken into account when modelling present-time vehicle behavior. However, by using two non-overlapping time scales, the TANs defined in Sect. 2 provide an efficient framework for modelling anticipatory swarms. From assumption (b) in Sect. 2 it follows that the planning horizon for anticipatory decisions is always smaller than the corresponding time step in the outer time scale. Therefore, we can decompose the simulation and separate the performance of the swarm decision algorithms that are applied when the threats are mitigated from the simulation of traffic and other surveillance activities.

3.1 Initial Assumptions and the Structure of the Simulation Algorithm

The simulation of vehicles' autonomous capabilities is the main component of the model, which admits i.a. the following assumptions and parameter values:

1 N vehicles inspect a D meter long dual-loop road system (here $N = 8$, $D = 5000$).
2 All vehicles start their operation at the same time from the same point with an initial distance from each other varying from *10* to *20* m. The vehicles may be subdivided into two subgroups, each one starting the exploration from one of two ways available at the starting point. The minimum group size is *3* units.
3 The vehicles operate during an H-hour period (here $H = 8$ h or $H = 28800$ s); after this period, the overall performance of the swarm (function G) is assessed.
4 Two vehicles can pass or overtake each other.
5 A vehicle can change its movement direction at any time.

6 The initial maximum speed of each vehicle is v_{max} mps (here $v_{max} = 1$). Damage $0 \leq \delta(V_j,t) \leq 1$ can reduce it to $v_{max}(1 - \delta(V_j,t))$ m/s. When passing obstacles, the speed can be reduced randomly up to *80%* of the maximum with a uniform reduction distribution. Both speed reductions are independent from each other.

7 There are some *obstacles* in the mine. The obstacles are characterized by the central point of location x, time of appearance t, the severity $s(t,x)$, $0{,}1 < s(t,x) < 1$ that defines the coefficient of the speed reduction and the length $L(t,x)$ of road that is affected by the speed reduction resulting from this obstacle. The present version of the simulation does not allow removing the obstacles. Obstacles cannot overlap.

8 The maximum acceleration of each vehicle is ac_{max} m/s^2 (here $ac_{max} = 1$). Damage $\delta(V_j,t)$ can reduce it to $ac_{max}(1 - \delta(V_j,t))$ m/s^2.

9 *Threats* may appear at random locations x_l and time t_l with intensity $\sigma(t_l,x_l)$. The threat mitigation time depends on the number M of vehicles taking part in the mitigation and on the intensity σ, $\tau_w := \tau_w(\sigma,M)$. The additional energy consumption e_l by each team member during threat mitigation is a function of σ and M and is proportional to τ_w, i.e. $e_l := e_l(\sigma,M,\tau_w)$, $e_l(.,\lambda\tau_w) = \lambda e_l(.,\tau_w)$, for $\lambda > 0$.

10 The probability that a threat is discovered by a j-th vehicle from time t_1 to t_2 depends on the distance d to the threat according to the following simplified rule:

$$\text{if } d < 20 \text{ then } p = 1, \text{ else if } d < 100\, p = d^{-1} \text{ else } p = 0.$$

11 Once a vehicle is assigned to a mitigating team, it moves from its current location to the threat and stays there until this threat is removed.

12 The mutual communication range of a pair of vehicles V_i and V_j depends on their distance $d(V_i,V_j)$ and vanishes when $d(V_i,V_j) > d_0$ meters (here $d_0 = 300$ m).

13 Communication *fallouts* appear spontaneously; a fallout occurring at time t_c affects a certain road interval of length L_c centered around the location x_c with a duration τ_c, all three parameters being uniform distributed random variables. If a vehicle enters the fallout area, its communication range is reduced to ω meters, where ω is a random variable with a uniform distribution on $[0,d_0)$ (meters).

14 The swarm units are not able to forecast or anticipate a fallout, but any vehicle hit by a fallout immediately discovers its parameters. Vehicles affected by a fallout pursue their activities according to the anticipatory task execution principles.

