

E²Coop: Energy Efficient and Cooperative Obstacle Detection and Avoidance for UAV Swarms

Shuangyao Huang, Haibo Zhang, Zhiyi Huang

Department of Computer Science

University of Otago

133 Union Street East, Dunedin 9016, New Zealand

{shuangyao, haibo, hzy}@cs.otago.ac.nz

Abstract

Energy efficiency is of critical importance to trajectory planning for UAV swarms in obstacle avoidance. In this paper, we present *E²Coop*, a new scheme designed to avoid collisions for UAV swarms by tightly coupling Artificial Potential Field (APF) with Particle Swarm Planning (PSO) based trajectory planning. In *E²Coop*, swarm members perform trajectory planning cooperatively to avoid collisions in an energy-efficient manner. *E²Coop* exploits the advantages of the active contour model in image processing for trajectory planning. Each swarm member plans its trajectories on the contours of the environment field to save energy and avoid collisions to obstacles. Swarm members that fall within the safeguard distance of each other plan their trajectories on different contours to avoid collisions with each other. Simulation results demonstrate that *E²Coop* can save energy up to 51% compared with two state-of-the-art schemes.

Introduction

The society today is showing trends towards large-scale, infrastructure-less, and contactless connections. UAV swarms have become a promising solution to provide such connections in terms of safety and cost. A UAV swarm is a combination of several UAVs collaborating with each other to produce enhanced capabilities by providing resilience and increased variety. Swarm members accomplish a common mission and perform Obstacle Detection and Avoidance (ODA) cooperatively as one group. However, the biggest challenge that prevents large-scale and long-haul UAV applications is energy efficiency. The need for recharging UAVs requires densely distributed charging stations which are temporarily unavailable in most cities. Hence, planning trajectories for UAV swarms that can avoid collisions with minimum energy consumption remains a critical and open problem.

ODA is a long-standing problem in robotics. With the ever increasing number of applications of UAVs in recent years, extensive studies have been conducted on ODA for UAVs, including Velocity Obstacles (Liu et al. 2018; Snape et al. 2011), Artificial Potential Fields (Huang and Low 2018; Tsang et al. 2018), and hybrid methods combining more than one approach (Pradhan et al. 2019; Xia et al. 2020). However, existing solutions on ODA for UAV swarms either

fail to address energy efficiency or don't plan trajectories for members of UAV swarms to avoid collisions among UAVs within the swarm.

In this paper, we present *E²Coop*, a new scheme that tightly incorporates Artificial Potential Fields (APF) and Particle Swarm Optimization (PSO), and borrows the Active Contour Model from image processing to solve the problem of collision avoidance and energy efficiency for UAV swarms in trajectory planning. Even though the idea of combining APF and PSO is not new, our scheme differs from existing solutions from the following perspectives: (1) we propose a new concept called the environmental field, which can be generated at each swarm member by superposing the APFs of the swarm and the detected obstacles. With this global knowledge of the environment, the swarm members perform obstacle avoidance cooperatively in a safe and energy-efficient manner by tightly coupling APF and PSO; (2) in *E²Coop*, the relation between APF and PSO can be summarized as: APF defines the search space of PSO and PSO finds parameters for the optimal trajectories on APF. The PSO search spaces for individual UAVs are orthogonal to each other on APF, which provides coordination within the swarm.

In *E²Coop*, each swarm member first constructs the environmental field, and then uses PSO to plan the optimal trajectory for its next step by minimizing the designed fitness function. During ODA, if a member detects another member within its safeguard distance, the swarm members work cooperatively to reschedule their trajectories to avoid UAV-to-UAV collisions. The key contributions of this paper are summarized as follows:

- We propose a new fitness function for PSO-based trajectory planning that borrows the idea of contour extraction in image processing. The new fitness function takes into account both energy efficiency and safety requirement for obstacle avoidance.
- We propose a layered structure that tightly couples APF and PSO to optimize energy efficiency and enforce swarm safety. A compact expression of UAV trajectories with only two parameters is proposed, which reduces the dimensionality of the PSO search space and hence reduce the complexity.
- We evaluate our scheme through extensive simulations

and compare it with two state-of-the-art ODA schemes. Simulation results show that E^2Coop can save energy up to 51% compared with the two state-of-the-art schemes.

The rest of this paper is organized as follows: we first introduce the state-of-the-art work on UAV collision detection and avoidance. We then give a general introduction of E^2Coop . After that, we present the design of the fitness function for PSO-based trajectory planning. Trajectory adjustment is then introduced to avoid UAV-to-UAV collisions. Finally, the trajectory planning approach and simulation results are presented, followed by conclusion.

