Angle-Encoded Swarm Optimization for UAV Formation Path Planning

V.T. Hoang, M.D. Phung, T.H. Dinh, Q.P. Ha

Abstract— **This paper presents a novel and feasible path planning technique for a group of unmanned aerial vehicles (UAVs) conducting surface inspection of infrastructure. The ultimate goal is to minimise the travel distance of UAVs while simultaneously avoid obstacles, and maintain altitude constraints as well as the shape of the UAV formation. A multiple-objective optimisation algorithm, called the Angle-encoded Particle Swarm Optimization (θ-PSO) algorithm, is proposed to accelerate the swarm convergence with angular velocity and position being used for the location of particles. The whole formation is modelled as a virtual rigid body and controlled to maintain a desired geometric shape among the paths created while the centroid of the group follows a pre-determined trajectory. Based on the testbed of 3DR Solo drones equipped with a proprietary Mission Planner, and the Internet-of-Things (IoT) for multi-directional transmission and reception of data between the UAVs, extensive experiments have been conducted for triangular formation maintenance along a monorail bridge. The results obtained confirm the feasibility and effectiveness of the proposed approach.**

Keywords: **Quadcopter, θ-PSO, path planning, IoT, triangular formation, collision avoidance.**

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) have been found in many applications, from farm monitoring to civil infrastructure inspection, logistics to surveillance and rescue, from military to industrial applications with numerous studies available in the literature, see, e.g. [1]–[5]. However, the increasing demands in applications together with rapid development of technologies, especially computing, sensors, and communications have transcended the use of a single UAV to the formation and coordination of a group of them. Those UAVs in cooperation will be able to accomplish more challenging tasks like drag reduction, telecommunication relay and source and seeking in more efficient ways [6]–[8].

In multiple UAVs formation, path planning and shape preserving are essential for the success of tasks assigned as they provide references for lower level sensing and controlling as well as decide the usability of mission data collected. Over the last decade, there has been a number of studies on this topic with several algorithms introduced. The popular ones include A^*, D^*, Rapidly Exploring Random Tree (RRT) and Probabilistic Roadmap Method (PRM) [9]–[11]. The advantage of the A^* and D^* algorithms lie in the ability to judge or evaluate the best point-to-point path so they can provide a flyable trajectory for UAVs [9]. The RRT and PRM are based on probabilistic reasoning, and hence, robust but they require the discretization of the operating space which affects the smoothness of the planned trajectory [10], [11]. In [12], the approach based on the Voronoi diagram has been developed with a good obstacle avoidance capability. Similarly, path planning based on the visibility binary tree algorithm was introduced in [13]. In another attempt, the Fast Marching Square (FM2) algorithm [14]–[17] has been verified for obtaining excellent results on mobile vehicles, but kinematic constraints have not been considered to validate the feasibility of the trajectories. The Dubins path was introduced as a suitable method to solve this problem in 2D [18], [19] and 3D [20] environments. However, the resulting paths are not optimized. To this end, advances in machine learning have been exploited to provide better results for trajectory generation as in [21], [22]. Those approaches however require large training data as the input. The path planning problem therefore remains important, especially for coordination of UAV formation in executing a field task [23].

In this study, we aim to develop a path planning algorithm for multi-quadcopter formation conducting inspection tasks of built infrastructure. Our approach begins with a 3D model of the inspected surface and its surrounding environmental features extracted from a satellite map. Reference waypoints are then determined. A multi-objective particle swarm optimization algorithm using angle-encoded PSO (θ-PSO) is developed to generate a desired trajectory, for the centroid of the group, that is finally translated into an individual track for each UAV based on its defined position in the formation. Different from path planning for a single UAV [24], the advantages of the proposed approach rest with not only the simple implementation for the whole group and realistic execution of the planning algorithm, but also the generation of optimized paths for each UAV in the formation. Those paths should be both safely flyable and feasible in order to inspect the structure with the UAV dynamic constraints considered. The paths are also required to pass through pre-defined waypoints. Besides, some new constraints are introduced to improve the collision avoidance capability and task efficiency. Extensive simulation, comparison and experiments have been conducted for evaluation. The results illustrate the validity and effectiveness of the proposed path planning algorithm.

The paper is organized as follows. Section II briefly describes the model of the quadcopter formation. Section III presents the design of the path planning algorithm using θ-PSO. Section III-D describes the implementation of trajectory planning for UAVs. Simulation and experimental results are introduced in Section IV. The paper ends with a

The authors are with School of Electrical and Data Engineering, Faculty of Engineering and Information Technology (FEIT), University of Technology Sydney (UTS), 81 Broadway, Ultimo NSW 2007, Australia {VanTruong.Hoang, ManhDuong.Phung, TranHiep.Dinh, Quang.Ha}@uts.edu.au

conclusion and recommendation for future work.