Vehicles may start operating in one formation of *8* robots or they can be initially subdivided into two groups of *4,* or *5* and *3,* units each. In order to compare different algorithms and the impact of communication quality, each formation will be analyzed and simulated independently for the same configuration of threats and obstacles. Therefore, all environmental objects and external events prior to the start of the simulation should be initialized. Then, formation performance will be compared for the same simulated circumstances with and without communication fallouts.

The above assumptions, together with the embedded decision procedures (Algorithms 2 and 3 in the next section), yield the following simulation algorithm:

Algorithm 1 (simulation of a vehicle swarm as an anticipatory network)
Input data structure:
(i) Obstacles table $[x(i),t(i),s(i),L(i)]$ for $i=1,\dots,m_1$; $t(i)>t_0$ implies that this obstacle appears during the vehicle operation
(ii) Threats table $[x(i),t(i),\sigma(t(i),x(i)]$, where σ denotes the intensity, for $i=1,\dots,m_2$
(iii) Communication fallouts table $[x(i),t(i),s(i),L(i)]$, for $i=1,\dots,m_3$
Set vehicles V_j initial positions and directions: 1 (clockwise) or -1 (counterclockwise).
Set $t:=t_0$, $G(t_0):=0$, $F(t_0):=0$, define simulation time step Δt.
Step 1. Analyze the current location of vehicles in Ξ to select the best composition of the current exploratory team.
Step 2. Assign a new status to selected vehicles in Ξ according to the discrete-event system (DES) algorithm in [14]
Step 3. For $j=1,\dots,N$: Calculate the next position of $V_j(t+\Delta t)$ until the selected vehicles reach the working position at a threat to be mitigated.
Step 4a. Solve the problem (1) with the 0-th level objective function G applying the anticipatory network $A(t)$ as an additional preference structure.
Step 5. Set $G(t):=c(G(t), G_\tau(t))$, where c is the temporal combination rule for G and $G_\tau(t)$ is the compromise value of G selected in Step 4. Use a linear combination for additive criteria and a Bayesian rule for the probability.
Step 6. If another threat has been discovered or if the wait list is non-empty, pass the supervision to the appropriate vehicle. **Otherwize** Set $t:=t+\Delta t$.
Step 7n. If $t \le t_f$ continue exploration until it is interrupted by a threat discovery **then Goto Step 1.** Otherwize **Stop** exploration, all robots return to the start point. ■

As an output of Algorithm 1, we get the interim values of G and F as well as their final value after the full operation cycle. The class structure diagram implied by the assumptions 1–14 and the Algorithm 1 is shown in Fig. 2 below.

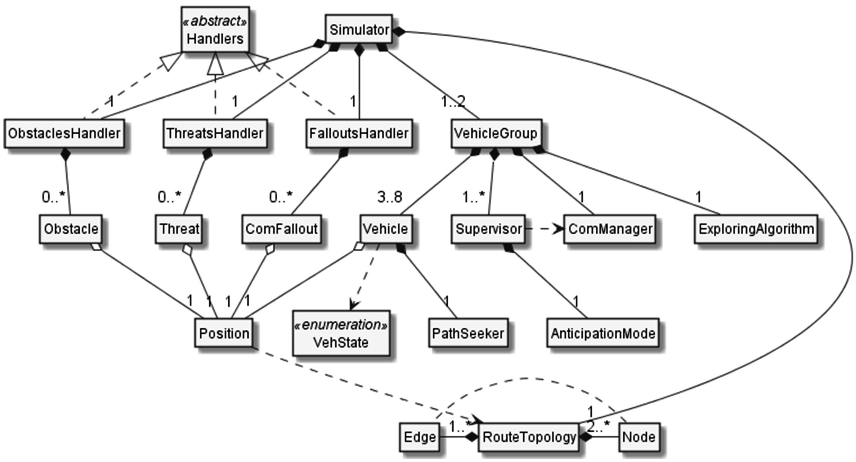

Fig. 2. A class diagram of the anticipatory vehicle simulation

The above diagram uses the standard UML notation (cf. www.uml.org). The classes correspond to the notions used previously, those introduced at the implementation stage are explained below:

