2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Autonomous deployment of swarms of micro-aerial vehicles in cooperative surveillance

Martin Saska[1,*], Jan Chudoba[1], Libor Přeučil[1], Justin Thomas[2], Giuseppe Loianno[2], Adam Třešňák[1], Vojtěch Vonásek[1] and Vijay Kumar[2]

***Abstract*— An algorithm for autonomous deployment of groups of Micro Aerial Vehicles (MAVs) in the cooperative surveillance task is presented in this paper. The algorithm enables to find a proper distributions of all MAVs in surveillance locations together with feasible and collision free trajectories from their initial position. The solution of the MAV-group deployment satisfies motion constraints of MAVs, environment constraints (non-fly zones) and constraints imposed by a visual onboard relative localization. The onboard relative localization, which is used for stabilization of the group flying in a compact formation, acts as an enabling technique for utilization of MAVs in situations where an external local system is not available or lacks the sufficient precision.**

I. Introduction

The task of cooperative surveillance in large outdoor areas by a fleet of Micro Aerial Vehicles (MAVs) is investigated in this paper. In the proposed scenario (see Fig. 1 for motivation), identical MAVs (unmanned quadrocopters) are aimed to cooperatively observe a set of Areas of Interest (AoI) assigned by a human operator in an environment with obstacles (considered as no-fly zones). The surveillance of a particular part of AoI can be realized by a single MAV flying at a sufficiently low altitude (the altitude optimal for the surveillance) or by several MAVs flying in a higher altitude with overlapping views of their surveillance cameras. It is assumed that the utilization of several MAVs observing the same point in the environment from different angles brings an added value for the mission. In addition, such cooperative surveillance of the same area of interest enables coverage of larger areas and increases robustness of the system. In the case of an MAV failure, the mission operator can still obtain data from surveillance sensors of the remaining MAVs. The added value of the cooperative surveillance in comparison to the surveillance of the particular area by a single MAV is expressed in a cost function that evaluates the obtained solution of the swarm deployment (see section III-A for details). Nevertheless, the closely operating group of unmanned quadrocopters requires a precise localization for which open access systems (such as GPS) are not sufficient.

The multi-MAV surveillance approach presented in this paper relies on a relative localization by onboard monocular cameras and simple localization patterns attached to MAVs [1]. This system provides sufficient precision, reliability and fast response to provide feedback for the mutual stabilization of a compact group of MAVs, if the group members respect a set of localization constraints. The constraints are specified mainly by image resolution and viewing angle of cameras and by the size of the localization pattern. This results in constrained relative positions of neighbouring MAVs in the group. Details on the allowed mutual position of the camera and the pattern if using the system of visual relative localization may be found in [2]. The localization system requires that the neighbouring MAVs of the group are kept in the constrained relative positions (later called as the relative localization constraints) during the entire mission to be able to track them and to ensure mutual collision avoidance. Therefore, these localization constraints need to be integrated into the proposed method designed for deployment of MAV groups into locations that are appropriate for the surveillance.

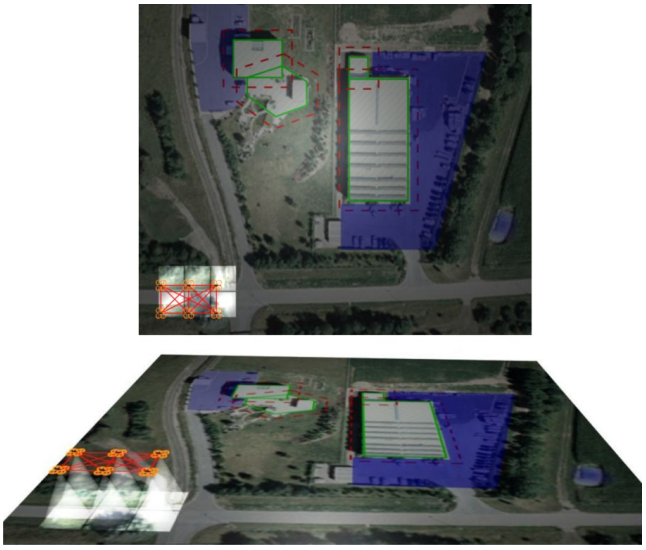
Since the MAV control system uses only the limited information on positions of neighbours and MAVs are assumed to be identical, let us call the employed MAV group a "swarm" to highlight that MAVs can be mutually replaced and that the localization system is not able to distinguish between the group members in its simple but fast version. Let us then call the set of static positions of all MAVs, which are connected into a compact group via the relative localization linkages, a *swarm distribution*. The complete task of the motion of MAVs from the initial depot into the static swarm distribution is called *swarm deployment* in this paper.

In the proposed swarm deployment method, the final swarm distribution is found together with the feasible trajectories of all MAVs from the initial positions into a swarm distribution in a single optimization process. This ensures that the obtained swarm distribution is reachable by MAVs keeping the motion and relative localization constraints. In the optimization, positions of all MAVs are collected into a unique optimization vector, since the coverage of AoI is realized cooperatively and the contribution of a particular MAV depends also on the positions of its neighbors as shown in section III-A. The optimization vector represents the entire solution of the swarm distribution problem. The swarm distribution (the solution of the optimization) is evaluated by an objective function, which reflects its quality with respect to the cooperative surveillance task.

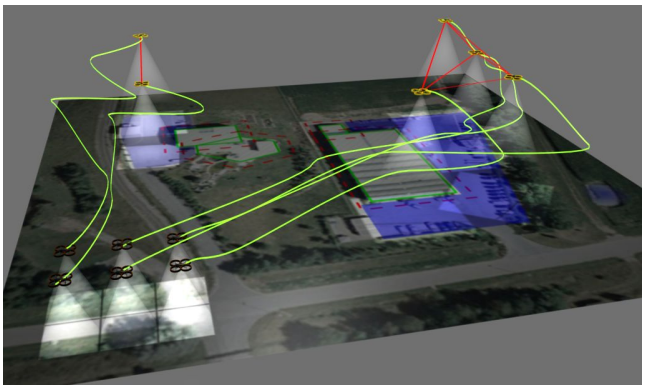
The dimension of the optimization vector is three times the number of MAVs in the swarm, since also the z co-

[1]The authors are with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague. saskam1@fel.cvut.cz, chudoba@labe.felk.cvut.cz, preucil@labe.felk.cvut.cz, spurnvoj@fel.cvut.cz, vonasek@labe.felk.cvut.cz

[2]The authors are with the GRASP Lab, University of Pennsylvania, 3330 Walnut Street, 19104, Philadelphia, jut@seas.upenn.edu, loiannog@seas.upenn.edu, kumar@seas.upenn.edu.



(a) Initial position of the MAV group prepared for the surveillance mission (top view and side view). Obstacles are denoted by the green lines, borders of non-fly zones (dilated obstacles) by the dashed red lines and areas of interest are depicted as the blue regions.



(b) MAV group deployed in the environment to cover the areas of interest by onboard surveillance cameras. The range of the cameras is visualized by the white pyramids. The yellow curves represent trajectories found during the swarm deployment optimization and the red lines represent the relative localization linkages.

Fig. 1. Motivation of the investigated swarm deployment task.

ordinate (the altitude of MAVs) influences the surveillance performance. We propose to solve this high dimensional optimization problem by the Particle Swarm Optimization (PSO) technique due to its evolutionary character. The key idea of the proposed approach is to consider the optimization vector as a swarm of real MAVs flying in the space of possible swarm distributions. During the evolution, the motion and relative localization constraints are tested in each transition between PSO populations as described in details in section III. The feasible trajectories of all MAVs can be then extracted directly from the progress of evolution of the best particle in the PSO population. Although the initial plan of the swarm deployment is obtained in a centralized way, the MAV motion into the obtained swarm distribution for the surveillance, may be realized using local controllers and information from onboard sensors only.

A. State of the art and progress beyond state of the art

The investigated task of swarm deployment for cooperative surveillance is related to the multi-robot coverage problem that was initiated for ground robots in the past decade (e.g. [3], [4]). For the surveillance task, multi-robot coverage was specified in [5], where the problem of positioning a team of mobile robots was solved with GPS. Approaches of biologically inspired swarm coverage can be found in [6], [7].

