Intelligent Information System Supporting
Observation, Searching and Detection for
Security of Citizens in Urban Environment

European Seventh Framework Programme
FP7-218086-Collaborative Project

Deliverable 2.6. Proposed algorithms and mechanisms for cooperation within groups of UAVs

The INDECT Consortium

AGH — University of Science and Technology, AGH, Poland
Gdansk University of Technology, GUT, Poland
InnoTec DATA GmbH & Co. KG, INNOTEC, Germany
IP Grenoble (Ensimag), INP, France
MSWiA — General Headquarters of Police (Polish Police), GHP, Poland
Moviquity, MOVIQUITY, Spain
Products and Systems of Information Technology, PSI, Germany
Police Service of Northern Ireland, PSNI, United Kingdom
Poznan University of Technology, PUT, Poland
Universidad Carlos III de Madrid, UC3M, Spain
Technical University of Sofia, TU-SOFIA, Bulgaria
University of Wuppertal, BUW, Germany
University of York, UoY, Great Britain
Technical University of Ostrava, VSB, Czech Republic
Technical University of Kosice, TUKE, Slovakia
X-Art Pro Division G.m.b.H., X-art, Austria
Fachhochschule Technikum Wien, FHTW, Austria

Document Information

Contract Number	218086
Deliverable Name	Proposed algorithms and mechanisms for cooperation within groups of UAVs
Deliverable number	Deliverable 2.6.
Editor(s)	Paweł Lubarski
Author(s)	Paweł Lubarski, PUT Krzysztof Witkowski, PUT Bartosz Wieloch, PUT Krzysztof Butrymowicz, PUT
Reviewer(s)	Maciej Gucma, PhD, Maritime University of Szczecin
Dissemination level	public
Contractual date of delivery	30-06-2010
Delivery date	30-06-2010
Status	v. 1.0
Keywords	tracking, positioning, UAV, mission planning

This project is funded under 7th Framework Program

Contents

Executive Summary

In this document proposed methods and algorithms for cooperation within groups of UAVs are described.

A general view of the problem is given in Section1.

In Section 2 a brief description of collision avoidance problem is given. Research will be conducted in this area, especially in area of visual detection of obstacles.

Cooperative sensing is described in section 3. It is focused on tracking.

In Section 4 a cooperative decision making problem has been described. In particular a short overview, problem definition, proposed architecture, and algorithms for cooperative observation of moving targets are provided.

Conclusions and future work is described in Section 5.

Introduction

Development of Unmanned Aerial Vehicles (UAVs), basis of the Integrated Air Surveillance System for police departments, is one of the three objectives for WP2. Successful execution of observation and object tracking missions by groups of UAVs requires cooperation mechanisms and algorithms to be proposed and implemented. This includes maintaining connectivity among UAVs with the use of limited range communication links, ensuring collision avoidance capabilities of the swarm, exploiting multiple sensors for more accurate localization of tracked features, and efficient and robust task division among UAVs.

	Awareness	Shared goal	Teamwork
Collective	No	Yes	Yes
Cooperative	Yes	Yes	Yes
Collaborative	Yes	No	Yes
Coordinative	Yes	No	No

Table 1: Multi-agent interactions types

1. Overview

1.1. Multi-agent interactions

Four types of multi-agent interactions can be distinguished (after Parker [13])

- collective,
- cooperative,
- collaborative,
- coordinative.

When entities are not aware of each other, but they share goals and their actions are mutually beneficial, then the interaction is collective. When entities are aware of each other and act to reach shared goals, then the interaction is cooperative. When entities are aware of each other, have individual goals, but their actions do help advance the goals of peers, the interaction is collaborative. Finally, when entities are aware of each other, have individual goals and their actions do not tend to help their teammates, the interaction is coordinative.

Multi-agent planning procedures make up only for a part of a successful cooperative decision framework. Successful cooperation requires answering the questions who is going to run the planning procedures (centralized planning for distributed plans), how execution failures are resolved etc. Accompanying deliverable deals with mission planning aspect for multi-agent teams.

1.2. Challenges

Sycara[16] has stated several problems inherent in the design and implementation of multi-agent systems (MAS):

1. Definition, decomposition, and allocation of problems and synthesis of results among a group agents.
2. Communication and interactions of agents, in particular choice of appropriate languages and protocols.
3. Interoperability of heterogeneous agents and discovery of useful agents in open environment.
4. Guarantees for coherent actions and decision of agents, avoiding harmful interactions and accommodating the non-local effects of local decisions.
5. Finding appropriate representation and reasoning methods for actions, plans, and knowledge of other agents. Also, reasoning about the state of the coordinated process (e.g. initiation and completion)?
6. Recognition and resolution of incoherent viewpoints and conflicting intentions among agents.
7. Practical aspects of software engineering of distributed artificial intelligence systems.

In the scope of the INDECT project we do not need to design universal, general-purpose MAS for which these questions remain open. Concrete application simplify development and allow to design dedicated multi-agent software architecture. In the INDECT project MAS architecture finely adjusted to surveillance is sufficient, where surveillance consists of two basic problems of area coverage and tracking. Given that set of problems is so heavily restricted, the representation of problem and results can be expressed with a set of predefined messages. Another simplification comes from the fact, that agents are benevolent and the issue of conflicting intentions do not arise.

Therefore, decomposition is conducted by dedicated planning procedures rather than general-purpose agreement/negotiation protocol. It removes the necessity for employing inference algorithms for reasoning about other agents actions. However, the question how the results of these procedures are distributed among agents requires an answer. The practical challenges of engineering distributed artificial intelligence systems can be minimized by adopting robotic architectures proved in similar projects and entrenching the development in the simulation environment through the whole time-span of project development.

1.3. Multi-agent robotics taxonomy

Dudek et al. [4] have proposed a taxonomy for multi-agent robotics. They propose multiple dimensions along which robot collectives can be classified.

By the size of the collective:

- SIZE-ALONE - minimal collective of one robot,
- SIZE-PAIR - the simplest group of two robots,
- SIZE-LIM - multiple robots, but their number is small comparing to the size of the task,
- SIZE-INF - effectively an infinite number of robots.

By communication range:

- COM-NONE - robots cannot communicate,
- COM-NEAR - communication is possible with sufficiently nearby robots,
- COM-INF - communication is possible between all robots in a collective.

By communication bandwidth:

- BAND-INF - communication is free,
- BAND-MOTION - communication costs roughly the same as moving,
- BAND-LOW - communication cost is considerably higher than moving,
- BAND-ZERO - there is no communication.

By collective reconfigurability:

- ARR-STATIC - spatial arrangement is constant,
- ARR-COMM - coordinated rearrangement with members that communicate,
- ARR-DYNAMIC - arrangement can change arbitrarily.

By collective composition:

- CMP-IDENT - agents are homogeneous in form and function,
- CMP-HOM - agents have same physical characteristics,
- CMP-HET - different physical characteristics, which often implies behavioral differences.

By communication topology:

- TOP-BROAD - every robot can communicate with all of the other robots,
- TOP-ADD - every robot can communicate with an arbitrary robot,
- TOP-TREE - robots are linked in a tree
- TOP-GRAPH - robots are linked in a general graph

According to these dimensions, group of cooperating UAVs in INDECT project consists of multiple robots (SIZE-LIM) with spatial configuration changing during mission lifespan (ARR-DYNAMIC). Aircrafts may differ, for example with regard to their speed and range (CMP-HET). Free communication (BAND-INF) with nearby robots in a general graph topology (TOP-GRAPH, see next section) is available.