- *Supervisor* implements both, the AN's initial node and coordination tasks. In non-anticipatory mode its activity is reduced to communicating new threats.
- *RouteTopology* - a vehicle route graph description with a list of nodes and adjacency matrix.
- *ExploringAlgorithm* - specifies how vehicles move in the route graph.
- *PathSeeker* – a class dedicated to seek optimal (the shortest) route to the desired target.
- *ComManager* - calculates the $N \times N$ communication matrix (its coefficients describe the communication quality for each pair of vehicles) based on vehicles' position and active communication fallouts.
- *Handlers* – an abstract class defining required properties and methods for all (fallout, obstacles, and threat) handlers.

An implementation scheme of the above simulation is presented in the next section.

3.2 Decision-Making Algorithms

The solution principle of the optimal surveillance task admitted in the previous section corresponds to problem P3 in [14]. For simulation purposes, a simplified vector performance criterion $G = (G_{1,1}, G_{1,2}, G_2)$ has been admitted, where:

– $G_{1,1}$ is the efficiency of efforts spent on mitigating threats, measured as the average mitigation time during the operation period, weighted by threat severity,
– $G_{1,2}$ is the use of energy, combining the distance travelled and threat mitigation,
– G_2 is the total amount of damage that hit all vehicles, $G_2(\Xi) := \Sigma_{1 \le j \le N}\, d_j(t_0, t_{fin})$.

All criteria will be minimized. To compare different team configurations, the threat occurrences have been initiated in such a manner so that they can be mitigated by all vehicle teams. The occurrences of threats and communication fallouts remain constant throughout the experiment reported in this paper. The above problem solution is accomplished with the following two optimization algorithms with a constant internal time step. The first one builds the team as an anticipatory network, while the other is applied in a 'naïve' team-building case. Both can be alternatively embedded as Step 4ab in Algorithm 1 to compare the simulation results with different decision procedures.

Algorithm 2 (solution to the timed anticipatory optimization problem)

Step 0. Initialize the data structure according to the initial configuration of the swarm. Set $t:=t_0$, $G(t_0):=0$, define the simulation time step Δt. Start exploring until interrupted by the discovery of a threat by a $V_j \in \Xi$.

Step 1. Analyze the current location of vehicles in Ξ within the communication range of V_j to select the best composition of the exploratory team coordinated by V_j.

Step 2. Assign the new status to selected vehicles in Ξ according to their task in A(t)

Step 3. Solve the problem (1) with the *0*-th level objective function *G* applying the anticipatory network *A(t)* as an additional preference structure.

Step 4. Set $G(t):=c(G(t), G_\tau(t))$, where *c* is the temporal combination rule for *G* and $G_\tau(t)$ is the compromise value of *G* selected in Step 3.

Step 5. If there is another discovery or if the wait list is non-empty, pass the supervision to the appropriate vehicle. **Otherwize** Set $t:=t+\Delta t$.

Step 6n. **If** $t \le t_f$ **then** continue exploration until it is interrupted by a new discovery, update the supervisor's state according to the feedback ξ from the swarm, **Goto Step 1. Otherwize Stop** exploration, return to the start point. ■

The above Algorithm 2 has been embedded in the simulation procedure as an exchangeable module. It can be replaced by other problem solving procedures, such as a natural swarm algorithm, which is outlined below.

Algorithm 3 (threat mitigation by a non-anticipatory swarm)

Step 0. Initialize the data structure according to the initial configuration of the swarm. Set $t:=t_0$, $G(t_0):=0$, $F_i(t_0):= 0$, $\Delta t := d$, the deviation from the benchmark *p*.

Step 1. Explore the road system always choosing the section with the longest time lapse since the last visit or left/right hand rule for counter/clockwise traffic.