II. Triangular Formation Model

Figure 1 shows the inertial and formation frames that represent a triangular UAV formation. All measurements are referred to the inertial frame O with axes x_O, y_O and z_O. Positions of UAV$_n$, $n = 1,2,3$, in the inertial frame are denoted as $P_n = \{x_n, y_n, z_n\}$. The formation frame, $\{x_F, y_F, z_F\}$, is defined such that the origin P_F is chosen to be coincident with the centroid of the triangle; the axis x_F is the direction from the centroid of the triangle to the UAV$_1$ position; the axis z_F is perpendicular to the plane containing three UAVs pointing downward; and the axis y_F is perpendicular to the plane formed by the x_F and z_F axes. This moving frame allows to determine its relative orientation with respect to the fixed inertial frame. The group position is represented by the centroid of the triangle and expressed as:

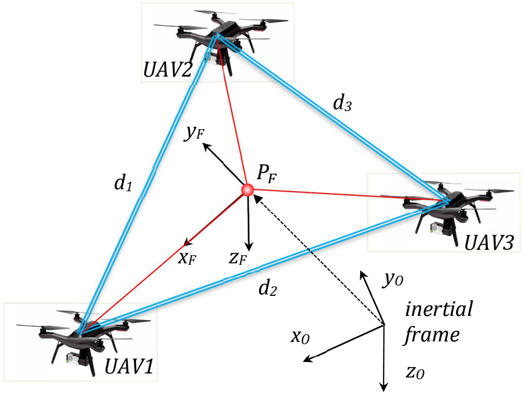


Fig. 1: Inertial and formation frames in UAV formation

$$P_F = \frac{1}{3}\sum_{i=1}^{3} P_n. \tag{1}$$

Let denote the distances between UAV$_1$, UAV$_2$ and UAV$_3$ to the centroid of the triangle as $d_{1,F}$, $d_{2,F}$, and $d_{3,F}$, respectively. The rotation matrix which represents the relation between the formation and inertial frames is determined as:

$$R_{OF} = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}, \tag{2}$$

where $s_x = sin(x)$, $c_x = cos(x)$, and ϕ, θ and ψ are Euler angles of the shape.

III. Formation Path Planning

When generating a path for the desired motion of multiple UAVs in a group, a number of constraints are required to be fulfilled for maintenance of the formation, maneuverability of every single UAV, operating space, and obstacle avoidance. In this work, all of the constraints will be incorporated into a multi-objective function. The path planning problem can be then simplified to the creation of a feasible path for the centroid of the UAV formation. Since our goal is to construct optimal paths for all UAVs in the group, it is essential to speed up the convergence of the optimization process for the whole formation. Therefore, we propose to use the angle-coded PSO (θ-PSO) described as follows.

A. Angle-encoded PSO or θ-PSO

The PSO is a population-based stochastic optimization algorithm inspired by social behavior of bird flocking [25], [26]. In PSO, a set of particles is generated, each seeks for the optimum solution by moving in a way that compromises between its own experience and the social experience. Initially, each particle is assigned a random position, x_i, and velocity, v_i. The particle motion is then updated by the following equations:

$$v_{ij}^{k+1} = w v_{ij}^k + c_1 r_{1i}^k (p_{ij}^k - x_{ij}^k) + c_2 r_{1i}^k (p_{gj}^k - x_{ij}^k) \tag{3}$$
$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1}, (i = 1,2,...,N; j = 1,2,...,S), \tag{4}$$

where N is the swarm size, S is dimension of the searching space, w is the inertial weight, r_1 and r_2 are two pseudorandom scalars, c_1 and c_2 are the gain coefficients, p_{ij} and p_{gj} are the local-best and global-best positions of the particle i, and subscript k is the iteration index. The values of p_{ij} and p_{gj} are evaluated based on a cost function to be defined in the next section.