1.4. Interaction topology

The interaction topology establishes the means for the software agents to achieve given goals effectively and without conflicts. It determines how the human operator interacts with a group and how the outcomes of this interaction are forwarded to other team members. Interaction topologies focus on the functional aspect of communication, it builds up on the physical communication topology and abstracts from it. More precisely, direct interaction between agents might in fact be achieved with communication relayed be several third-party agents. On the other hand, agents interaction topology may prohibit direct functional interaction between agents, even though they are capable of direct physical communication. By communication we mean explicit acts of passing and receiving information, rather than implicit information extracted from side-effects of other actions.

Zhu [18] lists four basic type of interaction topologies:

Web-like topology

In the web-like topology every agent can directly interact with all other agents, therefore it follows TOP-BROAD category of communication topology dimension of Dudek taxonomy. Each agent acts as an equal member of the community, and shares goals, domain knowledge and possible action choices. It can be implemented by employing agent-activation scheme called request-and-service protocol, a blackboard kind of communication and task activation approach in which every agent in the MAS can respond to a call issued by one of the agents and perform the task requested, and could be called by other agents to perform specific tasks.

Star-like topology

In the star-like topology distinct supervisory agents exist, that coordinate actions of a whole group. Functional communication is brokered by these supervisory agents. Whenever an agent requires a service or information, it submits high-level description of the required service (or information) to the broker agent instead of asking a specific target agent. The facilitator agent chooses appropriate agent from a dynamic network of agents that contribute services. This scheme enables dynamic addition of agents to the community and extending its capabilities.

Star-like topology offers robust framework for interaction of multiple agents, whose different functionality determines assigned tasks. Existence of supervisory agents facilitates central planning, team negotiation and global capabilities awareness.

Grid-like topology

In the grid-like topology agents cooperate with agents in their vicinity, that form a subset of all agents. Each interconnected group of agents have one area coordinator. Interactions with distant agents in different areas are orchestrated by these supervisory group agents and can be further relayed through multiple in-between agents. These coordinator agents are also responsible for decomposing and distributing tasks to agents within their areas.

The topology is well suited to situation in which benevolent agents act in a well-defined environment and pursue well-defined objectives. It enables increasing total number of agents, while still providing robust means of direct communication for agents engaged in joint tasks.

Hierarchical Collective Agent Network

In the hierarchical collective agent network agents are grouped in layers, that are themselves organized in hierarchy. Agents are not connected in each layer, but are fully connected between layers. Agents at lower layers are coordinated by the agents at higher level. Agents in layers are specialized for concrete, well-defined tasks. Agents at the lower level (the data managing module) interface directly to individual sensor/information resources, while the upper levels layers coordinate their activities using a centralized goal-driven strategy. It balances the centralized control and distributed computation.

Zhu [18] enumerates required functionality that suggest HCAN topology:

1. a flexible software architecture for accommodating system augmentation and evolutions,
2. a powerful representation schema for accommodating heterogeneous forms of information,
3. a diverse interface for various input resources, output formats, and human interactions,
4. an ability of reasoning on incomplete and inconsistent information, and extracting useful knowledge from the data of heterogeneous resources,
5. an ability of incorporating real-time dynamics of the information resources into the system anytime during the operation, and promptly adjusting the reasoning mechanisms,
6. an ability of summarizing and refining knowledge extracted, and distinguishing mission and time critical knowledge from insignificant and redundant ones,
7. a capability of supplying meaningful and accurate explanations, both qualitatively and quantitatively, of the automated system actions,
8. a capability of providing adequate control and scrutinizing of the system operations under the environmental constrains of the given situation.

2. Collision avoidance

2.1. Introduction

One of the major goals for Unmanned Aircraft Vehicles (UAVs) to archive routine access to civil airspace is to develop a collision avoidance system for UAV, which will provide a level of safety comparable with piloted aircrafts. Procedures, analysis and control, which reduces the risk of an air-to-air collision, to the same level currently achieved by manned flight. The most common term for this capability is Detect Sense and Avoid (DSA) that allows to detect and safely steer clear of aircraft or other obstacles. Progress which has been made in DSA technology development is continuing, and more advances are inevitable. Such systems would allow manned and unmanned aircraft to coexist peacefully in the same airspace. The DSA system should detect airborne traffic and respond with appropriate avoidance maneuvers in order to maintain minimum separation distances. Mid-air collisions (MAC)s are rare however they do occur and near misses happen. Most MACs occur under Visual Flight Rules (VFR) conditions. In General Aviation (GA) standards and the recommended practices are established in Annex 2 of the Chicago Convention[1]. One of these rules – ‘See-and-Avoid Rule’ states that:

> “Regardless of whether an operation is conducted under instrument flight rules or visual flight rules, vigilance shall be maintained by each person operating an aircraft so as to see and avoid other aircraft.”

To satisfy this requirements, all UAVs must therefore be able to reliably avoid collisions with all aircraft—cooperative and non-cooperative—at all times.

2.2. Potential Solutions

In certain classes of airspace, air traffic control can give information on traffic and clearance in order to prevent collisions, however See-and-Avoid remains the basic rule, in VFR as in Instrumental Flight Rules (IFR) flights. The ASTM Committee (American Society for Testing and Materials) has issued a published standard for DSA collision avoidance, (F2411-04 DSA Collision Avoidance) that requires a UAV to be able to detect and avoid another airborn object within a range of ± 15° degrees elevation and ± 110° azimuth and to be able to response so that collision is avoided by at least 500 ft. The 500 ft safety bubble is derived from the commonly accepted definition of what constitutes a Near Mid Air Collision.

In the modern air traffic management systems, there are two types of traffic: cooperative and non-cooperative. Cooperative traffic broadcasts its position using a transponder, while non-cooperative traffic does not broadcast information.

2.2.1. Cooperative

There are several active detection sensors such as a transponder or ADS-B[15]. These technologies, though designed for manned aircraft, will likely work with larger UAVs but may present problems to the smaller UAVs which have limited payloads and low electrical generation capabilities. Today's transponders are heavy and require a lot of power. The Traffic Collision Avoidance Systems (TCAS) is another cooperative system that has been proposed as a potential collision avoidance system for UAVs. But its performance, even in large UAVs, has been called into question. One of the reason is that Non-transponding aircraft are not detected. There is another concern that, even if TCAS does work in some UAVs, the slow cruise speeds and maneuvering capabilities may lead to an increase in nuisance alarms in manned aircraft [12].

2.2.2. Non-cooperative

The problem in detecting and tracking non-cooperative aircraft is that they are flying VFR without a transponder [9]. One solution is the use of small passive EO/IR cameras to search the required field of regard and detect the traffic in a manner similar to a pilot's eyes. Such a small, light-weight, low-power system will need to effectively detect and track traffic early enough to avoid conflicts – creating a resolution requirement based on traffic range and the closing velocity. However, the information provided by passive sensors is limited to azimuth and elevation. UAV must be able to detect traffic conflicts and yield the right-of-way when required. Also, all aircraft must avoid creating a hazard, even if they have the right-of-way.

While TCAS may, or may not, be suitable for UAV, a transponder based system provides an obvious solution for cooperative aircraft collision avoidance. On-board air-to-air radar is another demonstrated system that provides sufficient information for aircraft collision avoidance. But many of these technologies have been discounted due to size and power consumption requirements, others hold promise due to advances being made in miniaturization and subsystem capability improvements. There are others seeking novel ways to fuse information from these sensors as well as to develop new sensor/surveillance technologies specifically designed for UAVs.