Related Work

A reciprocal collision avoidance scheme is developed to avoid collisions among robots collaboratively using Velocity Obstacles (VO) (Snape et al. 2011, 2009). In VO, each robot avoids collisions with others by selecting a collision-free velocity from a velocity pool excluded by other robots. Therefore, the robots keep changing velocities, causing oscillations of individual robots which is not energy efficient.

Another popular method for ODA is Artificial Potential Field (APF). For example, a method combining an attractive field and a repulsive field is developed to generate safe and smooth trajectories for UAVs in urban environments (Huang and Low 2018). Though this method is more energy efficient than VO, it didn't address the coordination of a swarm of robots in ODA, which may result in collisions between robots themselves in a swarm. APF is combined with a sampling based method, with a robot heading term introduced as a virtual force for swarm planning (Waller and Plaku 2013). However, it failed to consider energy efficiency. Vehicle-to-vehicle repulsive forces are introduced to resolve potential collisions between robots of a swarm in APF methods (Liang et al. 2019). However, such method results in zigzags and irregular trajectories of the robots due to the lack of coordination between robots in the swarm. Therefore, this method has the same issue as VO regarding energy efficiency.

Another method for ODA worth mentioning is Particle Swarm Optimization (PSO). For example, PSO is used to find intermediate points between source and destination by which smooth trajectories are interpolated (Tharwat et al. 2019). Another method, PSO is used to search for parameters to interpolate splines as smooth trajectories (Song et al. 2019). However, they didn't solve the problem of path planning for swarms. Some methods based on PSO incorporate APF in ODA for swarms, in order to provide environmental information regarding obstacles and the destination for the fitness function in PSO. For example, a force field PSO is proposed, called FFPSO, to consider the potential field in the fitness function of PSO (Parker, Butterworth, and Luo 2019). A potential field based PSO is proposed, called PPSO, which adds a new smoothing field in order to ensure smoothness of the trajectories (Cai and Yang 2014). These methods are similar to our E^2Coop in the sense that 1) APF and PSO are closely coupled, and 2) smoothness of trajectories are considered. However, they only consider discrete way points on a trajectory which are independent from each

other. In E^2Coop , we consider a whole trajectory in our PSO solution, which results in much smoother trajectories. We will compare the performance of E^2Coop with FFPSO and PPSO to show the advantages of E^2Coop through simulations.

Overview of E^2Coop

In this section, we give an overview of our proposed scheme E^2Coop for ODA in a UAV swarm. E^2Coop is triggered when an obstacle is detected within a threshold distance by the sensors of any UAV in a swarm. During a mission, each UAV in the swarm periodically broadcasts its own location and velocity to other swarm members. Each UAV also keeps sensing the environment and shares the location and velocity of the obstacles it has detected to other members in the swarm. Hence, each UAV in the swarm maintains the same global view of the environment including all swarm members and the detected obstacles. As this work focuses on designing trajectory planning algorithms, we assume the communication among swarm members is well maintained and guaranteed at all time. Hence, the synchronization among swarm members is maintained as well. Previous work shows that the energy consumed in wireless communication is a thousand times smaller than flight energy (Shan et al. 2020). Hence, we only consider the flight energy in trajectory planning. The **key idea** of E^2Coop is to use this global view of the environment for all members in the swarm to perform obstacle avoidance cooperatively in a safe and energy-efficient manner.

Fig. 1 illustrates how APF is combined with PSO in E^2Coop for ODA. The system consists of three layers. The top layer is the *APF* layer designed for environmental awareness. Once a UAV detects an obstacle, it constructs an APF called the environment field based on the locations and velocities it estimates of the obstacles or receives from other swarm members (details on environment field construction is presented in the section on Fitness Function). The bottom layer is called the *Trajectory Planning* layer that uses PSO to find the most energy-efficient trajectory to bypass the obstacle without any collision. We will show the safe and energy-efficient paths planned in this layer are contours in the en-

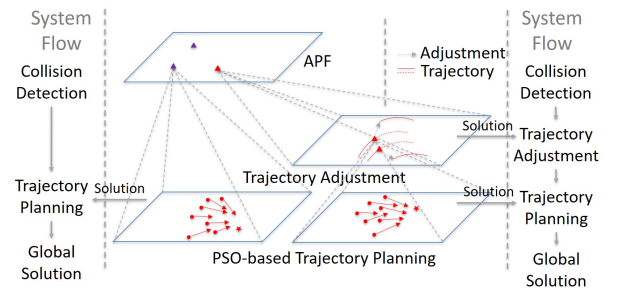


Figure 1: Illustration of combining APF and PSO in E^2Coop . UAVs that are close to each other and need adjustment are marked with red triangles. Other UAVs are marked with blue triangles.