For MAVs, methods of a dynamic coverage (so-called continuous coverage problem), in which MAVs repeatedly follow a surveillance trajectory, can be found in [8], [9] and the static multi-MAV cooperative coverage, in which MAVs hover in fixed positions, in [10], [11]. The most relevant work to the proposed method is published in [10], where a distributed control strategy is designed for deploying the hovering MAVs with multiple downward facing cameras to collectively monitor an environment. The optimization criterion for the multi-camera placement problem is also the gained information per pixel from the defined areas for surveillance as it is proposed here. Nevertheless, the method in [10] relies on precise global localization (VICON motion capture system is used), which limits its utilization. In [11], a visual-SLAM algorithm is used for deploying a team of flying robots in surveillance coverage missions, where GPS is not available. In comparison with our approach, the method in [11] does not provide local coordination and stability of several MAVs that observe the same area, and it is focused mainly on the visual-SLAM method and spreading the team to maximize distances between the robots.

The evolutionary optimization methods (such as particle swarm optimization or genetic algorithms) are often used in robotics for trajectory planning [12], design of optimal controllers [13], and optimization of robot structures (morphologies of modular robots) [14], all with the aim of reaching a particular behavior whose efficiency is described by a fitness function. This function is then optimized with the aim of exploring a search space of parameters (trajectory parameters, parameters of controllers, joint angles in modular robots and manipulators etc.). In the proposed method, positions of all MAVs in the swarm are encoded into a unique solution of the evolutionary optimization. The fitness function describes efficiency of the swarm shape and its position with respect to the cooperative surveillance. In addition, swarm and MAV constraints are considered during the evolution.

The main contributions of this paper are the following:

- It presents a novel approach to multi-MAV cooperative surveillance, which is suited for outdoor deployment of closely cooperating MAVs flying in small relative distances.
- It provides a study of performance of the evolution based optimization method that uses tangible particles with motion constraints enforced by deployment of MAVs in a real world scenario.
- It shows an advantage of using a motion planning

technique integrated in the core of evolution based optimization methods.

The utilization of the optimization method inspired by the swarm intelligence for steering the real 3D swarms of MAVs brings interesting results, which were not observed in simulations of dimensionless swarm particles. Therefore, the proposed approach of swarm stabilization via the local sensory information may act as a test-bed for the study of swarming behaviour and evolution in artificial systems.

II. Problem statement and preliminaries

In the investigated scenario of autonomous cooperative surveillance, a set of AoI is given to a limited number n_r of autonomous robots (MAVs) with the aim to find a static swarm distribution to cover these areas (see Fig. 1 for an overview of the mission scenario). In addition, a set of no-fly zones, considered as obstacles for MAVs, is known prior to the mission. The areas of interest are gathered into an *AoI Map* and the workspace of robots with the no-fly zones is described in an *Environment Map*, as shown in section III-A. During the autonomous swarm deployment into the proper MAVs locations, the system has to respect motion, localization, and sensing capabilities of MAVs. In addition, constraints imposed by swarm stabilization based on relative localization have to be satisfied in both, the final static swarm distribution at the surveillance locations as well as during the swarm deployment from its initial location (a depot). In case of insufficient size of the swarm, which is not capable to cover the given set of locations of interest completely by its limited sensors, the set of AoI is covered to maximize the information collected together by the MAV group. The relevant part of AoI can be observed either by one MAV flying at an altitude, which is specified as the optimal altitude for the surveillance, or by several MAVs flying at higher altitudes that are observing the same area. The value of cost function, which indicates quality of the swarm distribution, depends on the overall amount of information gained together by all MAVs (details on the cost function implementation are presented in section III-A).

A. Quadrocopter model and control

A quadrotor vehicle model [15] with four identical propellers located at vertices of a square is used for the motion simulations, which are included in the core of the optimization process of the swarm deployment. As further explained in section III, the motion simulation in the optimization loop is crucial to ensure that the obtained trajectories are feasible with respect to MAV dynamics.

In the MAV model, each of the propellers j generate a thrust f_i^j along its axis. For each MAV i in the swarm, an inertial reference frame and a body-fixed frame with the origin located at the center of mass of the i-th MAV are considered. The relative positions of these frames are defined by the location of the center of mass $x_i \in \mathbb{R}^3$ in the inertial frame and by the rotation matrix $R_i \in \mathbb{R}^{3\times 3}$ from the body-fixed frame to the inertial frame.

The motion model of MAVs according to [15] is

$$\begin{aligned} \dot{x_i} &= v_i, \\ m_i \dot{v_i} &= m_i g e_3 - f_i R_i e_3, \\ \dot{R_i} &= R_i \hat{\Omega}_i, \\ J_i \dot{\Omega}_i + \Omega_i \times J_i \Omega_i &= M_i, \end{aligned} \tag{1}$$

where $v_i \in \mathbb{R}^3$ is velocity of the center of mass in the inertial frame, $m_i \in \mathbb{R}$ is mass of the MAV, $\Omega_i \in \mathbb{R}^3$ is the angular velocity in the body-fixed frame and $J_i \in \mathbb{R}^3$ is the inertia matrix with respect to the body frame. The hat symbol $\hat{}$ is defined by the condition $\hat{x}y = x \times y$ for all $x, y \in \mathbb{R}^3$, g is the gravity acceleration and $e_3 = [0; 0; 1]$. The total moment $M_i \in \mathbb{R}^3$ along all axes of the body-fixed frame and the thrust $f_i \in \mathbb{R}$ are control inputs. The total thrust, $f_i = \sum_{j=1}^{4} f_i^j$, acts in the direction of the axis of the body-fixed frame, which is orthogonal to the plane defined by the centres of the four propellers.

In the motion simulations within each iteration of the particle swarm optimization, the control inputs f_i and M_i are obtained using the tracking controller presented in [15]. In the presented experiments with quadrotors, it is supposed that the first body-fixed axis is parallel with the optical axis of the localization camera. The orientation of each MAV is then specified by position of the neighbour, which needs to be observed by the system of the visual relative localization (see [2] for details on the system).

B. Particle Swarm Optimization

The aim of the presented work is to verify possibility of utilization of optimization methods, which design relies on an evolution of populations of dimensionless particles, with tangible particles constrained by their motion and collision avoidance abilities. From the available population based optimization algorithms, the PSO method [16] was chosen. The PSO method aims to take advantage of the swarm intelligence to increase convergence of the optimization and to reduce probability of getting stuck in a local optima. In PSO, each solution (called particle) of the optimization problem represents a point in a multidimensional space of solutions. The group of particles is often called "PSO swarm", but we will rather use the term population, since the group of relatively stabilized MAVs is denoted as the swarm in this paper. In the proposed tangible PSO approach, each PSO particle represents the whole swarm (positions of all MAVs) and the population is then a set of swarms of MAVs.

The PSO algorithm is suited for the swarm deployment problem due to its ability to deal with nonlinear functions with many local extremes in a space of a high dimension. In addition, the evolutionary approach of PSO enables to track and remember the progress of the solution during the optimization. The essence of the PSO method is inspired by a social behavior of birds and fish (by movement of their flocks and schools). Therefore, it offers the possibility to consider the movement of swarms of MAVs analogically as the movement of particles in PSO. The required movement of the

MAV swarm (the complete solution of the swarm deployment task) is then obtained from the history of evolution of the best particle in the search space (the space of MAVs swarm distributions).

During the optimization, movement of each particle is influenced by its own experience and also by a social knowledge. Each particle remembers its state with the best function value (cost function) visited so far; later called the individual best position of j-th PSO particle and denoted as b_j. The social knowledge is represented by the best position visited by a particle in the whole population; later called the global best position and denoted b_g. In fact, the global best position can be obtained as $b_g = \text{argmin}_{j\in\{1\cdots n_p\}} CF(b_j)$, if we are looking for minimum of the cost function $CF(\cdot)$. Constant n_p is the number of particles in the PSO population.