For traditional path planning, the position of particles often represents the location of UAVs. This representation can give good results, but it also slows down the swarm convergence if the momentums of particles are not well adjusted [27]. To overcome this problem, we propose to use the angle-encoded PSO or θ-PSO, motivated by [26], in which the location of particles is encoded by the angle of UAVs. The θ-PSO is described as:

$$\begin{cases} \Delta\theta_{ij}^{k+1} = w\Delta\theta_{ij}^k + c_1 r_{1i}^k (\lambda_{ij}^k - \theta_{ij}^k) + c_2 r_{1i}^k (\lambda_{gj}^k - \theta_{ij}^k) \\ \theta_{ij}^{k+1} = \theta_{ij}^k + \Delta\theta_{ij}^{k+1}, (i = 1,2,...,N; j = 1,2,...,S) \\ x_{ij}^k = \dfrac{1}{2}\left[(x_{max} - x_{min}) sin\left(\theta_{ij}^k\right) + x_{max} + x_{min}\right], \end{cases} \tag{5}$$

where $\theta_{ij} \in [-\pi/2, \pi/2]$ and $\Delta\theta_{ij} \in [-\pi/2, \pi/2]$ are respectively the phase angle and phase angle increment of the ith particle in dimension j; $\Lambda_g = [\lambda_{g1}, \lambda_{g2}, ..., \lambda_{gS}]$ and $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, ..., \lambda_{iS}]$ are respectively the global and personal best positions; and x_{max} and x_{min} are the upper and lower restrictions of the search space.

For path planning of the centroid, each particle is associated to a specific path instance T_{Fi} and the phase angle-encoded population can be presented as:

$$\Theta = [\Theta_1, \Theta_2, ..., \Theta_N]^T \tag{6}$$

Suppose each path T_{Fi} consists of $v+2$ waypoints, including the start and target ones. As those start and target waypoints are predetermined, they can be excluded from the particle. Thus each particle has the dimension of $3v$ and can be represented by the following fixed-length phase angle-encoded vector:

$$\Theta_i = [\theta_{i1}, ..., \theta_{iv}, \theta_{i,v+1}, ..., \theta_{i,2v}, \theta_{i,2v+1}, ..., \theta_{i,3v}]. \tag{7}$$

Using the mapping $f : [-\pi/2, \pi/2] \rightarrow [x_{min}, x_{max}]$, we obtain the position of a particle as:

$$X_i = f(\Theta_i) = [x_{i1}, ..., x_{iv}, x_{i,v+1}, ..., x_{i,2v}, x_{i,2v+1}, ..., x_{i,3v}], \tag{8}$$

where $x_{ij} = f(\theta_{ij})$ is the jth dimension of the ith particle's position ($i = 1, ..., N$; $j = 1, ..., 3v$), $x_{i1}, ..., x_{iv}, x_{i,v+1}, ..., x_{i,2v}$ and $x_{i,2v+1}, ..., x_{i,3v}$ represent the x, y and z coordinates of the vth waypoint of path T_{Fi}, respectively.

B. Cost Function

The selection of a proper cost function for the PSO is essential to achieve the globally optimal solution for the searching process. The cost function for any trajectory often forms by two major evaluations, the length and violation cost of the path. The former helps to minimize the total travelling distance of the path whereas the latter is to avoid collisions of UAVs with each other and with obstacles. In a 3D site, other constraints should also be required, e.g., restrictions in flying altitude, heading angle and path curve. In our system, we use the quadcopters that have capabilities to carry out sharp and abrupt changes in angles and curves [28]. Thus, the constraints in angle and curve can be relaxed. The multi-objective constraint is now incorporated into the cost function in the following form:

$$J_F(T_{Fi}) = \sum_{n=1}^{3} \beta_n J_n(T_{Fi}), \tag{9}$$

where T_{Fi} is the formation path; β_n is the weighting factor indicating the corresponding threat intensity; and $J_n(T_{Fi})$, $n = 1, 2, 3$, are the costs associated with the path length, collision violation and flying altitude, respectively. To determine $J_n(T_{Fi})$, we split the formation path T_{Fi} into m segments. Each segment is represented by coordinates of its ending nodes $P_{Fi,l} = \{x_{Fi,l}, y_{Fi,l}, z_{Fi,l}\}, l = 0..m$. By denoting the length of the segment connecting nodes $P_{Fi,l}$ and $P_{Fi,l+1}$ as $\left\| \overrightarrow{P_{i,l}P_{i,l+1}} \right\|$, the cost J_1 corresponding the path length is then calculated for all segments:

$$J_1(T_{Fi}) = \sum_{l=0}^{m} \left\| \overrightarrow{P_{i,l}P_{i,l+1}} \right\| \tag{10}$$