To use a passive system similar to a pilot's vision, one must first recognize the information available from cooperative and radar based systems. These systems provide the relative position of the traffic and a velocity indication. This allows the system to predict the traffic's flight path and determine whether a conflict exists. Once the conflict is determined, then the pilot can act to avoid the conflict. With passive systems however, position and velocity determination are a little more problematic. While there are techniques to resolve both position and velocity, these add significant amounts of time to the entire collision avoidance solution. Time in this respect, greatly increases the resolution required – and consequently the size, weight and power requirements – which adversely affects the overall UAV design. In order to determine the minimum resolution required of a camera based, passive system there are some important considerations and assumptions to address. In the collision avoidance algorithm first assumption is that the traffic will not maneuver.

2.3. Electro-optic sensor

There are some ways to fuse information from electro-optic sensors. A system capable of real-time detection of a small aircraft (a Beechcraft Bonanza) approaching in different configurations, with sufficient time to perform avoidance maneuvers (though they do not divulge the actual detection range)[17]. Using three cameras, they achieve high-resolution (about 0.5milliradians/pixel), whilemaintaining a large field of regard (about 90 degrees) on one side of the aircraft. An FPGA system identifies potential targets in real-time, and a high-level system tracks these candidates and decides when one of them represents a threat. More in-depth tests using varied types of target aircraft are ongoing. This technology has the benefit of having the potential for small-light implementation suitable for smaller UAVs.

Also a two stage approach, an image processing stage followed by a tracking stage is proposed [8]. The image processing stage isolates potential features and the tracking stage tracks these features to distinguish the real targets from background clutter. For detecting objects on a collision course morphological filtering is used in the image 5 processing stage and the rate of translation and expansion in the tracking stage. For detecting crossing objects a series of filters is applied to the image followed by a tracking algorithm based on Kalman filter.

2.4. Avoiding the collision for passive Sense-And-Avoid

The first assumption [9] is to avoids the collision point by 500 feet or more. We can choose another distance for the calculation, but 500 feet is considered well clear. For a given bank angle, the time required to displace the UAV by 500 feet is relatively independent of the aircraft's velocity. For forward aspect conflicts (head-on ± ˜60°), this can be shown directly through application of the turn rate and radius equations:

$$R = \frac{V^2}{g \tan \beta}$$

$$\gamma = \frac{g \tan \beta}{V}$$

where R is the turn radius in ft, γ is turn rate, g is the acceleration of gravity, β is the bank angle, and V is the aircraft velocity. One can examine the flight path of a turning aircraft a little easier in standard Cartesian coordinates. Using a nominal airspeed of 120 knots true airspeed (KTAS ˜200 ft/s) a bank angle of 18° gives (approximately) a standard rate turn of 3°/s and a turn radius of 3927 feet. The position of the non-maneuvering aircraft is simply:

$$y = Vt$$

The position of the maneuvering aircraft is

$$\begin{aligned} x' &= R - Rcos(\gamma t) \\ y' &= Rsin(\gamma t) \end{aligned}$$

The goal of the maneuver is to be 500 feet away from the non- maneuvering position at the point of collision. This distance (d) is the hypotenuse of the right triangle:

$$\begin{aligned} d &= \sqrt{x'^2 + (y - y')^2} \\ d &= \sqrt{(R - Rcos(\gamma t))^2 + (Vt - Rsin(\gamma t))^2} \\ d &= \sqrt{2R^2(1 - cos(\gamma t)) + V^2 t^2 - 2RVtsin(\gamma t)} \end{aligned}$$

Now iterating from $t = 0$ at 1 second intervals, one can find the value of t for which $d \geq 500$ feet. The surprising thing is that for a given bank angle (in this case, 18°) the time, t, that achieves 500 feet separation is 10 seconds, independent of airspeed, when airspeed is > about 70 KTAS. Using t value, one can calculate the range needed to detect a conflict by adding the velocities and multiplying by t. One can also add any computational delays or pilot response time to t in order to improve the margin of safety.

$$V_{detection} = (V_{UAV} + V_{traffic})t$$

For traffic conflicts along the wingline, the UAV will need to perform a check turn away from the aircraft, or maneuver much sooner than t prior to collision. In either case, the distance for detection needed is less than the maximum for a head-on collision.

2.5. Conclusion

UAVs requirements must be placed upon the establishment of minimum UAV safety criteria with particular emphasis on collision avoidance. Sense-and-avoid capability for UAVs has to be the same equivalent level of safety that “see and avoid” provides for manned aircraft operations. The new approach needs to be adopted now to introduce UAVs into a mixed peacetime manned and unmanned airspace. UAVs needs to remain well clear of traffic conflicts, allows us to develop a straightforward method to determine the resolution requirement of a passive sense and avoid system. The collision avoidance system should be focused on how UAV can “sense-and avoid” traffic in airspace.

3. Cooperative sensing of the targets

3.1. Cooperative tracking

A target tracking problem is to estimate a state of the tracked object basing on information from some sensor(s). In our application we suppose that the target might be a moving object (typically a car or other vehicle) so its state is not only its position but also its velocity vector (orientation and velocity). In our mission scenario the sensor is an on-board video camera mounted on UAV which observes the tracked object and returns its position in the image.

In case of the cooperative tracking, the object (vehicle) is observed by several UAVs simultaneously. Such situation has several advantages over a scenario with only a single UAV:

- having multiple measurements from different UAVs we can get much better estimation of the real object’s position,
- it enables us to calculate the full object’s position (x, y, z) even without any additional information about object’s (ground) altitude or/and distance between the UAV and the target,
- in presence of obstructions (e.g. in urban areas) even if the tracked object will be lost by one of our UAVs the target still may remain visible to the rest of them.

The presented method to exploit observations from multiple UAVs in estimating target’s state follows Bethke et al.[2] article which describes a simple method using a linear Kalman filter and minimizing the errors in distance from each measurement. In following we introduce some notation and describe details of the proposed algorithm.

Let’s assume that there are at least two UAVs, which position may be expressed as:

$$\mathbf{x}_i = \hat{\mathbf{x}}_i + \delta\mathbf{x}_i,\ \mathbf{x}_i \in \mathbb{R}^3$$

where $\hat{\mathbf{x}}_i$ is the measured position of the UAV (e.g. by on-board navigation sensors), and $\delta\mathbf{x}_i$ is a random variable with a known distribution representing possible measurement error.

Each UAV is equipped with a video camera system (or any other sensor which gives as output an angular measurement interpreted as a direction toward some object). The direction from the UAV to tracked vehicle may be presented as a vector:

$$\mathbf{d}_i = \hat{\mathbf{d}}_i + \delta\mathbf{d}_i,\ \mathbf{d}_i \in \mathbb{R}^3$$

where $\hat{\mathbf{d}}_i$ is a normalized (unit-length) vector pointing tracked object returned by the vision system, and $\delta\mathbf{d}_i$ represents uncertainty in the direction vector (which results both from inaccuracy of the gimbal and image interpretation).

Finally, with each UAV a weight w_i is associated. It represents the importance of a given UAV's observation and may vary between all UAVs to express differences in the quality of their estimates. These differences may result from e.g. different video quality, distance to target etc.