In the PSO algorithm, each particle j is represented by its D-dimensional position $p_j(t)$ and its current velocity vector $u_j(t)$. The velocity vectors of all particles are updated by the following rule in each iteration t:

$$u_j(t) = u_j(t-1) + \Phi_1(b_j - p_j(t-1)) + \Phi_2(b_g - p_j(t-1)), \quad (2)$$

where Φ_1 and Φ_2 are obtained as

$$\Phi_k = c_k \begin{pmatrix} ran_{k,1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & ran_{k,D} \end{pmatrix}, k \in \{1,2\}. \quad (3)$$

Influence of particles' own experience and the social knowledge is weighted by constants c_1 and c_2. Random numbers $ran_{1,1}, \cdots, ran_{1,D}$ and $ran_{2,1}, \cdots, ran_{2,D}$ are obtained with a uniform distribution between 0 and 1.

In the second step of the PSO iteration, the positions of all particles are updated as

$$p_j(t) = p_j(t-1) + u_j(t). \quad (4)$$

The velocity and position update rules are applied in each PSO iteration. Maximum number of iterations t_{max} is used as the stopping criterion in the proposed approach.

Two commonly used extensions of the velocity update rule are tested in the algorithm performance analyses in section IV-A. 1) A limit on velocity of particles, which is applied if any component of the result of the velocity update rule $u_j(t)$ exceeds a given constant u_{max}. Then the exceeding vector components are replaced by u_{max}. 2) A linear decrease of the inertia of particles from an initial value w_{start} to a value w_{end} at the end of the PSO process. It ensures that the influence of the velocity vector from the previous iteration is decreasing during the optimization process. The stored velocity from the previous iteration is then updated before the new calculation of the velocity update rule as

$$u_j(t-1) := \left(w_{start} + t\frac{w_{end} - w_{start}}{t_{max}}\right) u_j(t-1). \quad (5)$$

III. Tangible Particle Swarm Optimization

We propose to solve the problem of deployment of MAVs swarm for the static surveillance by a modified PSO with integrated swarm constraints. In the PSO population, the j-th PSO particle is formed as a set of 3D positions of all MAVs in the swarm: $p_i = [x_1, y_1, z_1, x_2, y_2, z_2, \cdots, x_{n_r}, y_{n_r}, z_{n_r}]$, where n_r is number of employed MAVs. The PSO population is composed of n_p such virtual swarms as visualized in Fig. 2. The dimension of the space of solutions is then equal to $3n_r$.

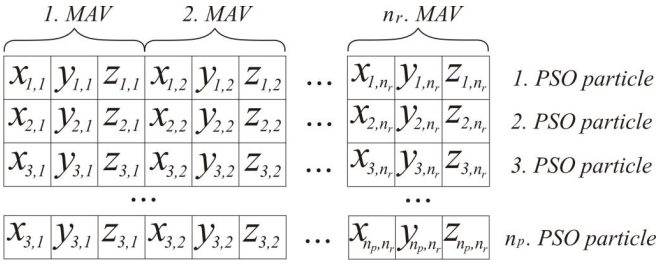


Fig. 2. Scheme of the PSO population.

The aim of the PSO optimization is to find a feasible static shape of the swarm (locations of particular swarm entities encoded in the position of the best PSO particle). Moreover, the important output of the algorithm is a feasible plan of the MAVs movement from the initial configuration to this target shape (trajectories for all MAVs are encoded in the history of the best PSO particle evolution). It leads us to a swarm shape optimization with the necessity of keeping the history (a feasible MAV movement) of the evolution of the swarm shape from its initial state. As mentioned, the constraints of the real swarm are considered during the evolution of the population as they have to be satisfied for all obtained trajectories between the initial location of MAVs into the found swarm distribution.

In comparison with the standard PSO, where the velocity and position update rules are simply applied in each PSO iteration, in the proposed tangible PSO, the simulation of the MAV swarm motion is used to verify that the transition between two consequent positions $p_j(t-1)$ and $p_j(t)$ of the j-the PSO particle is feasible. A simple scheme of such an approach is shown in Fig. 3. In each iteration, the new positions of PSO particles (and therefore positions of virtual swarms) are proposed based on the PSO update rules. In the next step, mutual collisions in the proposed positions of MAVs are found in all virtual swarm separately. If the i-th MAV would collide in its new position with another MAV, the part of the PSO velocity corresponding to the i-th MAVs is decreased to stop the i-th MAV before the collision. This correction mechanism is repeated for all MAVs of all virtual swarms. Similarly, the relevant component of the PSO velocity vector is decreased if a new position lies inside an obstacle or outside the operational area. The operational workspace of the swarm and the obstacles (the set of no-fly zones) are denoted by the mission operator as a set of convex polygons. These polygons are dilated with a safety

radius (defined by size of MAVs and its operational limits) and gathered in the *Environment Map* as shown in Fig. 5.

In the following step, the actual positions of swarms and the updated new positions are used as inputs of a motion planning and coordination algorithm. The presented results have been obtained with a fast multi-MAV motion planning method based on 3D visibility graph, since the time complexity is a crucial aspect in the proposed swarm deployment method (the motion planning problem has to be solved in each iteration for each MAV of each virtual swarm). In the algorithm, mutual collisions between paths obtained by the visibility graph technique [17] for each of the MAVs are identified and resolved by an iterative permutation of target locations assigned to MAVs being involved in the detected collision. This simple but very efficient approach is allowed since the MAVs in the swarm are considered as equivalent entities. It was experimentally verified that 96.8% of all collisions detected during the PSO optimization can be resolved this way. The statistics was obtained from 200 runs of the PSO algorithm with 40 particles and 70 iterations, which means that the motion planning and coordination algorithm was tested in 560000 runs. The remaining approximately 3.2% of situations (usually if more than one collision is detected in a plan of a pair of MAVs) are again resolved by shortening the part of the PSO velocity vector corresponding to the MAVs that are involved in the detected possibility of the collision. The velocity is shortened in such a way that the updated new position of the MAV is in a sufficient distance before the location of the detected collision. Due to the rare occurrence of such a situation, the convergence of the PSO process is decreased only slightly. Nevertheless, any exhaustive multi-robot coordination approach may be utilized in applications where the convergence decrease would be unacceptable, but usually at the cost of growth of computational complexity.

The plan obtained by the MAVs planning and coordination algorithm is followed in a simulation using the trajectory tracking mechanism [15] with the MAV model introduced in section II-A. The simulation is run until the desired PSO positions are reached or a violation of swarm constraints is detected. If a violation of the relative localization constraints (range, viewing angle, mutual MAV heading etc.) is detected, the simulation is reversed into the last state that is still considered as the feasible swarm distribution. The corresponding part of the PSO particle position vector is replaced by this position and also the particle's velocity is updated as

$$u_j(t) = p_j(t) - p_j(t-1), \tag{6}$$

where $p_j(t)$ is the updated position vector and $p_j(t-1)$ is the previous particle's position, which is used as the initial state of the motion planning algorithm. The uncertainty added into the PSO rules in equation (3) is crucial to increase the probability that the optimization will not end up in the same constraints violation, but it will escape from this potential deadlock. The new achieved (or updated) position of the PSO particle is evaluated by the cost function, introduced in section III-A, and the PSO algorithm continues in the next iteration from this state.

In addition to the localization constraints, also the motion constraints and the collisions are tested during the simulation. Although the visibility graph technique provides collision-free solutions, the obtained paths composed of straight segments cannot be passed by MAVs flying with a reasonable velocity without any deviation. In the above mentioned statistics of 200 runs of the PSO algorithm with 40 particles and 70 iterations, a collision with obstacles due to the deviation from the followed path was detected in 0.06% of particles' evaluations. The prediction of an impending collision again resulted in the change of the components of the PSO vector in eq. (6).