Step 2. If (threat discovered)=true **then**:

- **Step 2.1.** Analyze the current location of vehicles in Ξ to select the best composition of the current assisting team. Force the other to assist.
- **Step 2.2.** **If** the threat mitigating team is completed then start joint activities. **Otherwize** perform individual mitigation.
- **Step 2.3.** Compare the mitigation results *H* with the benchmark. If $H < p*H_{avg}$, seek unassigned vehicles to extend the current team.
- **Step 2.4.** **Goto Step 2.2** until the threat is removed.

Step 3. Update G, F_i. Set $t:=t+\Delta t$.

Step 4. If $t \le t_f$ **then** continue exploration until it is interrupted by a new discovery, **Goto Step 1. Otherwize STOP** exploration. ■

Due to the assumed separation of mitigation and exploration, the above Algorithms 2 and 3 can be exchanged to compare the efficiency of the cooperative threat mitigation models. Algorithms 2 or 3 are employed in case of a threat discovery only, after which the application returns to the basic surveillance mode. Therefore, to speed up computation without affecting the generality of results, Algorithms 2 and 3 can first be applied to simulate a sufficient number of different initial configurations of anticipatory networks. The average incremental change of the values of criteria *G* for each initial configuration resulting from this simulation can be calculated and input into the basic algorithm. In this mode, the computation has been considerably faster, cf. the next section for details. The implementation scheme of Algorithms 1 and 2 or 3 is shown in Fig. 3 below.

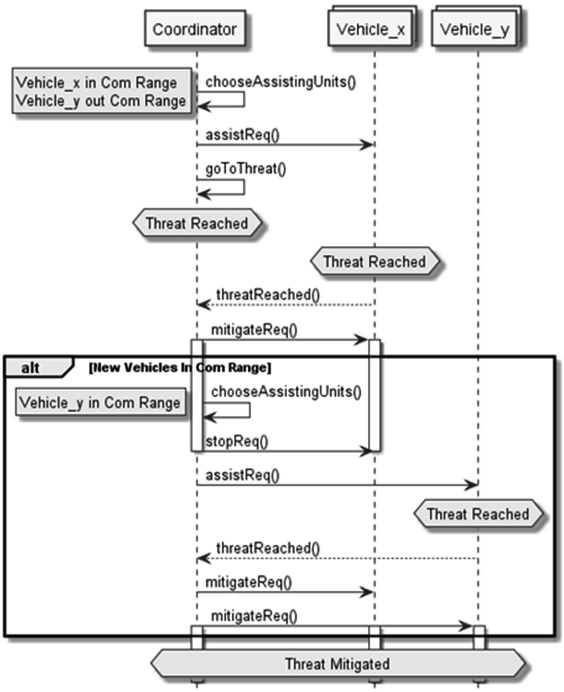

Fig. 3. A scheme of the anticipatory vehicle simulation environment

The procedure *chooseAssistingUnits()* serves to build anticipatory vehicle networks to mitigate threats. Based on current communication matrix, vehicles' positions and threat parameters, the supervisor assigns vehicles with the message *assistReq()* to mitigate the threat just processed. Depending on external circumstances and individual preferences, a vehicle may respond with an internal command *goToThreat()*. The command *threatReached()* is sent to the superviser after the vehicle is ready to start mitigation. With *mitigateReq()*, *stopReq()* and other commands, the supervisor controls the activity of an anticipatory network mitigating a threat.

3.3 Simulation Results: The Impact of Communication on Swarm Performance

The vehicle swarm operation simulation, including the anticipatory multicriteria optimization problem solving procedure, was programmed in Matlab™. The simulated situation presented below can be interpreted as the cooperation of *8* autonomous vehicles that look for threats such as leaks of water or falling rock in a monitored area. The admissible swarm configurations are coded as *(m,n)*, by definition it means that the swarm is subdivided into 2 groups of m and n-elements respectively. If $n = 0$ or $m = 0$

then there is only one group of vehicles. By definition, group 1 (or a single group) selects the way left at the starting point.

Figure 4 shows the visualization of a swarm operation which is delivered during the simulation. Current vehicle parameters and criteria values are shown in separate windows. In the simulation experiment presented in this section, the anticipatory network formation (Algorithms 2) was enabled.