Following the transformations presented in [2] we can define the optimal target's state estimate (and the error in this estimate) calculated from measurements as:

$$\hat{\mathbf{q}}^* = \hat{\mathcal{A}}^{-1}\hat{\mathbf{b}}$$

$$\delta\mathbf{q}^* = \hat{\mathcal{A}}^{-1}\delta\mathbf{b} - \hat{\mathcal{A}}^{-1}\delta\mathcal{A}\hat{\mathcal{A}}^{-1}\hat{\mathbf{b}}$$

where

$$\hat{\mathcal{A}} = \sum_{i=1}^{n} w_i(I - \hat{\mathbf{d}}_i\hat{\mathbf{d}}_i^T)$$

$$\delta\mathcal{A} = -\sum_{i=1}^{n} w_i(\delta\mathbf{d}_i\hat{\mathbf{d}}_i^T + \hat{\mathbf{d}}_i^T\delta\mathbf{d}_i)$$

$$\hat{\mathbf{b}} = \sum_{i=1}^{n} w_i\left(\hat{\mathbf{x}}_i - \left(\hat{\mathbf{x}}_i^T\hat{\mathbf{d}}_i\right)\hat{\mathbf{d}}_i\right)$$

$$\delta\mathbf{b} = \sum_{i=1}^{n} w_i\left(\delta\mathbf{x}_i - (\hat{\mathbf{x}}_i^T\delta\mathbf{d}_i)\hat{\mathbf{d}}_i - (\delta\mathbf{x}_i^T\hat{\mathbf{d}}_i)\hat{\mathbf{d}}_i - (\hat{\mathbf{x}}_i^T\hat{\mathbf{d}}_i)\delta\mathbf{d}_i\right)$$

The optimal estimate $\hat{\mathbf{q}}^*$ can be computed in linear time $O(n)$, where n is the number of cooperating UAVs. Let us emphasize, that solving above equation to calculate $\hat{\mathbf{q}}^*$ require at least two UAVs (of course which have different positions) and the fact that all observed direction vectors $\hat{\mathbf{d}}_i$ are not parallel. In other case the matrix $\hat{\mathcal{A}}$ cannot be inverted and we have to make additional assumption such that the target is located on a known altitude.

When the estimate of target location is computed, it can be used as a measurement into a Kalman filter[10] and therefore we can get a better results by reducing the influence of noise. Although we can use the extended Kalman filter here we propose an approach with linear Kalman filter and very simple target's movement model. So let's assume, that the tracked object moves with a constant velocity. Therefore the state of the target may be expressed as a vector

$$\mathbf{x} = [x, y, z, x', y', z']^T$$

The system dynamics at time k are then given by

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{v}_k$$

$$\mathbf{z}_k = \hat{\mathbf{q}}^* = H\mathbf{x}_k + \delta\mathbf{q}^*$$

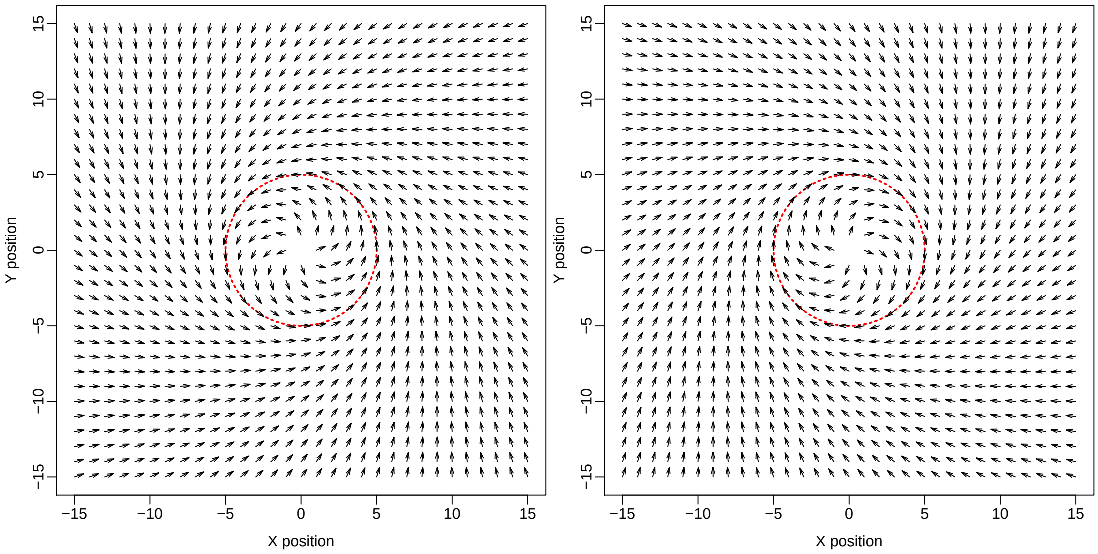
where $\mathbf{v}_k$ is a process noise with a known covariance, and $\delta\mathbf{q}^*$ is the measurement noise. The covariance of $\delta\mathbf{q}^*$ can be calculated as the probability distribution of random variables $\delta\mathbf{x}_i$ and $\delta\mathbf{d}_i$ are known. State transition model F and observation model H are given by

$$F = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

where Δt is a time interval between two measurements (sampling rate).

Now, applying the discrete Kalman filter we can combine information about estimated position of tracked object calculated on UAVs sensor measurement with the assumed model of vehicle movement. Such widely applied solution tends to reduce the influence of noise and therefore returns more stable results. Of course to get even more accurate results, above motion model may be simply extended to a model which reflects also vehicle acceleration etc.



(a) counterclockwise (b) clockwise

Figure 1: Exemplary Lyapunov vector field for $R = 5$

3.2. Cooperative stand-off tracking

Stand-off tracking is a special type of mission in which vehicles (UAVs) should keep a specific minimal distance to a tracked object. This scenario is particularly desired when the UAVs have to avoid some threats or should stay invisible to the observed object (i.e. to flight far enough away from the object to be not audible or visible).

Cooperative stand-off tracking has several issues that have to be addressed:

- a control algorithm is needed that allow unmanaged aerial vehicle to follow simultaneously the target and to preserve the desired distance to the object,
- a mechanism of coordinating the multiple cooperating aircraft which avoids UAVs collisions,
- a cooperating method that minimize the chance of losing the mobile object.

Proposed approaches [7, 6] to this kind of mission set up a circular or elliptical orbit patterns around tracked object. Cooperative phasing around the determined orbit is achieved by different velocity commands sent to each UAV.

The algorithm described here follows Frew paper[6]. It is based on Lyapunov guidance vector field that directs aircrafts to a circular orbit. However, this method stretch the circular pattern into an ellipse to reflect the covariance matrix of estimated object position. In this way, the method process the uncertainty about the target location to increase the probability of stand-off tracking (maintaining the desired separation distance).