Another solution of the above mentioned problems (the unresolved collisions and the deviations from the simple path) would be utilization of any sufficiently fast technique suited for trajectory planning and coordination of multiple MAV systems. For example Rapidly-exploring Random Tree (RRT) based approaches [18] or the algorithm in [19], which is well suitable for quadrotors, seem to be especially appealing. Nevertheless, even the utilization of simple visibility graph technique enables us to show usefulness of motion planning integrated within the evolutionary optimization. Such an algorithm enables to steer MAVs directly using results of the optimization. This can be understood as a novel approach to multi-objective optimization, where a motion planning technique is integrated directly into the core of the optimization engine.

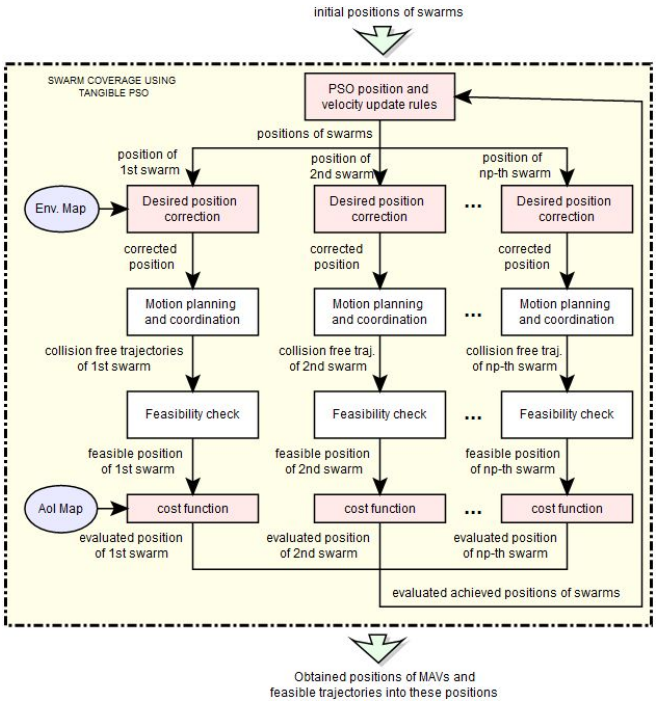


Fig. 3. Scheme of the planning system for the environment coverage by MAV swarms stabilized with the visual relative localization.

A. Cost function evaluation

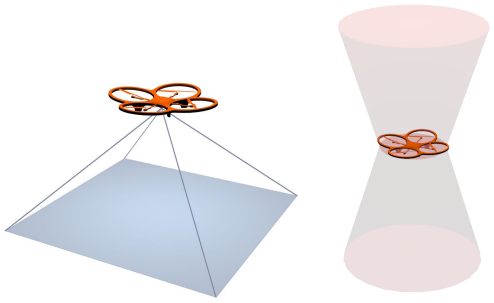
The evaluation of PSO particles in each iteration of the evolution, which is necessary for determination of the global best and individual best values, depends on the position of all MAVs in the swarm relative to the AoI. The *AoI Map*, which describes the set of areas of interest, is used to effectively compute the cost describing quality of the particular solution (positions of all MAVs of the swarm). An example of construction of the *AoI Map* is shown in Fig. 5. Similarly as the obstacles and borders of the operational area, also the borders of the desired areas of interest have to be specified by the mission operator. These areas are described as a set of polygons and circles in experiments presented in this paper.

The *AoI Map* is then represented as a matrix $A \in \mathbb{R}^{a \times b}$ (see Fig. 5(c) for example) with values of its elements equal to 0 or to a constant A_{max}. The zero elements represent regions not assigned as AoI, while the cells occupied by AoI are evaluated with the A_{max} value. The number of elements of A, which is determined by constants a and b, depends on the application. As usual, if the shape of AoI does not need to be determined precisely, a sparse grid with smaller constants a and b can be used, which reduces computational complexity of the particle evaluation.

The cost function evaluating particular solutions of the swarm deployment problem, as shown in Fig. 3, can be then expressed as:

$$CF(X) = \sum_{x=1}^{a} \sum_{y=1}^{b} \min \left(0, A_{x,y} - \sum_{i \in Swarm(x,y)} \frac{S_{opt}}{S_i} A_{max} \right), \tag{7}$$

where S_i is area of the rectangle representing the part of the workspace that can be observed by the i-th MAV. The size and position of the rectangle depend on the viewing angle of the employed sensor, actual altitude of the i-th MAV and its Pitch and Roll angles. S_{opt} is area of the region, which is observed by MAV flying at the altitude determined as the "optimal" altitude based on the particular application (this altitude is usually specified by the security expert). We assume that MAV flying at the "optimal" altitude provides a sufficient amount of information on the part of AoI, which is covered by its surveillance sensor. An MAV at a lower altitude does not gain more information per square unit. On the contrary, we suppose that the information per square unit decreases linearly with increasing area S_i (the total amount of gained information from the sensor is constant). $Swarm(x, y)$ in equation (7) is a subset of all MAVs of the swarm. For each MAV from $Swarm(x, y)$, it has to be satisfied that the cell of workspace represented by the element $A_{x,y}$ of the *AoI Map* is completely observable by its surveillance sensor. The required information from the particular element $A_{x,y}$ can be then obtained by a single MAV flying at the "optimal" altitude or cooperatively by a set of MAVs flying at higher altitudes. The sum of remainders (after subtraction of contributions of all MAVs) of the initial values A_{max} assigned to each cell $A_{x,y}$ occupied by AoI is considered as the cost value of the particular solution.



(a) Model of the surveillance camera used in the cost function of the swarm motion planning. (b) MAV model employed in the collision avoidance.

Fig. 4. Models used in the PSO planning loop.

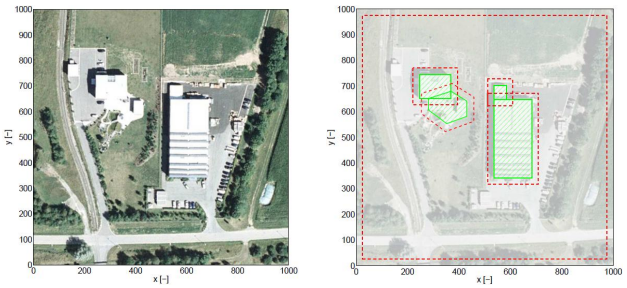
In such defined objective function, the mission operator can easily require surveillance of particular area of map via setting of a none zero value of $A_{x,y}$ of the map cell x, y. In the cost function, this value is decreased in current solution of the swarm distribution if it is observed by one or more MAVs. The reduction of the value depends on the altitude of each MAV observing the area. MAVs flying in lower altitude decrease this value more significantly (they gain more information during the surveillance). The required zeroing of $A_{x,y}$ may be then obtained by one MAV flying in lower altitude or by more MAVs flying higher. The cost function value is represented as sum of remainders $A_{(\cdot,\cdot)}$ of all cells in the map. This forces the optimization engine to find such distribution of the swarm, which covers the required areas of interest as much as possible.

Finally, we should emphasize that the proposed method does not guarantee to find the optimal distribution of the swarm and the optimal trajectories from initial positions into the found locations, but the feasibility of the solution with respect to the motion and localization constraints is guaranteed. Regarding the presented relative visual localization, it is important that the plan of the swarm deployment in the environment satisfies constraints given by the range of the relative localization, viewing angle of the on-board cameras and it respects mutual MAVs heading.

IV. Experimental results

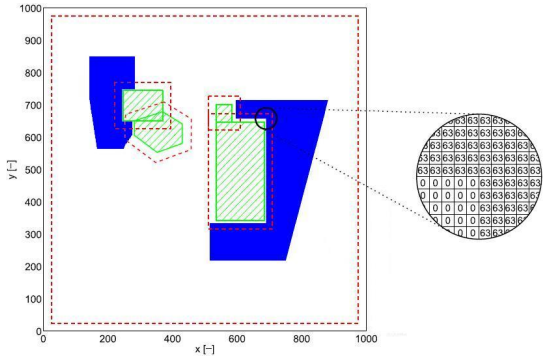
A. Algorithm parameters tuning and performance analyses

The performance of the particle swarm optimization markedly depends on the setting of its parameters. Mainly the convergence of the optimization, and therefore the necessary number of iterations to achieve a sufficient solution, may vary based on proper algorithm setting. In most of the PSO applications, this aspect is important due to the computational time constraints. Here, in addition to the common requirement on the low computational time, the smaller number of iterations means also simpler trajectories with less



(a) Satellite map of an industrial area that has to be under surveillance of MAVs.