vironment field. The middle layer is called the *Trajectory Adjustment* layer designed for avoiding collisions within the swarm. When the distance between any two swarm members is less than the safeguard distance, this layer is used to adjust the trajectory of these two UAVs to avoid collision. The main idea is to force the two UAVs to fly on different contours of the environment field so that the distance between them is no less than the safeguard distance. When no UAVs are close to each other, it enters *Trajectory Planning* directly; otherwise it first enters into the middle layer to adjust the contours for the UAVs, and then uses *Trajectory Planning* to find the energy-efficient and safe trajectory for each UAV to move to the new adjusted contour.

In E^2Coop , a path is composed of many small trajectories generated by trajectory planning. Due to the dynamics of the urban airspace and the limitations on sensing ability and onboard memory, we assume a UAV can only plan the trajectory for its next step based on its current knowledge and previous memory. For each step, UAVs find their trajectories by minimizing the following fitness function in the *Trajectory Planning* layer using PSO:

$$f(C_t) = \lambda_1 f_e(C_t) + \lambda_2 f_s(C_t), \quad (1)$$

where t is the current time and C_t is a curve combining the previous trajectory before t with the next trajectory to be planned. $f_e(C_t)$ is the cost function to ensure energy efficiency whereas $f_s(C_t)$ is the cost function to ensure safety along the trajectory. These functions will be defined in detail shortly. λ_1 and λ_2 are coefficients where $\lambda_1 + \lambda_2 = 1$.

Fitness Function

In this section, we define each term of the fitness function in Eq. (1) and show how the trajectories are planned with minimum energy consumption and no collisions with obstacles.

Energy Efficiency

The first item $f_e(C_t)$ in Eq. (1) is modelled as:

$$f_e(C_t) = e_v \int_{C_t} |C_t''(p)| dp. \quad (2)$$

where $C_t''(p)$ is the second-order derivative of the curve and e_v is an energy constant that will be defined shortly. In the following, we show that the energy consumption of a UAV traveling along C_t is minimized when Eq. (2) is minimized.

In this paper, we consider quadcopters, as they are widely used and have the ability of vertical take-off and landing. For a clear illustration of energy consumption of a UAV flying along a trajectory, we start with modelling the forces $\mathbf{T}$ and power P generated by the propellers. The thrust forces $\{T_j, j = 1, 2, 3, 4\}$ generated by the propellers and the orientation of the UAV, i.e., the Euler angles $[\alpha, \beta, \gamma]$ (pitch, roll, yaw), can be represented with the North-East-Down world coordinates and the UAV's Forward-Right-Down self-coordinates.

The total thrust $\mathbf{T}$ generated by the four propellers is always perpendicular to the UAV body plane. For theoretical analysis on trajectories, the altitude drop and horizontal shift

of UAVs caused by rolling motions are neglected. For UAVs turning at constant speed, it can be modelled as follow.

$$\mathbf{T} = \sum_{i=1}^4 \mathbf{T}_i = F_{drag} \mathbf{e}_f - mg \mathbf{e}_d + F_n \mathbf{e}_r, \quad (3)$$

where F_{drag} is air drag force, m is the total mass of the UAV including battery, load and sensors, g is the gravity, and F_n is the centripetal force required to perform turnings. $\mathbf{e}_f$, $\mathbf{e}_r$ and $\mathbf{e}_d$ are the unit vectors along the axes of the UAV's self-coordinates inside the world coordinates.

The power required to generate the total thrust $\mathbf{T}$ is modelled as (Stolaroff et al. 2018):

$$P_{min} = \|\mathbf{T}\|(v \sin \alpha + v_i), \quad (4)$$

where v is the ground speed, α is the pitch angle, and v_i is the induced velocity by the propellers.

Let η_s and η_e denote the start and end positions of a curve representing a UAV trajectory, the energy consumed by a UAV flying along this trajectory can be derived as follows:

$$\begin{aligned} E &= \int_{\eta_s}^{\eta_e} \|F_{drag}(p) \mathbf{e}_f - mg \mathbf{e}_d + F_n(p) \mathbf{e}_r\| (v \sin \alpha + v_i) dp \\ &= \int_{\eta_s}^{\eta_e} F_{drag}(p) \sin \alpha \cos \beta (v \sin \alpha + v_i) dp \\ &\quad + \int_{\eta_s}^{\eta_e} mg \cos \alpha \cos \beta (v \sin \alpha + v_i) dp \\ &\quad + \int_{\eta_s}^{\eta_e} F_n(p) \sin \beta (v \sin \alpha + v_i) dp, \end{aligned} \quad (5)$$

where F_{drag} can be estimated as below (Stolaroff et al. 2018):

$$F_{drag} = \frac{1}{2} \rho v_a^2 C_D A, \quad (6)$$

where ρ is the air density, v_a is the relative air velocity, C_D and A are the drag coefficient and projected area perpendicular to v , respectively. C_D is determined empirically based on the geometry, weight and pitch angle of the UAV.