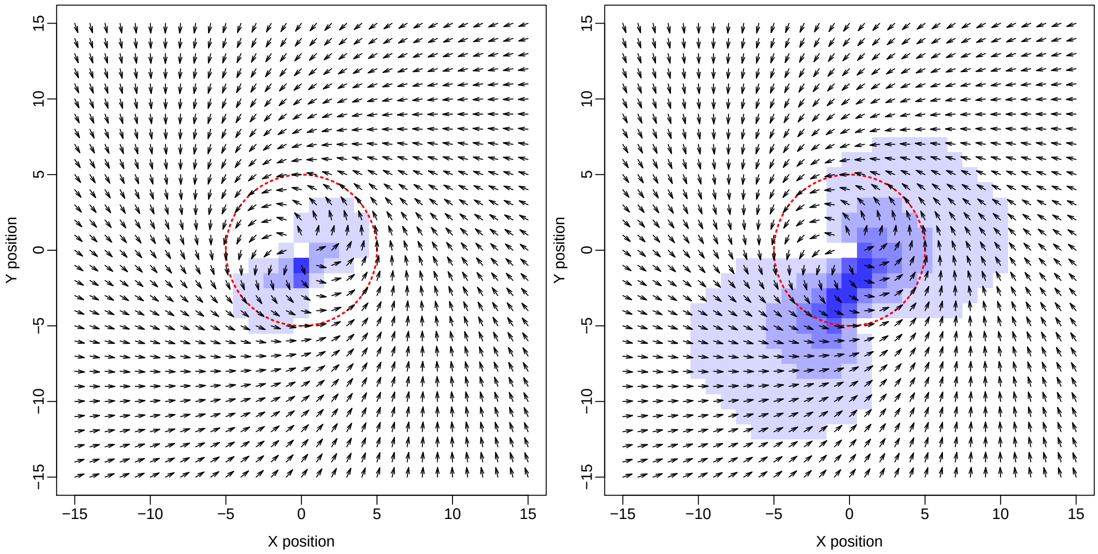
The used Lyapunov guidance vector field has a stable limit cycle centered on the target location. Figure 1 presents an example of the vector field (both counterclockwise and clockwise) for stationary object located at position $(0, 0)$ and stand-off distance set to 5. The image clearly shows that a UAV starting at any point and following a path pointed by the vectors finally end up in a desired orbit (red, dashed circle) around the target.

The equations describing guidance vectors $[x', y']^T$ around a target located at position (x_t, y_t) with the desired stand-off distance R in the Lyapunov vector field may be defined as follow:

$$LVF\left(\begin{bmatrix} x - x_t \\ y - t_t \end{bmatrix}\right) = \begin{bmatrix} x' \\ y' \end{bmatrix} = \left(\frac{-v_0}{r(r^2 + R^2)}\right) \cdot \begin{bmatrix} (x - x_t) & \beta(y - y_t) \\ (y - y_t) & -\beta(x - x_t) \end{bmatrix} \begin{bmatrix} r^2 - R^2 \\ 2rR \end{bmatrix},$$

where

$$r = \sqrt{(x - x_t)^2 + (y - y_t)^2}$$



(a) $v = 2\frac{units}{\Delta t}$ (b) $v = 5\frac{units}{\Delta t}$

Figure 2: Course changing when target is moving northwards.

and v_o is the velocity of UAV. The parameter β equals $+1$ for a counterclockwise orbit around target and -1 for a clockwise direction. Now, the course determined by the vector field can be easily calculated as

$$\chi = \arctan \frac{y'}{x'}$$

An interesting fact is how the course change when the target is moving. To illustrate this, the course changing in some specific point at location (x, y) is calculated simply as a angular difference between guidance vectors in that point at time t and $t - \Delta t$. Figure 2 presents the course change when the target is moving northwards. Figure 2a shows the situation when in time Δt the target moves two units in North direction, and Figure 2b – when moves five units. In each figure the current (at time t) target position is in point $(0, 0)$ and the red circle denote a desired orbit around target (i.e. relative to it). A color map represents how much the course has changed. White colour means that the angle change was smaller than 30 degrees, a bit darker denote change between 30 and 60 degrees and so on till the darkest blue which means change between 150-180 degrees.

Another important problem is how the vector field should be modified when the target is moving. In such case it is possible that the aircraft will not be able to follow the observed object at all due to the UAV's speed limits. If the aircraft velocity is set to v_0 and assuming that the estimated target velocity equals $\mathbf{v}_t = [x'_t, y'_t]^T$ then basing on the guidance vector field the desired aircraft velocity vector $[x'_a, y'_a]^T$ in point (x, y) can be calculated as

$$\begin{bmatrix} x'_a \\ y'_a \end{bmatrix} = \begin{bmatrix} \alpha x' + x'_t \\ \alpha y' + y'_t \end{bmatrix}$$

where α is a non-negative parameter set in such a way that the norm of the above vector is equal to the desired aircraft velocity v_0 (or at least lie in a range of the acceptable velocities). In other words, α is a non-negative root of the equation:

$$\alpha^2 \cdot (x'^2 + y'^2) + \alpha \cdot 2(x'x'_t + y'y'_t) + x'^2_t + y'^2_t - v_0^2 = 0$$

The wind influence can be quite easily included in the above equations by substituting $[x'_t - W_x, y'_t - W_y]^T$ for $[x'_t, y'_t]^T$ where $[W_x, W_y]^T$ is the background wind velocity. As we can see, the wind velocity can be treated as a modification of target speed (some kind of virtual target velocity) because we consider only relative target and UAV positions. However it is very important to emphasize that the wind (unlike the target) affects the aircraft dynamics and therefore high wind speed can for example cause saturation of bank angle limit.

As mentioned earlier, the method can take into account an uncertainty about the estimated target position to avoid violating the minimum stand-off distance condition. Let $\mathbf{x} = [x, y, x', y']^T$ denote a target state estimation including position and velocity, and a matrix P describing uncertainty is an estimate error covariance. The original Lyapunov vector field can be modified to an elliptical pattern guidance by transformation:

$$LVF'\left(\begin{bmatrix} x - x_t \\ y - t_t \end{bmatrix}\right) = M \cdot LVF\left(M^{-1} \cdot \begin{bmatrix} x - x_t \\ y - t_t \end{bmatrix}\right).$$

The transformation matrix M in the simplest form may be just equal:

$$M = n_\sigma \sqrt{P_p}$$

where n_σ is a scalar parameter describing a confidence level, and P_p is the upper left part of the covariance matrix P associated to the position uncertainty (i.e. the first two rows and columns of P).

But following [6] to guarantee stand-off tracking the loiter pattern should keep the aircraft in a separation distance r_{off} away from the boundary ellipse determined by confidence level n_σ. To do this, we need to calculate singular value decomposition of the above matrix:

$$n_\sigma \sqrt{P_p} = U \cdot \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \cdot V$$

In this representation, the diagonal matrix describes the major and minor axis of the ellipse, and U and V matrices – the orientation of it. Now we add our separation distance r_{off} to the singular values and multiply all the matrices:

$$M = U \cdot \begin{bmatrix} \sigma_1 + r_{off} & 0 \\ 0 & \sigma_2 + r_{off} \end{bmatrix} \cdot V$$

In this way we obtain a new transformation matrix M to remap the Lyapunov vector field to follow elliptical orbit with desired stand-off distance.

The coordination of multiple UAVs in stand-off tracking is understood as a maintenance of a constant angular separation between each UAV. In the described approach this is achieved by varying aircraft velocity proportional to the angular separation error:

$$v = kR(\theta - \theta_D) + v_0$$

where θ is the angle between the given aircraft and the next one which is ahead of it (see Figure 3); θ_D is the desired phase angle between these two UAVs . Of course, the resulting commanded velocity has to be corrected to reflect the influence of wind and target movement.

Usually it is appropriate to spread all aircrafts evenly around the target, so

$$\theta_D = 2\pi/n$$

where n is the number of loitering UAVs. Therefore in case of two UAVs the phase offset equals $\theta_D = \pi = 180°$.

4. Decision making

4.1. Recommended cooperation topology

The proposed interaction topology follows work of Chandler and Patcher [3], who developed a hierarchical distributed control system for wide area search munitions (Fig. 4). Similar architecture was also described by Frew and Elston [5]. The three level hierarchy consists of:

1. team agent,
2. sub-team agents,
3. and vehicle agents.