(b) Highlighted obstacles and borders of dilated no-fly zones.



(c) Final map with depicted areas of interest.

Fig. 5. An example of construction of a map with no-fly zones (obstacles) and areas of interest.

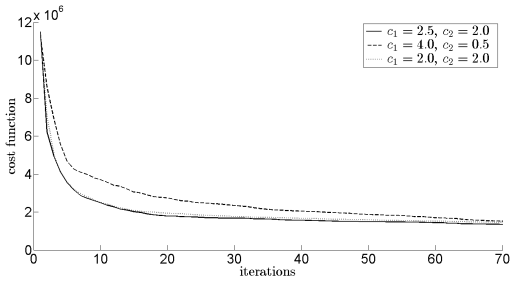


Fig. 6. Progress of cost function values. Influence of different values of PSO parameters c_1 and c_2 on the algorithm performance.

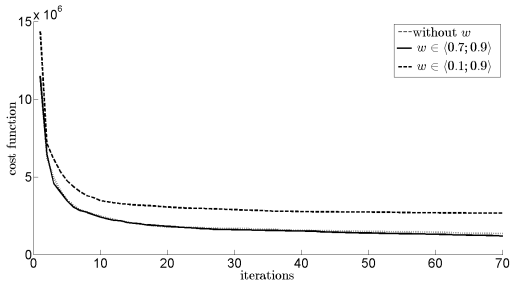


Fig. 7. Progress of cost function values depending on the value of inertia w used in the planning algorithm.

waypoints that have to be flown through. Such trajectories can be realized faster or more efficiently postprocessed to achieve an optimal motion plan to reach the obtained swarm distribution.

Numerous statistically evaluated experiments were realized to analyse the sensitivity of the algorithm on its setting (see Fig. 11 for the testing scenario used in these experiments). In the experiments presented in this section, the average values of cost function were obtained always from 100 runs of the algorithm for each set of PSO parameters. The most dominant effect is caused by changing c_1 and c_2 parameters from eq. (3). For example, the algorithm with parameters $c_1 = 2.5$ and $c_2 = 2.0$ would need 30 iterations (in average) to obtain the same solution as it is obtained after 60 iterations with setting $c_1 = 4.0$ and $c_2 = 0.5$, see Fig. 6. It was also observed that the algorithm performance is affected only slightly by a small deviation from these optimal values (see almost the same progress of mean cost values with setting $c_1 = 2.0$ and $c_2 = 2.0$) and that the results do not depend on the employed testing scenario (its scale, complexity etc.).

An interesting result was achieved from statistical testing the influence of the decrease of inertia w (defined in eq. 5) during the PSO process on the algorithm performance. As shown in Fig. 7, this often used improvement of the basic PSO rules has a minimal positive effect. The algorithm with the best tuned values of the inertia decrease has only slightly better performance than without considering the decrease of the inertia (constant $w = 1$). The performance of the algorithm is significantly worse if the inertia is deviated from the best tuned values. The best performance was achieved if the inertia is linearly decreasing from value $w_{start} = 0.9$ at the beginning of the PSO process to $w_{end} = 0.7$ at the end, but the difference of the performance compared to the performance of the algorithm with constant value $w = 1$ is minor. On the contrary, for example, the algorithm with setting $w_{start} = 0.9$ to $w_{end} = 0.1$, which is often recommended as the optimal setting in literature, provides significantly worse performance in comparison with the basic version with constant inertia (see the progress of mean cost function value in Fig. 7). The reason of this mostly negative influence of the inertia, which is not common in the classical PSO with dimensionless particles, could be the utilization of the motion model and localization constraints that limits MAVs. Similar results may be seen if the velocity of particles is "artificially" limited by the parameter u_{max}. Also utilization of u_{max} either decreases the PSO convergence if tuned improperly or has a minimal effect. Again, the reason can be that the influence of the parameter u_{max} overlaps with the motion constraints of MAVs, which are integrated in the optimization. The compactness of the swarm, which has to be artificially kept by further improvement of the velocity rule in PSO with dimensionless particles, is obtained as a side effect of the relative localization constraints in the proposed tangible PSO. To conclude this parameters tuning, let us choose the setting of the PSO parameters proper for utilization of the tangible particles, which will be employed in the following experiments, as $c_1 = 2.5$, $c_2 = 2.0$, $u_{max} = inf$ and $w = 1$.

Let us now analyse the influence of the constraint on the

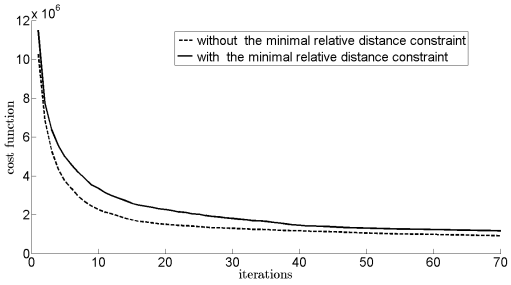


Fig. 8. Progress of cost function values. Comparison of performance of the algorithm that considers dimensionless PSO particles and the variant with tangible PSO particles.

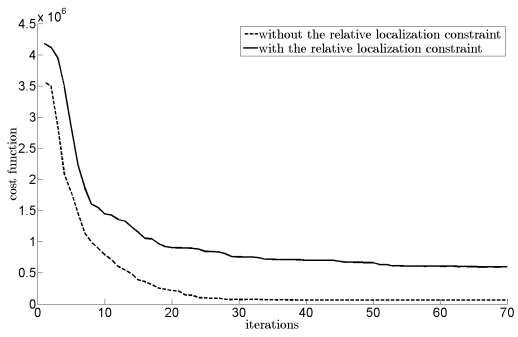


Fig. 9. Comparison of the algorithm performance with and without the relative localization constraint. Progress of mean of cost function values was obtained from 100 randomly initialized runs of the algorithm.

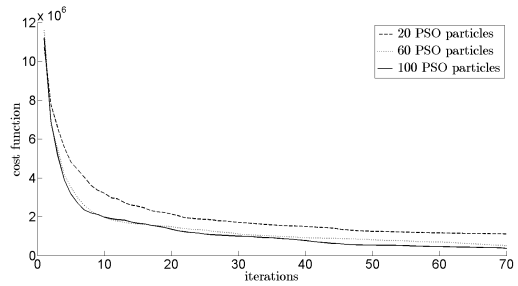


Fig. 10. Progress of cost function values depending on the number of PSO particles used in the planning algorithm.

minimal distance between PSO particles, which has to be used due to the collision avoidance and to protect MAVs from mutual air disturbances. The algorithm that considers the constraints imposed by the real size of MAVs and their mutual influence is compared with the simple PSO method that does not consider collisions between obstacles in Fig. 8. The convergence of the PSO process is decreased due to the collision avoidance constraint mainly in the initial phase. During the exploration of the workspace, the evolution of particles (their motion) is more chaotic and the probability of appearance of several MAVs close to each other is greater. Therefore, the velocity of PSO particles has to be often shortened to avoid the collisions and the PSO convergence is decreased. In the following exploitation phase, the swarm is already spread to effectively cover the areas of interest and an MAV gets close to its neighbor only rarely.