$$= \sum_{l=0}^{m} \sqrt{dx_{Fi,l}^2 + dy_{Fi,l}^2 + dz_{Fi,l}^2}. \tag{11}$$

Let K be the set of all obstacles for a given UAV within its operation space. Assume that each obstacle is prescribed in a cylinder with the center's coordinate C_k and radius r_k, as shown in Fig. 2. The surfaces of cylinders then can be used to form constraints for obstacle avoidance. Specifically, the safe distance $d_{s,k}$ from the obstacle k is calculated from the cylinder center to its surface at the altitude $z_{M,l}$ as:

$$d_{s,k} = \begin{cases} \sqrt{r_k^2 + (z_{M,l} - z_k)^2} & \text{if } z_{M,l} \le z_{max,k} \\ \sqrt{r_k^2 + (z_{max,k} - z_k)^2} & \text{if } z_{M,l} > z_{max,k}, \end{cases} \tag{12}$$

where $z_{max,k}$ is the height of obstacle k. To compute the violation cost between each generated path and obstacle centres, we first assumed that the formation is rigid and can be fit within a sphere with the radius

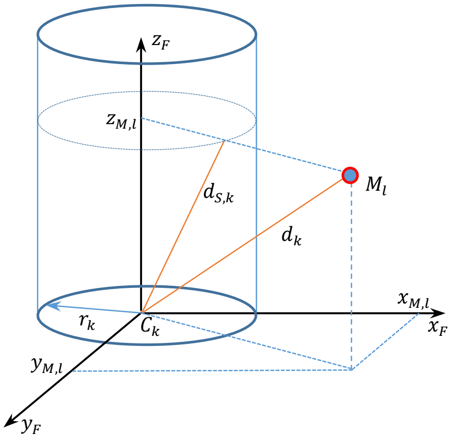


Fig. 2: Obstacle representation and safe distance calculation

$$r_{s,k} = r_Q + r_F + d_{s,k}, \tag{13}$$

where r_Q is the radius of quadcopters including propellers, r_F is the radius of the formation,

$$r_F = \max(D_{i,F}), \tag{14}$$

in which $D_{i,F}$ is the distance from quadcopter i to the formation centroid. The violation cost can be now derived as follows:

For the kth obstacle, compute the distance from its center C_k to the segment $\overrightarrow{P_{i,l}P_{i,l+1}}$:

$$d_{l,k} = \sqrt{(x_{M,l} - x_k)^2 + (y_{M,l} - y_k)^2 + (z_{M,l} - z_k)^2}, \tag{15}$$

where $M_l = \{x_{M,l}, y_{M,l}, z_{M,l}\}$ is the midpoint of the segment as shown in Fig. 2. At a given altitude $z_{M,l}$, $d_{l,k}$ is then compared with the safe distance to the obstacle. The comparison results in the following violation function:

$$V_{l,k}(T_{Fi}) = \sum_{k=1}^{K} \max(1 - \frac{d_{l,k}}{r_{s,k}}, 0). \tag{16}$$

This function ensures that the distance $d_{l,k}$ must be larger than the safe distance for obstacle avoidance. The violation cost is then computed for all obstacles as:

$$V_l(T_{Fi}) = \frac{1}{K} \sum_{k=1}^{K} V_{l,k}(T_{Fi}). \tag{17}$$

For all m segments, the final violation cost on average (with respect to the centroid) is represented as:

$$J_2(T_{Fi}) = \frac{1}{m} \sum_{l=0}^{m} V_l(T_{Fi}), \tag{18}$$

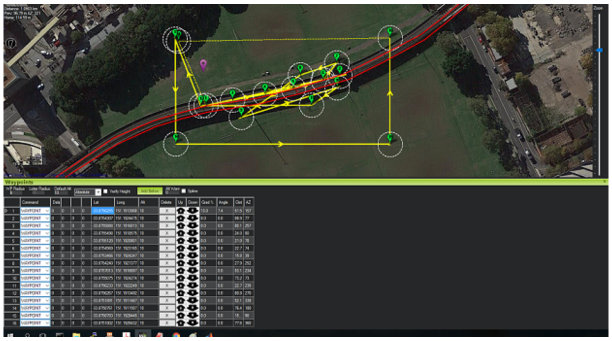
In terms of flying altitude, UAVs are often required to follow the terrain at a certain height to avoid crashing. Thus, the altitude of each UAV must be within a predefined interval between two given extrema, the minimum and maximum safe clearances $z_{\min}$ and $z_{\max}$. Thus, the corresponding cost component can be expressed as

$$\begin{cases} J_3(T_{Fi}) = \sum_{l=0}^{m} dz_{Fi,l} \\ dz_{Fi,l} = \begin{cases} z_{Fi,l} - z_{\max}, & \text{if } z_{Fi,l} > z_{\max} \\ 0, & \text{if } z_{\min} \leq z_{Fi,l} \leq z_{\max} \\ z_{\min} - z_{Fi,l}, & \text{if } 0 < z_{Fi,l} < z_{\min} \\ \infty, & \text{if } z_{Fi,l} \leq 0. \end{cases} \end{cases} \tag{19}$$

This last condition in $dz_{Fi,l}$ is critical for safe operations as negative values of the altitude could be generated causing UAVs to crash to the ground.