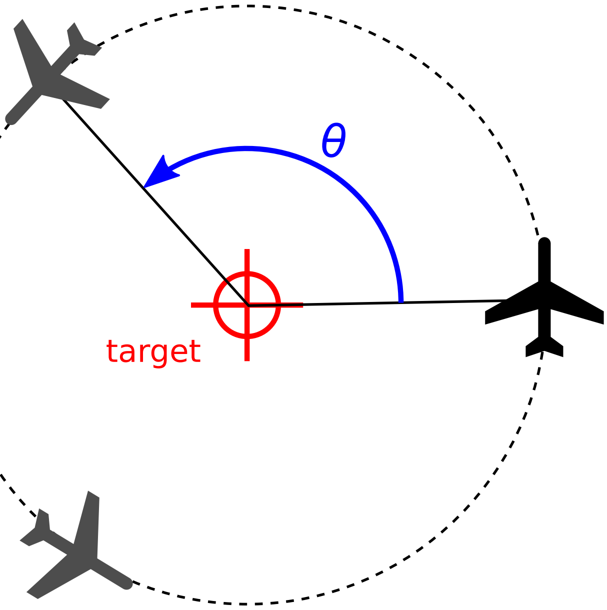


Figure 3: Phase angles between UAVs

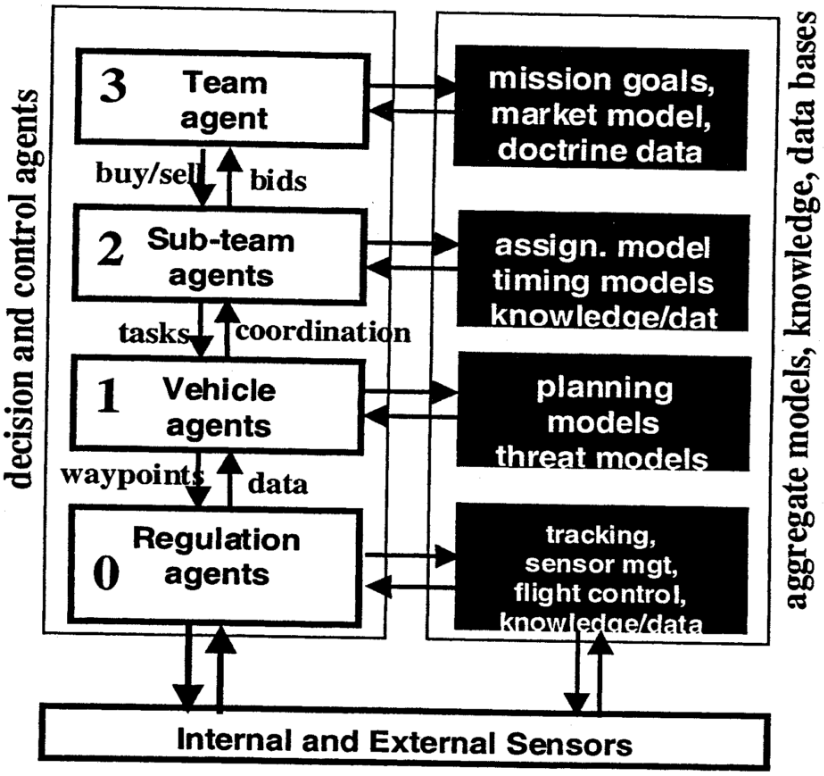


Figure 4: Hierarchical decomposition presented by Chandler and Patcher [3]

At the highest level is one team agent, that manages the mission objectives and is responsible for preparing sub-objectives for sub-teams. It divides swarm vehicles into sub-teams and assigns tasks to them. At the middle level, sub-team managers coordinate the work of robots in their teams in order to efficiently accomplish the tasks that may be divided between multiple vehicles. They are responsible for balancing the responsibilities across their sub-team. Vehicles agents deal with single robot path planning, trajectory generation and maintaining the model of environment (no fly zones, obstacles etc.) for these purposes. Fourth regulation layer is distinguished by Chandler and Patcher, but we treat it as a part of vehicle agent layer.

This hierarchical system is a modification of a grid-topology, as sub-team agents play the role of area supervisors.

4.1.1. Team agent

It is reasonable to assume, that a diameter of the surveillance area might be considerably bigger than maximum communication range of aircrafts. Under such assumption algorithms that require constant communication between cooperating agents can not be applied. Unfortunately, this is a common assumption in many presented algorithms. At the same time it is reasonable to assume that maximum communication range is much bigger than sensor footprint diameter. Therefore using multiple robots for covering an area of diameter close to maximum communication range would be beneficial and would considerably decrease the coverage time. Simple and effective solution to this problem is a decomposition of surveillance area into sub-areas inside which communication is guaranteed.

Circle with a diameter R is an optimal, non-redundant area that guarantees communication of robots located inside, given maximum communication range R. However, it is impossible to decompose plane into circles without overlays. Since inscribed regular polygons have been traditionally used to approximate circle and hexagon provides the best circle approximation from regular polygons that allows construction of tessellations (others are triangles and squares), the hexagon tessellation will be used as the aforementioned decomposition.

Beside the decomposition, the team agent is responsible for high level task assignment based on market analogy. The team agent, which is running at UAV Ground Segment, announces the pending tasks to the available sub-team leaders. They take part in an auction in which they bid for the tasks and also for additional resources (UAVs). Finally, the ground segment announces the auction winners and confirms the tasks assigned to each sub-team.

4.1.2. Sub-team agents

Sub-teams are expected to work in mutual communication proximity within single hexagon tile, hence their work might be coordinated and supervised by a local leader. The local leader is chosen arbitrary and can be changed dynamically if it is malfunctioning. However, the succession rule should be simple and unambiguous, e.g. based on absolute ordering of sub-team members, rather than requiring reaching distributed consensus.

Leader receives sub-mission objectives from team agent and uses multi-robot mission planning algorithms to allocate required tasks among sub-team members. The estimated cost of mission execution determined by the produced plans is used in the auction procedure. Also, sub-team reports potential decrease on mission performance in a hypothetical situation of giving up a vehicle member. This allows for trading vehicles between sub-teams.

The sub-team leader has a supervisory role also during mission execution. For example, if during mission execution sub-team member becomes non-operational, and the mission plan does not handle the issue of robustness explicitly, the interrupted task returns to the task pool managed by the supervisor and is subsequently rescheduled to operational aircrafts.

4.1.3. Vehicle agents

Individual agents receive mission plan constructed by their sub-team agent. They calculate their motions and act according to these plans and, possibly, take over mission responsibilities of non-operational robots.

Single agents architecture follow three-layer hybrid robotic paradigm, that distinguishes between planning, executive and reactive aspects of robot control. Reactive layer is responsible for low-level control of the robot and is operating in tight sensor-action loop. Executive layer is responsible for commanding the reactive layer in order to follow the plan produced by the deliberative layer. Deliberative layer, or planning layer, is responsible for constructing motion plans from the mission plan received from sub-team leader.

4.2. Cooperative Multi-Robot Observation of Multiple Moving Targets

Deployment of mobile robots team, whose task is to observe a set of moving targets with the aim of keeping maximum number of targets under observation by at least one robot, first formalized by Parker in [14], is described as Cooperative Multi-Robot Observation of Multiple Moving Targets (CMOMMT). This particular type of tracking problem finds multiple applications among which tracking people during search and rescue efforts or tracking adverse targets in surveillance operations are of biggest interest in the context of INDECT project.