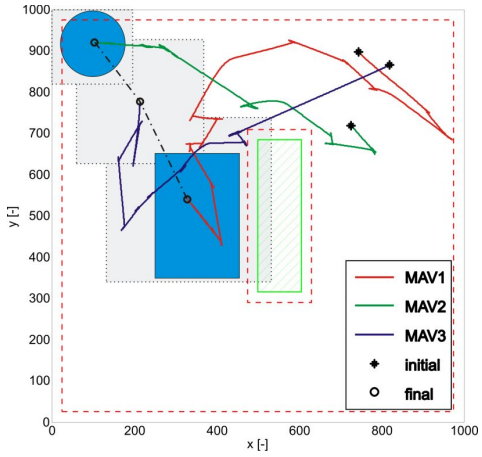
The relative localization constraint affects the algorithm performance more significantly (see Fig. 9). The convergence of the PSO process that takes this constraint into consideration is difficult to compare with the convergence of the PSO algorithm without the localization constraint included, since both variants solve significantly different optimization problems. The solution of the swarm spreading without the relative localization is often not feasible if the localization constraint is considered. Moreover, a solution obtained in the simple variant can be unreachable by the swarm stabilized under the relative localization even if the particular solution satisfies the localization constraints, as shown in Fig. 14 and Fig. 13.

In the last analyses of the algorithm performance, the evolution of the PSO population was evaluated with different number of employed particles (Fig. 10). The localization and collision avoidance constraints were integrated into the planning of deployment of swarms that consist of three MAVs. The influence of the number of PSO particles on the convergence of the optimization is in the proposed tangible PSO similar to the classical PSO with dimensionless particles. The rate of convergence of the algorithm is increased with the number of particles mainly in the initial exploration part of the PSO process, since the probability of finding a perspective part of the search space growths with the number of particles. During the consequence exploitation part of the optimization, the influence of the number of particles decreases as the PSO population is more shrunk and some of the particles appear in the same position and may be considered as redundant. Usually, a trade-off between the number of PSO particles and the number of allowed iterations is chosen with the aim to satisfy a computational intensity constraint. Here, also the aim to obtain a short solution has to be considered, which results in the effort to minimize the number of iterations as mentioned above. In the following experiments, PSO populations with 40 particles are used, since another increase of the group size has a minimal effect to the algorithm performance.

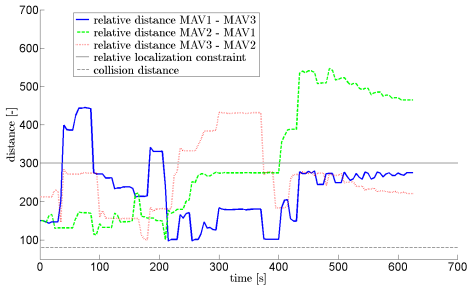
B. Visualization of results of simulations of the swarm deployment

In the first simulation, a swarm of 6 MAVs is deployed in a simple environment without obstacles. In the planning of the swarm deployment, each MAV has to keep the localization linkage with at least two neighbors. In the visualization of the swarm movement into the found configuration, each MAV is connected with other MAVs by red lines if the particular pair is in the distance, which satisfies the constraint of the relative localization. Four circular areas of interest are designed in the environment. The group is randomly initialized in a circle representing a starting area of robots. The initial positions satisfy the constraint on the minimal distance between MAVs (the safety distance). The diameter of the starting area is equal to the maximal distance that satisfies the localization constraint.

The PSO optimization was run in 20 iterations with 40

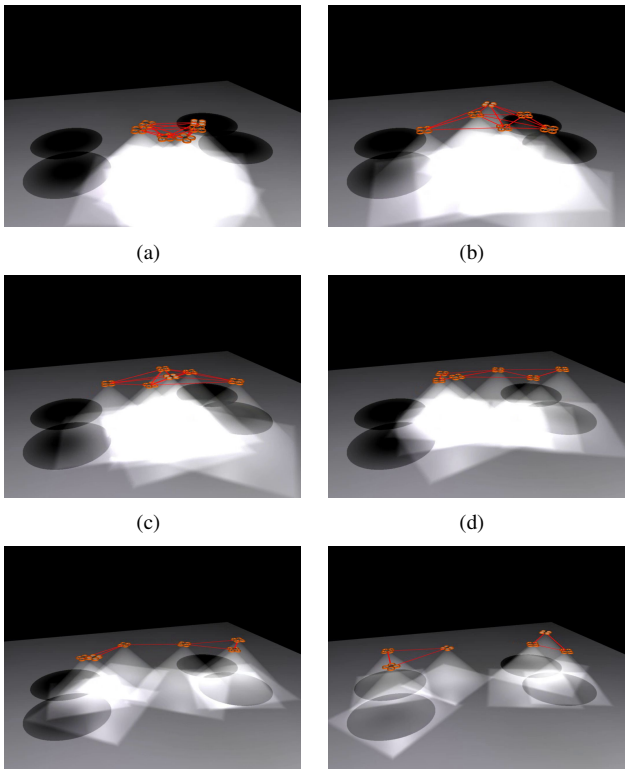


(a) The obtained trajectories depicted in the workspace with the areas of interest and obstacles



(b) Relative distances between the MAVs. The relative distance between MAVs shorter than 70 map units (the lower dotted threshold) is considered as infeasible. Always at least one neighbor of each MAV has to be closer than 300 map units (the upper dotted threshold) to satisfy the relative localization constraint.

Fig. 11. The testing scenario used for the analyses in Fig 6-10 and an example of planned trajectories if the relative localization constraint was employed.



(a) (b) (c) (d) (e) (f)

Fig. 12. Simulation with a swarm of 6 MAVs.

TABLE I

COMPARISON OF THE PERFORMANCE OF THE PSO ALGORITHM WITH AND WITHOUT THE MOTION PLANNING INTEGRATED INTO THE PSO OPTIMIZATION. THREE SCENARIOS WITH DIFFERENT REQUIRED NUMBER OF NEIGHBORS THAT HAVE TO BE KEPT UNDER THE RELATIVE VISUAL LOCALIZATION ARE CONSIDERED.

algorithm specification		collisions	deadlocks	mean cost
0 neighbor	without MP	98%	31%	2.3×10^5
	with MP	1.3%	0%	1.0×10^5
1 neighbor	without MP	100%	73%	9.2×10^5
	with MP	4.5%	0%	2.4×10^5
2 neighbors	without MP	100%	81%	6.7×10^7
	with MP	9.8%	0%	3.8×10^6

PSO particles. The swarm distribution with the lowest cost function was found in the 17-th population. Progress of the evolution of the best particle, the particle with the same personal best value as is the obtained global best value, is shown by the motion simulation in Fig. 12. Only the first 17 iterations are considered as the result of the swarm shape optimization with the trajectory planning. During the movement simulation, the swarm is emergently splitted into two groups to cover both pairs of the locations of interest. The splitting was not preprogrammed, but it autonomously appeared during the optimization. The obtained sub-swarms form triangles, in which the requirement on two neighbors in the localization distance may be satisfied independently to the second group.

In the second scenario with 3 MAVs, a non-fly zone was added to verify the obstacle avoidance function included into the swarm shape optimization. It also shows importance of the path planning algorithm, which is included in the core of the optimization. The PSO process without the path planning gets stuck due to the constraint which ensures that MAVs do not enter into the zone. Often, all MAVs end up on the border of the obstacle, while the PSO rules repeatedly force them to continue inside. Such a deadlock significantly decreases convergence of the optimization or even freezes it if the random character of PSO does not enable to release from this local extreme as shown in table I. The percentage of runs of the algorithm that were finished in a deadlock (the movement of PSO particles was stopped due to the non-fly zone) is stated in the fourth column of the table. The third column of the table denotes the number of PSO runs in which a collision with the non-fly zone was detected in the *Feasibility check* block. The last column shows average cost value of solutions achieved after 70 iterations.