When the trajectory is small, like the one in a planning step, it is reasonable to assume that the pitch angle α , the UAV velocity v , and the induced velocity v_i do not change when the UAV is flying along the trajectory. According to Eq. (6), F_{drag} can be seen as a constant within one planning step as C_D and A depend on the rotation angle. Since mg is also a constant, Eq. (5) can be rewritten as

$$\begin{aligned} E &= E_n + E_{const} \\ E_n &= \int_{\eta_s}^{\eta_e} F_n(p) (v \sin \alpha + v_i) \cdot \sin \beta dp \\ E_{const} &= (F_{drag} \sin \alpha \cos \beta + mg \cos \alpha \cos \beta) \\ &\quad \cdot (v \sin \alpha + v_i) \cdot (\eta_e - \eta_s), \end{aligned} \quad (7)$$

where E_n is the energy spent on generating centripetal force F_n to perform turnings and E_{const} is a constant within one step. Hence, minimizing E is equivalent to minimizing E_n .

The above analysis is consistent with our field experiment. In the experiments, a hexacopter with a diagonal

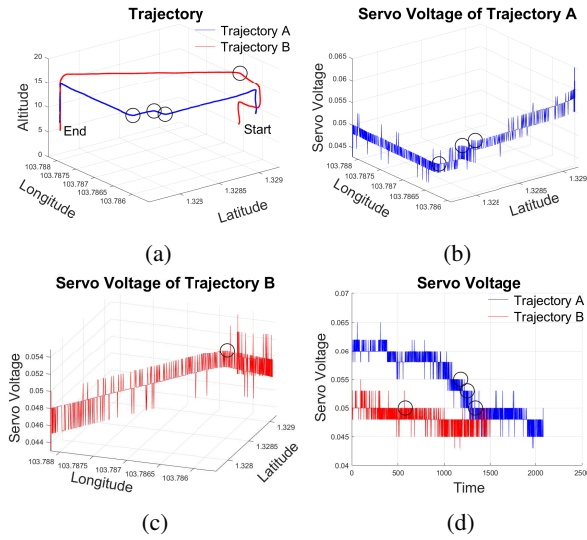


Figure 2: Field experiment results.

wheelbase of 1000 mm and controlled by Pixhawk 2.0 is set to fly at a constant speed along two trajectories respectively, as shown in Fig. 2a. Trajectory A and B are generated using a grid search based genetic evolutionary algorithm (Tan et al. 2019) and a potential field based algorithm (Huang and Low 2018), respectively. Trajectory A has three turnings that are approximately in right angles. On the other hand, Trajectory B has just one smooth turning. The servo voltage along the two trajectories are shown in Fig. 2b, 2c and 2d. Turnings and corresponding voltage variations are marked with circles. As shown in Fig. 2b and 2c, sharp turnings on Trajectory A cause more radical drops in servo voltage than the smooth turning on Trajectory B. From Fig. 2d, the energy consumed by Trajectory B is much smaller than that of Trajectory A. The full video is available at (ATMRI 2020).

Let $r(p)$ be the turning radius of the UAV, we have $F_n(p) = mv^2/r(p)$. Hence,

$$\begin{aligned} E_n &= mv^2(v \sin \alpha + v_i) \sin \beta \int_{\eta_s}^{\eta_e} \frac{1}{r(p)} dp \\ &= mv^2(v \sin \alpha + v_i) \sin \beta \int_{\eta_s}^{\eta_e} |\kappa(p)| dp \quad (8) \\ &= e_v \int_{\eta_s}^{\eta_e} |S''(p)| dp, \end{aligned}$$

where $S(p) = [x(p), y(p)]$ denotes the segment of trajectory considered in the current planning step. $\kappa(p)$ is the curvature of trajectory $S(p)$ and $S''(p)$ is the second order derivative of $S(p)$. $e_v = mv^2(v \sin \alpha + v_i) \sin \beta$ is a velocity dependent coefficient, and it is constant in a small step.

It can be seen that Eq. (8) is exactly same as Eq. (2). Therefore, the energy efficiency of a trajectory is ensured by minimizing Eq. (2) in our fitness function.