C. Path planning implementation

The implementation starts with choosing the operation space of UAVs and the infrastructure to be inspected. This can be done by using a navigation map with satellite images. For example, here a monorail bridge as a testbed subject to inspection can be loaded on the Mission Planner, as shown in Figure 3. The obstacles are also identified based on this map. Furthermore, range sensors such as lidars can be used to form a 3D map of the environment as in our previous work [24]. Based on those inputs, the cost function together with constraints can be defined as described in the previous section. The θ-PSO algorithm will then be run to obtain the desired path. The steps can be summarized as follows:



	Command	Dela				Lat	Long	Alt	Delete	Up	Down	Grad %	Angle	Dist	AZ
1	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			10.0	7.4	61.5	157
2	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	99.9	77
3	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	60.1	257
4	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	24.3	60
5	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	21.0	78
6	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	22.7	74
7	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	15.8	39
8	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	27.9	252
9	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	50.1	234
10	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	70.2	73
11	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	22.7	236
12	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	80.9	270
13	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	52.1	339
14	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	76.4	180
15	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	15...	90
16	WAYPOINT	0	0	0	0	[illegible]	[illegible]	10	X			0.0	0.0	77.6	360

Fig. 3: Mission Planner incorporating Google Satellite Map to create initial information and an inspection plan

(1) Compute all parameters for UAVs and formation model, based on the desired task.
(2) Select appropriate parameters for PSO such as the population size, the number of iterations, gain coefficients and so on.
(3) Identify environmental information of the flying field and initialise parameters of obstacles.
(4) Initialize randomly the path of each particle from the start point to the target point.
(5) Evaluate each path based on the cost function (9).
(6) Compute each particle's personal best and the global best positions by running PSO repeatedly.
(7) The desired path is chosen as the maximum number of iterations is reached.

D. Path generation for individual UAV

Given the optimal path, T_F^*, generated by the θ-PSO for the formation centroid, it is necessary to produce a specific path for each UAV so that the shape of the formation during the flight can be maintained. Those paths can be computed based on the PSO's generated path and the desired relative distances among the UAVs. Let $P_{n,d} = [x_{n,d}, y_{n,d}, z_{n,d}]^T$ be the reference position for each UAV during the flight and $P_n = [x_n, y_n, z_n]^T$ be the actual position, P_n can be obtained from GPS data of the nth UAV. We then define the relative position errors of the nth UAV during the flight in the inertial frame as:

$$\begin{bmatrix} e_{nx} \\ e_{ny} \\ e_{nz} \end{bmatrix} = \begin{bmatrix} x_{n,d} - x_n \\ y_{n,d} - y_n \\ z_{n,d} - z_n \end{bmatrix} \tag{20}$$

Using the rotation matrix (2), with $R_{FO}(t) = R_{OF}^{-1}(t)$, the errors in (20) can be converted into the errors in the formation frame as:

$$\begin{bmatrix} e_{nxF} \\ e_{nyF} \\ e_{nzF} \end{bmatrix} = R_{FO}(t) \begin{bmatrix} e_{nx} \\ e_{ny} \\ e_{nz} \end{bmatrix} \tag{21}$$

The customized path for each UAV can then be represented in terms of trajectory control command as:

$$T_n = T_F^* + \Delta T_n, \tag{22}$$

where T_F^* is the trajectory of the centroid, computed by θ-PSO, and ΔT_n is the amount added to direct the UAV away from the centroid. ΔT_n is calculated from the desired relative distances among the UAVs and the relative position errors in (21). The output T_n will be fed to the internal controller of UAVs for trajectory tracking.

IV. EXPERIMENTS

A series of experiments have been conducted to evaluate the feasibility and efficiency of the proposed algorithm.