4.2.1. Definition (after [14])

Given:

S: a two dimensional, bounded,enclosed spatial region,

V: a team of m robot vehicles, v_i,$i = 1$, 2, ..., m, with 360° field of view observation sensors that are noisy and of limited range,

O(t): a set of n targets, $o_j(t)$,$j = 1, 2, ..., n$, such that target $o_j(t)$ is located within region S at time *t,*

We say that a robot v_i is observing a target when the target is within v_i sensing range (defined explicitly below).

Define an $m \times n$ matrix $B(t)$, as follows

$$B(t) = [b_{ij}(t)]_{m\times n}$$

such that

$$b_{ij}(t) = \left\{ \begin{array}{cc} 1 & if\, robot\, v_i\, is\, observing\, target\, o_j(t)\, in\, S\, at\, time\, t \\ 0 & otherwise \end{array} \right\}$$

Then, the goal is to develop an algorithm that maximizes the following metric A:

$$A = \sum_{t=1}^{T} \sum_{j=1}^{n} \frac{g(B(t),\, j)}{T}$$

where:

$$g(B(t),\, j) = \left\{ \begin{array}{cc} 1 & if\, there\, exists\, an\, i\, such\, that\, b_{ij}(t) = 1 \\ 0 & otherwise \end{array} \right\}$$

under the assumptions that:

- the maximum region covered by the observation sensors of the robots is much less than total region to be observed,
- the robots have a broadcast communication mechanism that allows them to send (receive) messages to (from) each other within a limited range. The range of communication is assumed to be larger than the sensing range of the robots, but (potentially) smaller that the diameter of S. This communication mechanism will be used only for one-way communication. Further, this communication mechanism is assumed to have bandwidth of order $O(mn)$ for m robots and n targets,
- following robots can move faster than targets,
- robots share a global coordinate system.

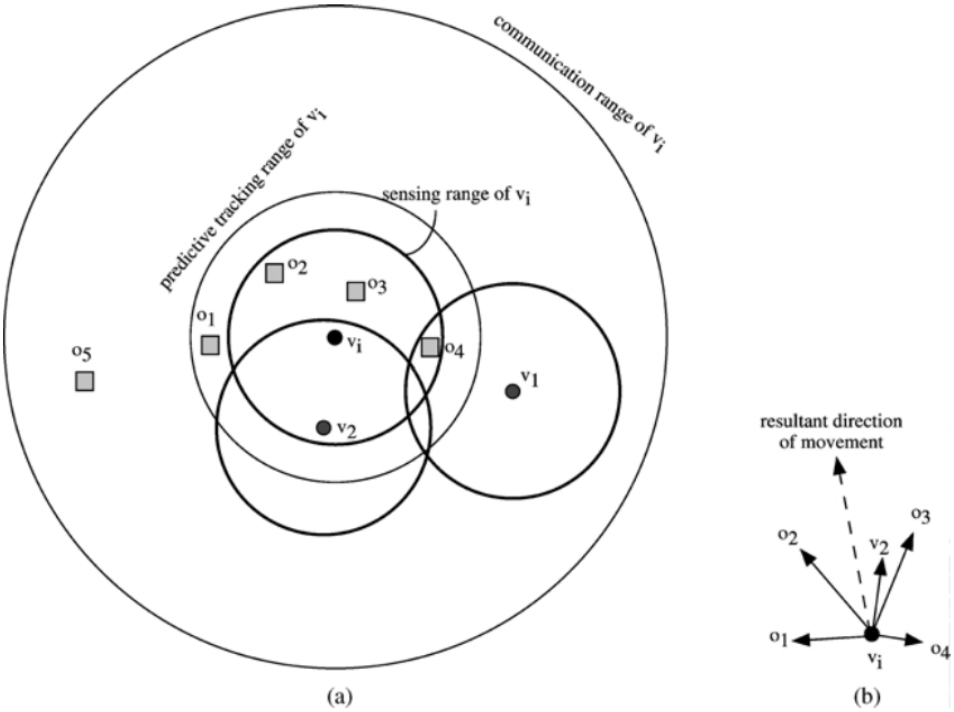


Figure 5: Examples of forces acting on robot v_i (after [14])

4.2.2. A-CMOMMT

Parker presented distributed heuristic behavior based system, in which each agent chooses action independently, without negotiation procedures, and only announces it to the neighboring agents. Each robot broadcasts its own position and position of targets in its field of vision to the neighboring robots. A robot maintains observation of targets within its sensing range using visual tracking algorithms. It tries to predict motion of unseen targets, whose position and velocity has been observed or sent by other robot in the past, within its predictive tracking range. Predictive tracking assumes linear motion according to the last observed state. Therefore robot decision can be influenced by the position of currently seen targets and predicted positions of nearby unseen targets.

The control of the robot is based on the local force vector. Robot is attracted to nearby targets and repulsed from nearby robots. Magnitude of attracting force decreases with a distance to target and reaches zero at predictive tracking range. Analogically, magnitude of repelling force decreases with a distance to another robots.

Although actions are chosen independently, they should not be chosen without considering other team members. In order to do so, the impact of local attraction forces is weighted, so that robot attraction to the nearby target is decreased if that target is within sensing range of another robot. More precisely, weights w_{lk} are introduced, such that when robot v_l detects another robot v_r within sensing range of target o_k, then w_{lk} should be set to low value. Concrete method for determining weights w_{lk} should depend on estimated density of targets.

Under this considerations the direction and speed of each following robot is determined by formula:

$$\sum_{k=1}^{n} w_{lk}\mathbf{f_{lk}} + \sum_{i=1,i\neq l}^{m} \mathbf{g_{li}}$$

where f_{lk} and g_{li} are attraction and repulsion forces.

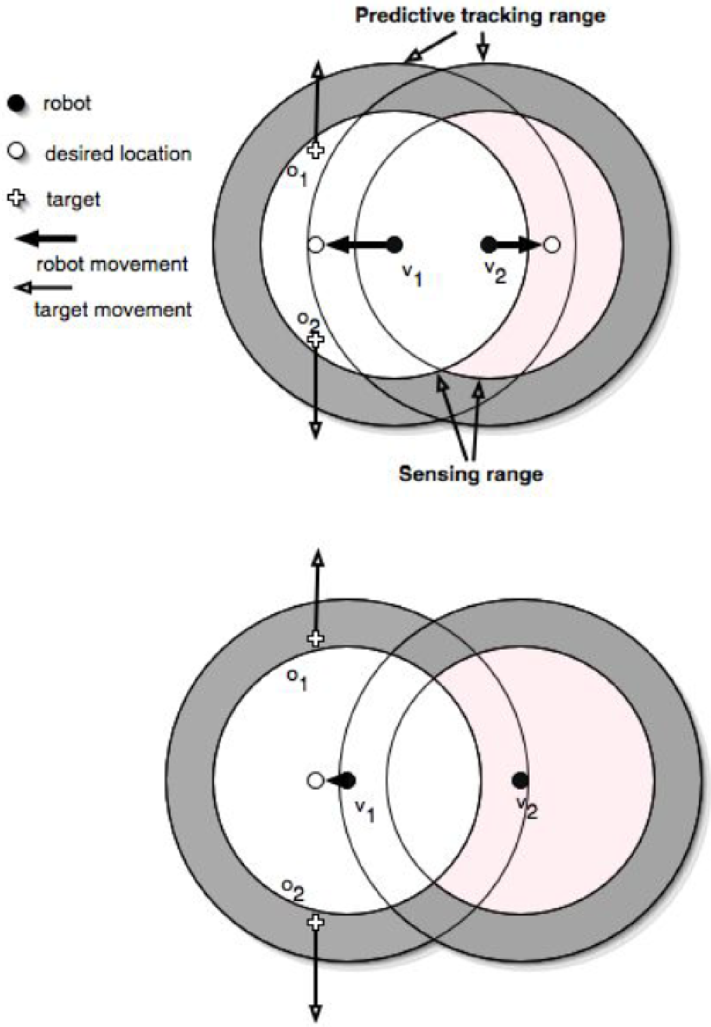


Figure 6: Schematic figure of a situation with undesirable behavior (after [11])