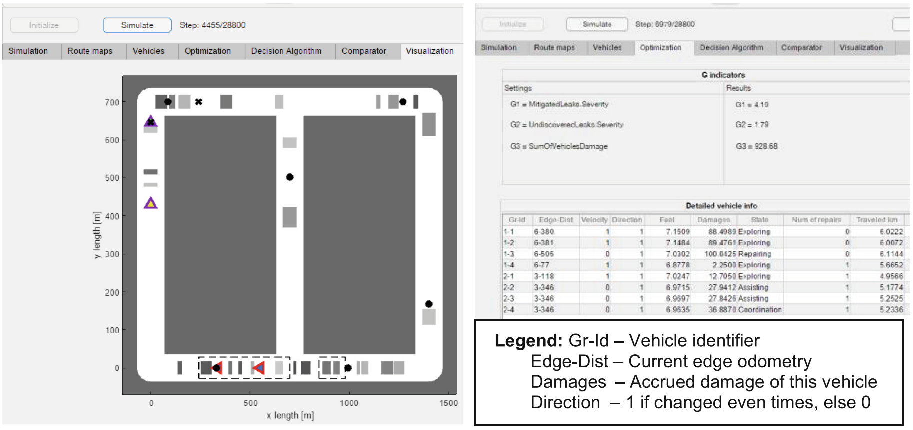

Fig. 4. *Left*: the simulation main visualization screen showing the route system with vehicles marked as triangles. Active threats are marked as dots, those mitigated as "**x**" rectangles, obstacles as shadowed rectangles. Dashed rectangles represent the communication fallouts. *Right*: An auxiliary screen displaying the current values of performance indicators.

The exploration algorithm applies a simple navigation rule based on the 'last-visited' timestamp. Specifically, at a crossing, route intervals (edges) which have not been visited since the longest time ago are likely to be chosen for the immediate inspection. A vehicle can break this rule when it is called by a supervisor to become a member of a threat mitigating team. After completing their mitigation task, vehicles continue journey in the same direction as they did before this task. The performance of the swarm operation simulated with the above assumptions, with the same route, obstacles, threats and communication fallouts in *50* runs, *10* for each of *5* swarm configurations, is presented in Table 1. Its column headings are explained below:

Table 1. Simulation results of a swarm of 8 autonomous anticipatory vehicles in different configurations (1 or 2 groups), compared with and without communication fallouts. An 'Av.' along a configuration mark points out the line with the average value of all indicators. The lines with a 'σ' contain the standard deviations of indicators for this configuration of vehicles

Conf.	Communication disturbances, range 10 m					No communication disturbances, range 300 m				
	TtT	AvMitT (G1,1)	Energy (G1,2)	Damage (G2)	CwOpt	TtT	AvMitT (G1,1)	Energy (G1,2)	Damage (G2)	CwOpt
8–0 av.	124881	634	177,82	3466	57,43%	127440	487	174,18	3149	85,64%
8–0 σ	5948	65	0,93	168	7,59%	4208	27	0,40	28	5,86%
5–3 av.	82960	799	175,06	3822	38,58%	112386	622	169,79	3486	55,08%
5–3 σ	15619	45	0,90	163	3,54%	3004	47	1,65	137	5,03%
4–4 av.	96128	837	172,66	3883	33,42%	86269	531	172,78	3213	71,03%
4–4 σ	3259	113	1,35	310	7,23%	13577	25	1,99	47	5,03%
3–5 av.	78109	961	170,92	4158	29,22%	92484	523	175,61	3135	69,14%
3–5 σ	8972	141	2,84	324	7,28%	203	5	0,57	18	1,63%
0–8 av	135692	589	179,03	3423	64,59%	124431	442	175,66	3105	97,44%
0–8 σ	30923	25	1,83	222	5,85%	536	2	0,76	11	0,53%