A. Experimental setup

The task assigned in experiments is to inspect simultaneously different surfaces of a bridge using three UAVs. As mentioned, the Mission Planner, a proprietary ground control station software, incorporating the Google Satellite Map (GST) is used to collect initial information about the structure and its surrounding environment. Here, the operation space is chosen with dimensions $\{141m \times 101m \times 40m\}$, equivalent to the GST coordinates $\{-33.87601, 151.191182, 0\}$ and $\{-33.875086, 151.192676, 40\}$, as illustrated in Fig. 3. The starting and target points are set as $P_{i,0} = \{40.0, 8.0, 30\}$ and $P_{i,l+1} = \{64, 108, 34\}$, respectively. Therein, ten obstacles are identified, each with a different radius. The formation platform chosen in this work is three identical 3DR Solo drones as shown in Fig. 4, whereby the controllers at the low level for each UAV have been reported in [28]. The communications among them were conducted by adding an additional Internet-of-Things (IoT) board to each drone and a base Wi-Fi router. The IoT boards together with the ground

Fig. 4: 3DR Solo drones used in experiments

Wi-Fi station form a network that can connect to the internet to transmit the data to other processing station. Also through this network, the drones can exchange their position, velocity and status data during flight. By employing that information, the onboard computer calculates the inverse kinematics (the formation variables based on the positions of the robots), compares it with their neighbours and the formation centroid to obtain the position errors. Those errors are then eliminated by the tracking control action generated. In our path planning algorithm, the number of particles, waypoints, and iterations are respectively selected as 100, 10, and 300. Parameters of the three quadcopters with respect to the centroid of the formation are $\Delta T_1 = [0, 0, 2]$ m, $\Delta T_2 = [3, 0, -1]$ m and $\Delta T_3 = [-3, 0, -1]$ m. The minimum and maximum clearances between UAVs and the terrain are set to $z_{\max} = 32$ m and $z_{\min} = 28$ m, respectively.

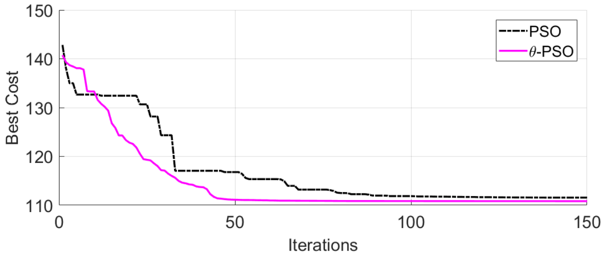


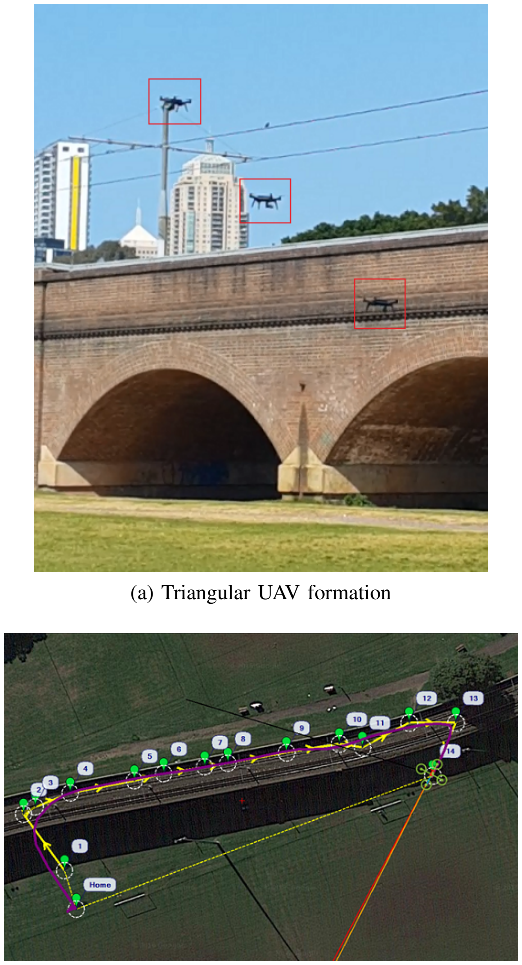
Fig. 5: Convergence comparison between conventional PSO and θ-PSO

B. Results

The path planning results are presented in this subsection, whereby it is expected that the designed method can generate collision-free paths for the three UAVs with sufficiently fast convergence using the proposed algorithm. For this, let us first compare the performance of the proposed θ-PSO with a conventional PSO algorithm. Figure 5 shows the cost values over iterations. It can be seen that although both algorithms are convergent, the θ-PSO introduces a faster and more stable conversion. The results are confirmed as recorded in Table I which shows the average cost value and convergence iterations.