An example of a solution found by the algorithm with the path planning included is shown in Fig. 14. In this scenario, three MAVs are considered and each MAV has to be

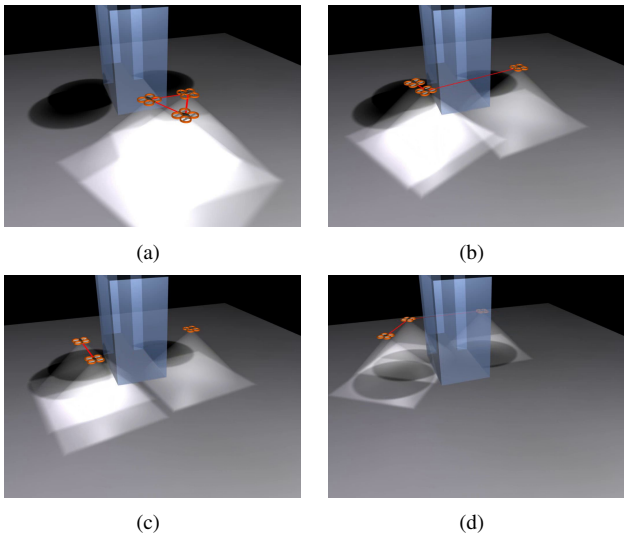


Fig. 13. Simulation of swarm movement into a location found by a simple state space search. The solution does not guarantee that the direct visibility is kept during the movement.

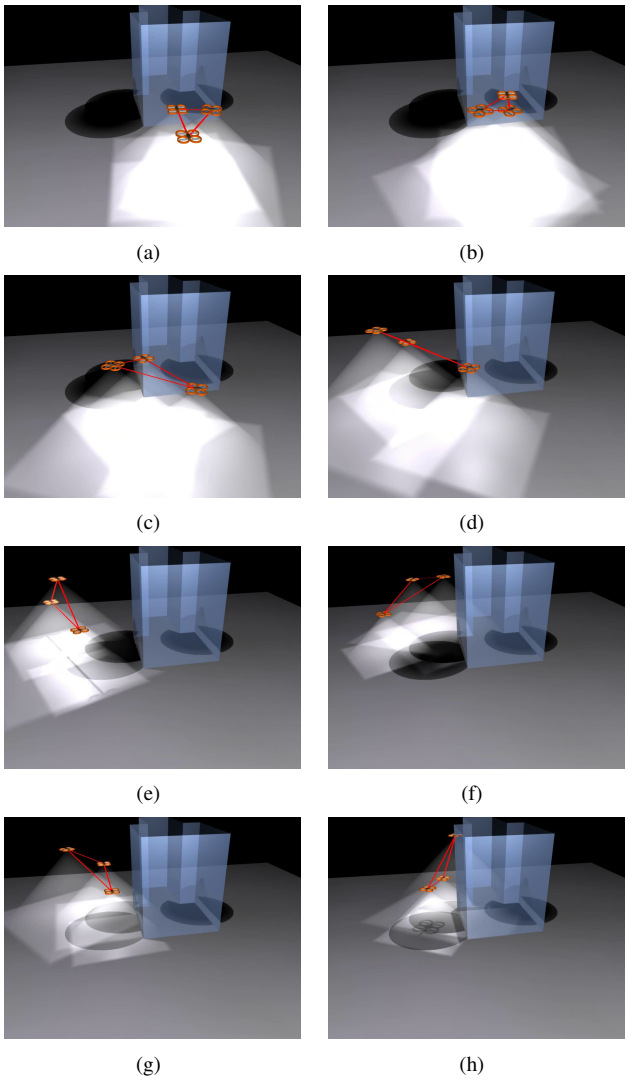


Fig. 14. Simulation with a swarm of 3 MAVs.

connected with at least one neighbor through the localization linkage during their movement into the surveillance location. Using a standard state-space search, which considers only the quality of the final swarm position, but not the existence of feasible trajectories to reach such a swarm distribution, a solution with MAVs flying each above one of the areas of interest can be found (see Fig. 13). Nevertheless, the size and position of the non-fly zone do not allow to reach this solution by the swarm flying from the initial depot, while keeping the localization constraints. As shown in Fig. 13(c), the non-fly zone is too wide to enable the required relative localization of MAVs following its border each on the opposite site.

The solution found by the proposed PSO-based method does not cover the areas of interest completely, as shown in Fig. 14, but the existence of feasible trajectories from the initial state of the swarm is guaranteed. The evolution of the best particle (the output of the method) is visualized in the sequence of snapshots in Fig. 14 and in the movie in [20]. One can see that the localization constraints (denoted by the red lines) are satisfied during the experiment.

Finally, let us present an example of control inputs and state variables of one of the MAVs of the group during its flight between two consequent positions generates by the PSO rules. In the selected trajectory segment, no obstacles or interferences with neighbors occur and the MAV can follow a straight path simply connecting the actual and desired positions. The actual position is $[x = 0, y = 0, z = 0]$ and the desired position is $[2, -10, 1]$. Fig. 15 and 16 show x, y, z positions, Euler angles, velocities and control inputs (moments and forces) of the MAV reaching the desired position proposed by the PSO algorithm. Although the MAV is navigated into the desired location relatively smoothly by the trajectory tracking mechanism, the overshoots in the position behind the coordinates $[2, -10, 1]$ shown in Fig. 15 have to be considered in the obstacle avoidance constraints.

C. Experimental evaluation of the planning technique in flight conditions.

The aim of the experiment in Fig. 17-19 is to verify the ability of the onboard vision system to keep the relative localization linkages between the neighboring MAVs during execution of the plan obtained by the proposed swarm deployment method. Notice that the control feedback of MAVs is realized by the Vicon motion capture system, which is also used as a ground truth for evaluation of the performance of the relative localization system in real-flight conditions. In addition, the feasibility of the obtained plan of swarm deployment in the environment with known sets of areas of interest, no-fly zones and initial positions of MAVs is verified in the experiment. The plan, obtained by the proposed PSO based method prior the experiment, satisfies constraints given by the range of the relative localization, viewing angle of the on-board cameras, mutual MAVs heading and movement constraints during the deployment of the system. Fig. 19(e)

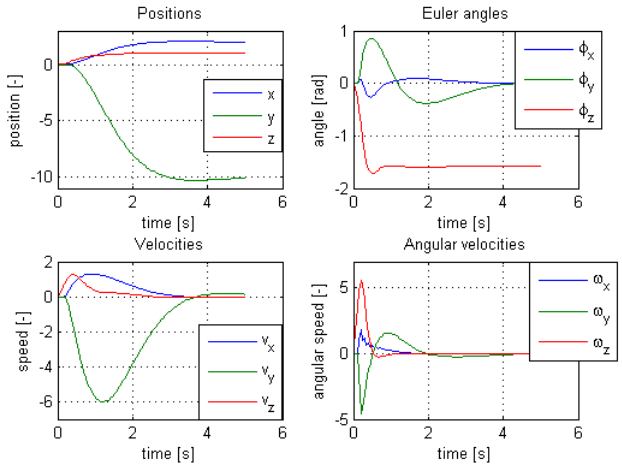


Fig. 15. Progress of positions, velocities, Euler angles and angular velocities during flight between two consequence positions of an MAV generated by the PSO rules.

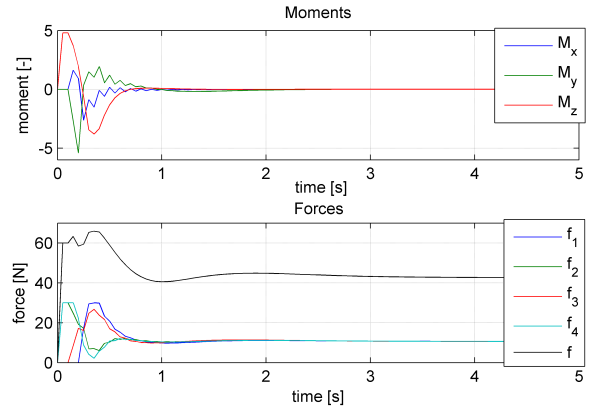


Fig. 16. Progress of moments and forces used as the inputs of the MAV model in the movement simulation from Fig 15.

shows that the guess of relative positions of neighbouring vehicles is continuously provided during the flight and that the limit on the relative position between the robots of the team (2.5m) is kept.

V. Conclusion

A novel approach for deployment of multi-MAV teams in the application of cooperative surveillance is presented in this paper. The proposed method based on visual onboard relative localization is verified in a feasibility study approved by numerous simulations and hardware experiments demonstrating utilization of the swarm deployment in various scenarios. In addition, the paper provides a study of performance of the evolutionary optimization method that uses tangible particles instead of commonly used dimensionless particles. During the optimization, the movement of particles satisfies motion constraints enforced by deployment of MAVs in the real world missions. The utilization of the optimization method inspired by the swarm intelligence for steering the real 3D swarms of MAVs brings interesting results, which were not observed in simulations of dimensionless swarm particles.