TtT – 'Time-to-Threat' – the average time to start mitigating a threat since its discovery, weighted by threat severities, calculated during the whole operation period.
Conf – Configuration of vehicles, e.g. *m-n* - two groups with *m* and *n* vehicles.
AvMitT – 'Average threat Mitigation Time' ($G_{1,1}$) calculated for all vehicles in the swarm during the whole operation period.
Energy ($G_{1,2}$) – the sum of energy consumption of all 8 vehicles during the operation.
Damage (G_2) – the sum of damage coefficients for all vehicles during the operation.
CwOpt – 'Compare with the optimal team'- the ratio (mitigation time with optimal team)/(actual mitigation time in current simulation run) averaged for all threats occurring during the operation period. The '*Mitigation time*' is the total duration of mitigating activities of all team members employed at a given threat, while the '*optimal team time*' is the theoretically minimal mitigation time, pre-calculated for each threat based on its parameters.

The overall impact of communication quality on the swarm performance in each (*j*-th) configuration shown in the above table and within the whole simulated operation period has been calculated as the following coefficient *Q(j)*:

$$Q(j) := (CwOpt(C,j)/CwOpt(CF,j) - 1) * 100\%, \text{ for } j \in [1 : K] \quad (4)$$

CwOpt(C,j) and *CwOpt(CF,j)* are the values of *CwOpt* for the *j*–th swarm configuration calculated without and with communication fallouts, respectively. In the above-presented 8 experiments *Q(j)* varies from 42,77% for two groups with 5 and 3 units) to 136,62% (2 groups of 3 and 5 units). After averaging the values of *Q(j)* over all *K* swarm configurations in the experiment we get the value of $Q(\Xi) = \Sigma_{1 \le j \le K}\, q_j(j)$. In the above presented simulation experiment $Q(\Xi)$ = 78,38%, which represents a considerable improvement of the anticipatory swarm behavior without communication disturbances compared to the case where the effective communication range is reduced to 10 m on average due to the fallouts of a leaky cable or other types of radio connection in areas monitored by autonomous vehicles [1]. It is to be noted that the simulation results yielded by Algorithm 3, i.e. without forming anticipatory networks, are worse in terms of $Q(\Xi)$ on average by about 12% in case without communication fallouts and on 16% when the communication disturbances may occur.

4 Conclusions

Reliable and robust communication between vehicles is a key issue in every application involving vehicle swarm coordination [5]. Anticipation capabilities can be regarded as a substitute for cloud communication in an environment where the reliability of the latter cannot be ensured. In addition, vertical hand-offs [4] may occur in a network when the vehicle changes the subareas of the exploration area with different wire systems. The behavior of a vehicle out of communication range can then be driven by anticipation.

The simulation results presented in the previous section show that an efficient communication enabling data exchange between vehicles is crucial to the performance of surveillance tasks. The organization of vehicles in timed anticipatory networks additionally increases the overall performance indicators by about 15%. The anticipation is based on the knowledge of vehicle technical parameters, such as maximum velocity, acceleration, communication range, navigation and threat mitigation capabilities, as well as on the knowledge of decision algorithms, which are initially identical for all vehicles.

During their activity, some vehicles may sustain damage so their actual parameters should be estimated with measurements of their current dynamics. Similarly, the decision algorithms may be altered due to knowledge base modification during on-the-job learning. However, anticipatory algorithms make it possible to calculate the most probable subsequent activities of other swarm units and use this information to define appropriate future actions. Moreover, forming teams with an anticipatory network structure and solving anticipatory decision problems ensure a fair balance between cooperative (reaching a common goal represented by the function G_1) and conflicting (reaching individual goals G_{2j}) behaviors.

The anticipatory-network-based vehicle team formation presented in [14] as well as the simulation procedure presented in this paper aim to provide initial proof of feasibility for real-life applications of autonomous vehicle swarms. The simulation results show that an autonomous vehicle swarm tasked with threat surveillance may outperform the threat detection capabilities of human staff, and may quickly and efficiently mitigate the discovered threats in tough environmental conditions without endangering human rescue teams.