TABLE I: PSO and θ-PSO performance comparison

Algorithm	Min cost	Max cost	Iterations
PSO	112.43	143.0	102
θ-PSO	111.02	142.84	68



(a) Triangular UAV formation

(b) Planned path (yellow) and flown path (violet)

Fig. 6: Bridge inspection with UAV formation

Given the generated path, we have conducted field-test experiments in which the triangular formation automatically navigated along the inspected surface, as depicted in Fig. 6. The 3D trajectories generated from that of the formation centroid path are shown in Fig. 7, where it can be seen that the three drones can take off, reach to their individual altitude set-point, descend and finally arrive their target position at almost the same interval, while maintaining the desired triangular shape. This result can be further verified via the altitude time responses of the three UAVs as recorded in Fig. 8. It is clear that the quadcopters are capable of avoiding obstacles and preserving the desired formation configuration during the inspection task. For further evaluation, Fig. 9 shows the error between the planned and flown paths computed by selecting the closest coordinates of the flight trajectories to the reference points. Those small errors, mainly caused by the positioning system of UAVs from the GPS signal received, imply feasibility and reasonable smoothness of the generated path for the deployment of the drone formation.

V. CONCLUSION

This paper has presented a novel approach for the path planning problem of multiple UAVs navigating in a desired shape for infrastructure inspection tasks. Here the angle-coded PSO is proposed to find feasible and obstacle-free paths for the whole formation by minimizing a cost function that incorporates multiple constraints for shortest paths and

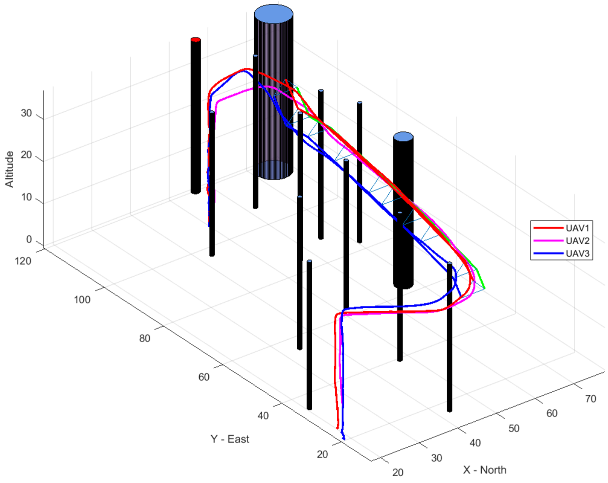


Fig. 7: Trajectories of three drones tracking the planned paths

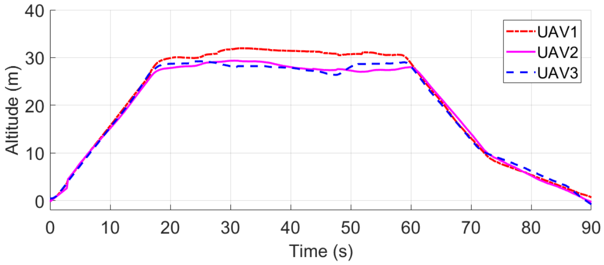


Fig. 8: Altitudes of the three drones in the formation test

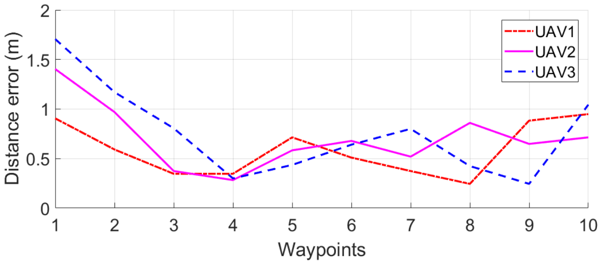


Fig. 9: Errors between the planned and flown paths

safe operation of the drones. From the centroid, customized paths are generated for individual UAV to maintain the formation using a proprietary software while inter-UAV communication is achieved via the IoT boards. Implementation on a triangular formation is reported along with field tests on a monorail bridge. The results confirmed the validity and feasibility of the proposes approach for UAV formation inspection of built infrastructure.

References

[1] B. D. Anderson, B. Fidan, C. Yu, and D. van der Walle, "Uav formation control: Theory and application," in *Recent Advances in Learning and Control*. Springer, 2008, pp. 15–33.

[2] L. Derafa, A. Benallegue, and L. Fridman, "Super twisting control algorithm for the attitude tracking of a four rotors uav," *Journal of the Franklin Institute*, vol. 349, no. 2, pp. 685–699, 2012.

[3] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2624 – 2661, 2016.