4.2.3. B-CMOMMT

B-CMOMMT, an extension to A-CMOMMT, has been proposed by Kolling and Carpin [11], in which significant weakness of A-CMOMMT is addressed. Lets consider a situation (presented in Fig. 6), when one robot movement is determined by two targets, that move in roughly opposite directions, while another robot that does not have any target in its predictive proximity remains unassigned. The first robot will therefore try to move in the center of gravity of these two targets and will eventually loose track of at least one (or pessimistically both) target. As a result team resources remain unutilized and the mission goal is poorly pursued.

Kolling and Carpin introduced additional *request help* messages. In the A-CMOMMT robots that do not know about any targets in their predictive range follow general purpose exploration algorithm. They might be given information about targets from other regions that will soon escape from sensing range of other robots. Robots which are not engaged in following any target will respond to help request from the most nearby robot and will try to intercept the escaping target.

Target marking mechanism is also introduced in order to ensure that not more then one robot is engaged in following of each target (although one robot might still follow multiple targets), therefore increasing the number of free robots ready to respond to help requests. Each robot marks all targets within its sensing range and broadcasts the distances to the marked targets to other team members within communication range. When a robot receives information about mark placed on the target it had previously marked, the robot with the larger distance to the target removes the mark and resigns from following it.

Only the attraction towards marked targets is taken into account upon calculating the desired direction of robot.

4.2.4. Coordinated binary integer programming approach

Assuming previously described hierarchical grid topology, a coordinated approach can be used in which an area supervisor broadcasts assignment to members of its sub-team. Thus, it is assumed that in the beginning of the task all robots are located in a communication proximity. It can be stated as a binary integer programming problem, where we want to minimize maximum distance between target and its following robot D under constraints:

1. $D \geq d_{ik}x_{ik}$, $i = 1, 2, ..., m$ and $k = 1, 2, ..., n$ — distance D is not smaller than distance between target and its following robot,
2. $\sum_{i=1}^{m} x_{ik} = 1$, $k = 1, 2, ..., n$ — every target is followed by exactly one robot,
3. $\sum_{k=1}^{n} x_{ik} \leq 1$, $i = 1, 2, ..., m$ — each robot follows not more than one target,
4. All of x_{ij} are binary variables (can have only value of 0 or 1),
5. All of d_{ij} coefficients denote Euclidean distance between robot i and target j.

Problem is solved by supervisory agent computing unit and assignments are broadcast to agents under its command. The assignment does not change throughout the mission lifespan and trajectories are determined using single-agent versions of target following algorithms (described in “Proposed algorithms for mission planning for groups of UAVs”).

5. Future work

Research conducted for this paper allows us to implement draft algorithms for testing purpose. Cooperative missions are last element in the schedule for operational UAV. Therefore for Deliverable 2.8 only draft algorithms will be implemented, because for that period the most important result is building a single autonomous UAV. More results will be presented in Deliverable 2.9

Conclusions

This document sums up research about UAV cooperation. Overview of basic terminology and challenges in this area have been presented.

It describes a mechanism of collision avoidance, which is important to maintain safety in group operations. It is also important for UAVs to fly in the same airspace as piloted aircrafts. Proposed research methodology has been described.

Next, in the following section, a cooperative sensing missions are presented. We described a robust method for estimating position of some object observed by several UAVs. This calculation can be done without any other assumption about the object location (e.g. without knowing its altitude) only when we have at least two UAVs observing the same target from different directions. Therefore we describe also a special algorithm determining loitering patterns for stand-off tracking missions in which the tracked object is followed by a group of UAVs preserving some minimal distance to it.

Finally a problem of cooperative decision making is considered. We have proposed system architecture for this solution, as well as appropriate algorithms.

This research enables implementation of algorithms for coopration of UAVs between a group, and will give important feedback to the future work. Starting from methodology through system architecture, and finishing with algorithms to be implemented.

References

[1] *Convention on International Civil Aviation, Ninth edition.* International Civil Aviation Organization, 2007.

[2] B. Bethke, M. Valenti, and J. How. Cooperative vision based estimation and tracking using multiple UAVs. *Lecture notes in control and information sciences*, pages 179–189, 2007.

[3] P. Chandler and M. Pachter. Hierarchical control for autonomous teams. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2001.

[4] G. Dudek, M.R.M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 1996.

[5] Jack Elston and Eric W. Frew. Hierarchical distributed control for search and tracking by heterogeneous aerial robot networks. *2008 IEEE International Conference on Robotics and Automation*, pages 170–175, May 2008.

[6] E.W. Frew. Cooperative Standoff Tracking of Uncertain Moving Targets using Active Robot Networks. In *2007 IEEE International Conference on Robotics and Automation*, pages 3277–3282, 2007.

[7] E.W. Frew and D.A. Lawrence. Cooperative stand-off tracking of moving targets by a team of autonomous aircraft. In *2005 AIAA Guidance, Navigation, and Control Conference and Exhibit; San Francisco, CA*, pages 1–11. American Institute of Aeronautics and Astronautics, 1801 Alexander Bell Drive, Suite 500, Reston, VA, 20191-4344, USA,, 2005.

[8] T. Gandhi, M. Yang, R. Kasturi, O. Camps, L. Coraror, and J. McCandless. Detection of obstacles in the path of an aircraft. *IEEE transactions on aerospace and electronic systems*, 2003.

[9] David E. Grilley. Resolution requirements for passive sense & avoid. *UAV MarketSpace*, 2005.

[10] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[11] a. Kolling and S. Carpin. Multirobot cooperation for surveillance of multiple moving targets - a new behavioral approach. *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, (May):1311–1316, 2006.

[12] James K. Kuchar. Safety analysis methodology for unnmaned aerial vehicle (uav) collision avoidance systems. In *6th Seminar Baltimore*, 2005.

[13] L.E. Parker. Distributed intelligence: Overview of the field and its application in multi-robot systems. *Journal of Physical Agents*, 2(1):5, 2008.

[14] Lynne E Parker. Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets. *Autonomous Robots*, 2002.

[15] Wes Stamper. Understanding mode s technology. *DefenseElectronics*, 2005.

[16] K.P. Sycara. Multiagent systems. *AI magazine*, 1998.

[17] J. Utt, J. McCalmont, and M. Deschenes. Development of a sense and avoid system. 2005.

[18] Qiuming Zhu. Topologies of agents interactions in knowledge intensive multi-agent systems for networked information services. *Advanced Engineering Informatics*, 20(1):31–45, January 2006.

Version	Date	Updates and revision history	Editor
0.1	09/03/2010	Document template	Grzegorz Sobański, PUT
0.11	30/03/2010	ToC Draft	Paweł Lubarski, PUT
1.0	14/06/2010	Stable version for reviews	Paweł Lubarski, PUT