Fig. 17. Experiment with swarm of 3 MAVs following trajectories obtained off-line by the proposed planning algorithm. MAVs are denoted by circles of different colours. (taken at GRASP laboratory, University of Pennsylvania facility)

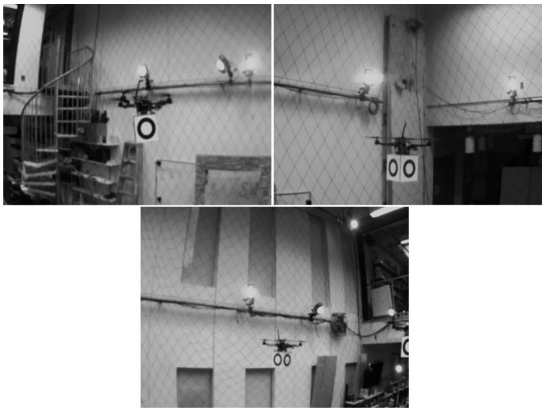


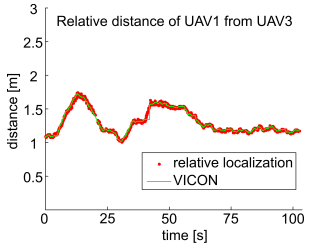
Fig. 18. Pictures taken by the onboard localization systems of all MAVs in the same moment (experiment in Fig. 17).

VI. Acknowledgements

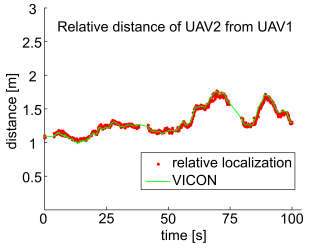
This work was supported by MŠMT under project Kontakt II no. LH11053 and by GAČR under M. Saska's postdoc grant no. P10312/P756

References

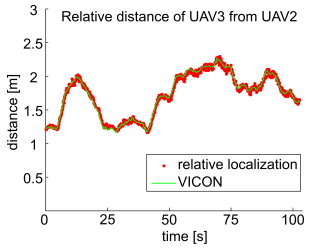
[1] T. Krajnik, M. Nitsche, J. Faigl, P. Vanek, M. Saska, L. Preucil, T. Duckett, and M. Mejail, "A practical multirobot localization system," *Accepted by Journal of Intelligent & Robotic Systems*, 2014.



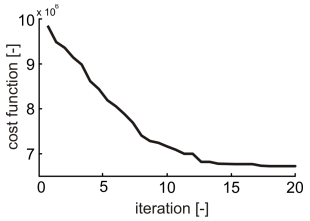
(a) Comparison of relative distances between MAVs captured by the onboard vision system and by Vicon motion capture system.



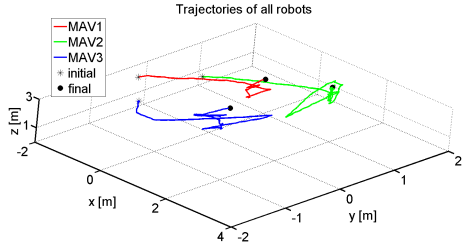
(b) Comparison of relative distances between MAVs captured by the onboard vision system and by Vicon.



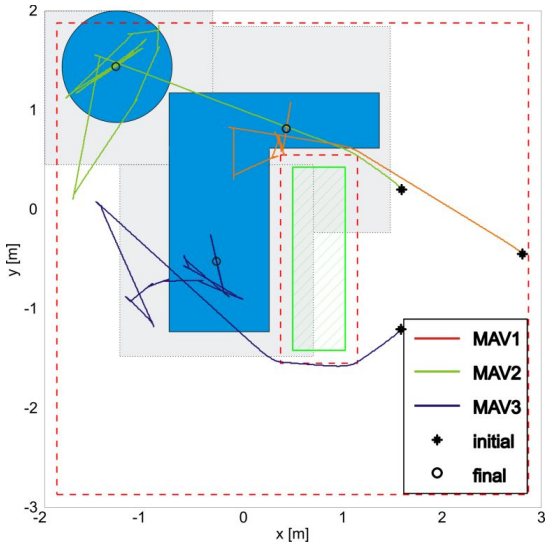
(c) Comparison of relative distances between MAVs captured by the onboard vision system and by Vicon.



(d) Progress of cost function values of the best PSO particle during the off-line optimization of the swarm deployment found for the experiment in Fig. 17.



(e) 3D view of positions of MAVs captured by Vicon during the experiment.



(f) Positions of MAVs captured by Vicon during the experiment, with denoted areas of interest and the non-fly zone.

Fig. 19. Swarm deployment in the environment to cover selected areas of interest. (experiment in Fig. 17)

[2] J. Faigl, T. Krajník, , J. Chudoba, L. Preucil, and M. Saska, "Low-cost embedded system for relative localization in robotic swarms," in *Proc. of IEEE International Conference on Robotics and Automation*, 2013.

[3] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 2, pp. 243–255, 2004.

[4] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. Kosmatopoulos, "Adaptive-based distributed cooperative multi-robot coverage," in *American Control Conference (ACC)*, 2011.

[5] ——, "Adaptive-based, scalable design for autonomous multi-robot surveillance," in *IEEE CDC*, 2010.

[6] E. Mathews, T. Graf, and K. S. S. B. Kulathunga, "Biologically inspired swarm robotic network ensuring coverage and connectivity," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, Oct 2012, pp. 84–90.

[7] J.-z. Liu, B.-l. Wang, J.-y. Ao, S. Wang, and Q. Wu, "An immune-swarm intelligence based algorithm for deterministic coverage problems of wireless sensor networks," *Journal of Central South University*, vol. 19, no. 11, pp. 3154–3161, 2012.

[8] I. Maza and A. Ollero, "Multiple uav cooperative searching operation using polygon area decomposition and efficient coverage algorithms," in *Distributed Autonomous Robotic Systems 6*, R. Alami, R. Chatila, and H. Asama, Eds. Springer Japan, 2007, pp. 221–230.

[9] T. Ha, *The UAV Continuous Coverage Problem*. Air Force Institute of Technology, 2010.

[10] M. Schwager, B. J. Julian, and D. Rus, "Optimal coverage for multiple hovering robots with downward facing cameras," in *IEEE ICRA*, 2009.

[11] L. Doitsidis, A. Renzaglia, S. Weiss, E. Kosmatopoulos, D. Scaramuzza, and R. Siegwart, "3d surveillance coverage using maps extracted by a monocular slam algorithm," in *IEEE/RSJ IROS*, 2011.

[12] M. Saska, M. Macas, L. Preucil, and L. Lhotska, "Robot path planning using partical swarm optimization of ferguson splines," in *IEEE ETFA*, 2006.

[13] J.-B. Mouret and S. Doncieux, "Encouraging behavioral diversity in evolutionary robotics: an empirical study," *Evolutionary Computation*, vol. 20, no. 1, pp. 91–133, 2012.

[14] V. Vonasek, M. Saska, K. Kosnar, and L. Preucil, "Global motion planning for modular robots with local motion primitives," in *IEEE ICRA*, 2013.

[15] T. Lee, M. Leoky, and N. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *49th IEEE Conference on Decision and Control (CDC)*, 2010.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings International Conference on Neural Networks IEEE*, vol. 4, 1995, pp. 1942–1948.

[17] J. O'Rourke, *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[18] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning." in *TR 98-11, Computer Science Dept., Iowa State University*, 1998.

[19] M. Turpin, N. Michael, and V. Kumar, "Concurrent assignment and planning of trajectories for large teams of interchangeable robots," in *International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013.

[20] Movies, "Movies of experiments of the mav cooperative surveillance. http://imr.felk.cvut.cz/mavsurveillance/ [online] [cit. 2014-12-02]."