[4] S. Rajappa, C. Masone, H. Bulthoff, and P. Stegagno, "Adaptive super twisting controller for a quadrotor uav," in *Robotics and Automation (ICRA), 2016 IEEE Int. Conf.* IEEE, 2016, pp. 2971–2977.

[5] N. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, 2016.

[6] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in uav communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016.

[7] A. D. Nguyen, Q. P. Ha, S. Huang, and H. M. Trinh, "Observer-based decentralized approach to robotic formation control," in *Australasian Conference on Robotics and Automation*, 2004, pp. 1–8.

[8] F. Liao, R. Teo, J. L. Wang, X. Dong, F. Lin, and K. Peng, "Distributed formation and reconfiguration control of vtol uavs," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 270–277, 2017.

[9] P. De Monte and B. Lohmann, "Position trajectory tracking of a quadrotor helicopter based on l1 adaptive control," in *Control Conference (ECC), 2013 European*. IEEE, 2013, pp. 3346–3353.

[10] F. Yan, Y.-S. Liu, and J.-Z. Xiao, "Path planning in complex 3d environments using a probabilistic roadmap method," *Int. J. Automation and Computing*, vol. 10, no. 6, pp. 525–533, 2013.

[11] M. Kothari and I. Postlethwaite, "A probabilistically robust path planning algorithm for uavs using rapidly-exploring random trees," *Journal of Intelligent & Robotic Systems*, pp. 1–23, 2013.

[12] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a voronoi diagram for path planning of autonomous uav," *Aerospace Science and Technology*, vol. 16, no. 1, pp. 47–55, 2012.

[13] A. T. Rashid, A. A. Ali, M. Frasca, and L. Fortuna, "Path planning with obstacle avoidance based on visibility binary tree algorithm," *Robot.& Autonom. Systems*, vol. 61, no. 12, pp. 1440–1449, 2013.

[14] S. Garrido, M. Malfaz, and D. Blanco, "Application of the fast marching method for outdoor motion planning in robotics," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 106–114, 2013.

[15] C. Arismendi, D. Álvarez, S. Garrido, and L. Moreno, "Nonholonomic motion planning using the fast marching square method," *International Journal of Advanced Robotic Systems*, vol. 12, no. 5, p. 56, 2015.

[16] D. Álvarez, J. V. Gómez, S. Garrido, and L. Moreno, "3d robot formations path planning with fast marching square," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 3-4, p. 507, 2015.

[17] V. González, C. Monje, L. Moreno, and C. Balaguer, "Uavs mission planning with imposition of flight level through fast marching square," *Cybernetics and Systems*, vol. 48, no. 2, pp. 102–113, 2017.

[18] P. Sujit, J. George, and R. Beard, "Multiple uav coalition formation," in *American Control Conference, 2008*. IEEE, 2008, pp. 2010–2015.

[19] H. Yu and R. Beard, "A vision-based collision avoidance technique for micro air vehicles using local-level frame mapping and path planning," *Autonomous Robots*, vol. 34, no. 1-2, pp. 93–109, 2013.

[20] H. Chitsaz and S. M. LaValle, "Time-optimal paths for a dubins airplane," in *Decision and Control, 2007 46th IEEE Conference on*. IEEE, 2007, pp. 2379–2384.

[21] H. Duan, P. Li, Y. Shi, X. Zhang, and C. Sun, "Interactive learning environment for bio-inspired optimization algorithms for uav path planning," *IEEE Trans. Educat.*, vol. 58, no. 4, pp. 276–281, 2015.

[22] B. Zhang, Z. Mao, W. Liu, and J. Liu, "Geometric reinforcement learning for path planning of uavs," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 2, pp. 391–409, 2015.

[23] Y. Chen, J. Yu, X. Su, and G. Luo, "Path planning for multi-uav formation," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 1, pp. 229–246, 2015.

[24] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for uav vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, 2017.

[25] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *International conference on evolutionary programming*. Springer, 1998, pp. 591–600.

[26] Y. Fu, M. Ding, and C. Zhou, "Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for uav," *IEEE Trans. Sys., Man, and Cybernetics-Part A*, vol. 42, no. 2, pp. 511–526, 2012.

[27] N. M. Kwok, Q. P. Ha, D. Liu, G. Fang, and K. C. Tan, "Efficient particle swarm optimization: a termination condition based on the decision-making approach," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 3353–3360.

[28] V. T. Hoang, M. D. Phung, and Q. P. Ha, "Adaptive twisting sliding mode control for quadrotor unmanned aerial vehicles," in *Control Conference (ASCC), 2017 11th Asian*. IEEE, 2017, pp. 671–676.