Acknowledgement. This paper implements selected results of the research project "Scenarios and Development Trends of Selected Information Society Technologies until 2025" financed by the ERDF within the Innovative Economy Operational Program 2006-2013, Contract No. WND-POIG.01.01.01-00-021/09.

References

1. Bandyopadhyay, L.K., Chaulya, S.K., Mishra, P.K.: Wireless Communication in Underground Mines: RFID-Based Sensor Networking, p. 477. Springer, New York, Dordrecht, Heidelberg, London (2010)
2. Bergenheim, C., Shladover, S.; Coelingh, E.: Overview of platooning systems. In: Proceedings of the 19th ITS World Congress, 22–26 October, Vienna, Austria (2012)
3. Ducatelle, F., Di Caro, G.A., Förster, A., Bonani, M., Dorigo, M., Magnenat, S., Mondada, F., O'Grady, R., Pinciroli, C., Rétornaz, P., Trianni, V., Gambardella, L.M.: Cooperative navigation in robotic swarms. Swarm Intell. **8**, 1–33 (2014). https://doi.org/10.1007/s11721-013-0089-4
4. Fan, C., Hsu, C.H., Sun, Q., Yang, F.: A vertical handoff method via self-selection decision tree for internet of vehicles. IEEE Syst. J. **10**(3), 1183–1193 (2016)
5. Hoogendoorn, S., Ossen, S., Schreuder, M.: Empirics of multianticipative car-following behavior. Transp. Res. Record J. Transp. Res. Board **1965**, 112–120 (2006)
6. Huang, W., Viti, F., Tampère, C.M.J.: Repeated anticipatory network traffic control using iterative optimization accounting for model bias correction. Transp. Res. Part C **67**, 243–265 (2016)
7. Pini, G., Gagliolo, M., Brutschy, A., Dorigo, M., Birattari, M.: Task partitioning in a robot swarm: a study on the effect of communication. Swarm Intell. **7**, 173–199 (2013). https://doi.org/10.1007/s11721-013-0078-7
8. Rosen, R.: Anticipatory Systems – Philosophical, Mathematical and Methodological Foundations. Pergamon Press, London (1985). (2nd Ed. Springer, 2012)
9. Skulimowski, A.M.J.: Methods of multicriteria decision support based on reference sets. In: Caballero, R., Ruiz, F., Steuer, R.E. (eds.) Advances in Multiple Objective and Goal Programming. LNEMS, vol. 455, pp. 282–290. Springer, New York (1997)
10. Skulimowski, A.M.J.: Freedom of choice and creativity in multicriteria decision making. In: Theeramunkong, T., Kunifuji, S., Sornlertlamvanich, V., Nattee, C. (eds.) KICSS 2010. LNCS (LNAI), vol. 6746, pp. 190–203. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24788-0_18
11. Skulimowski, A.M.J.: Future prospects of human interaction with artificial autonomous systems. In: Bouchachia, A. (ed.) ICAIS 2014. LNCS (LNAI), vol. 8779, pp. 131–141. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11298-5_14
12. Skulimowski, A.M.J.: Anticipatory network models of multicriteria decision-making processes. Int. J. Syst. Sci. **45**(1), 39–59 (2014). https://doi.org/10.1080/00207721.2012.670308
13. Skulimowski, A.M.J.: The art of anticipatory decision making. In: Kunifuji, S., Papadopoulos, G.A., Skulimowski, A.M.J., Kacprzyk, J. (eds.) Knowledge, Information and Creativity Support Systems. AISC, vol. 416, pp. 17–35. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-27478-2_2
14. Skulimowski, A.M.J.: Anticipatory control of vehicle swarms with virtual supervision. In: Hsu, C.-H., Wang, S., Zhou, A., Shawkat, A. (eds.) IOV 2016. LNCS, vol. 10036, pp. 65–81. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-51969-2_6
15. Veres, S.M., Molnar, L., Lincoln, N.K., Morice, C.P.: Autonomous vehicle control systems - a review of decision making. Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng. **225**, 155–195 (2011). https://doi.org/10.1177/2041304110394727