Safety

To avoid collisions with obstacles, each UAV first constructs a repulsive potential field for each obstacle and the swarm.

Then, the overall APF model, called the environment field, is constructed by superposing of the repulsive potential fields for all obstacles and the swarm. The reason to include the potential field for the swarm into the environmental field is twofold: 1) it stretches the environment field along the velocity direction of the swarm to improve smoothness of the UAV trajectories, and 2) it makes the environment field irregular around the swarm so that less UAVs are located at positions with similar intensity levels, thereby reducing the risk of UAV-to-UAV collisions.

To construct the potential field for the UAV swarm, a virtual leader is needed. The virtual leader is an imaginary UAV without the need to avoid collisions and is defined as follows:

$$\begin{aligned} p^* &= \bar{p} + \vec{p}d \cdot S_{adv}, \\ \bar{p} &= \frac{1}{N} \sum_{i=1}^N p_i, \end{aligned} \quad (9)$$

where p^* is the position of the virtual leader, p_i is the position of the i^{th} UAV in the swarm, $\bar{p}$ is the geometrical center of the swarm, and N is the number of UAVs in the swarm. $\vec{p}d$ is a unit vector pointing from the geometrical center $\bar{p}$ to the destination d , and S_{adv} is a distance offset that allows the virtual leader to be in the front of the swarm, so that it can provide navigating information in advance to UAVs. A proper setting for S_{adv} can be $\max_{i=1, \dots, N} \{|\bar{p}p_i|\}$, where $|\bar{p}p_i|$ is the distance between i^{th} UAV and the swarm geometrical center.

The velocity of the swarm, denoted by v_s , is represented by the velocity of the virtual leader. It is defined as follows:

$$v_s = \frac{1}{N} \sum_{i=1}^N v_i^u, \quad (10)$$

where v_i^u is the velocity of the i^{th} UAV, and v_s has the same direction as the unit vector $\vec{p}d$.

Based on Eqs. (9) and (10), the potential field of the swarm, denoted by $\Phi_s(q)$, is constructed as follows:

$$\Phi_s(q) = \begin{cases} \frac{v_s}{|qp^*|^2}, & |qp^*| \leq R^s \\ 0, & |qp^*| > R^s \end{cases} \quad (11)$$

where $\Phi_s(q)$ is the field intensity at position q , $|qp^*|$ is the distance from q to the swarm's virtual leader, R^s is the influential range of $\Phi_s(q)$ and $R^s \geq \max_{i=1, \dots, N} \{|p_i p^*|\}$, where $|p_i p^*|$ is the distance between i^{th} UAV and the virtual leader.

It is worth noting the swarm's potential field is symmetric around its virtual leader rather than its geometrical center. Swarm members closer to the virtual leader will have higher field intensities. In other words, the members closer to the virtual leader are in the front of the swarm, and thus are in greater danger than other members. This is because the selection of the virtual leader reflects the direction of the swarm's velocity. The virtual leader is imaginary for the sake of modeling and does not need to avoid any collision.

We assume large obstacles can be avoided by offline mission planning. In this paper, we focus on avoidance of small

dynamic obstacles that can be modeled as a point. We assume each UAV maintains a safeguard distance D_{obs} for each detected obstacle. The area centered at the obstacle with a radius of D_{obs} is called the forbidden area of the obstacle, within which no trajectory should be planned. D_{obs} is determined at each UAV based on its velocity and the obstacle's velocity. The potential field for the j^{th} obstacle is defined as follows:

$$\Phi_j^o(q) = \begin{cases} \frac{\max\{v_j^o, v_s\}}{(D_s)^2}, & |qp_j^o| \leq D_s \\ \frac{\max\{v_j^o, v_s\}}{|qp_j^o|^2}, & D_s < |qp_j^o| \leq R_j^o \\ 0, & |qp_j^o| > R_j^o \end{cases} \quad (12)$$

where $\Phi_j^o(q)$ is the intensity of position q in the potential field, R_j^o is the influential range of the potential field, and $|qp_j^o|$ is the distance from q to the position of the j^{th} obstacle p_j^o . The max operator ensures the field for the swarm and the fields for the obstacles have comparable intensities when $v_j^o < v_s$.

The environment field $\Phi(q)$ is defined as follows:

$$\Phi(q) = \Phi_s(q) + \sum_{j=1}^M \Phi_j^o(q) \quad (13)$$

where M is the number of obstacles.

Fig. 3 gives an example of the environment field that contains a swarm with two UAVs and one obstacle. Fig. 3 (a) shows the layout where swarm members are depicted with asterisks, the dark dot represents the swarm virtual leader, and the circle represents the obstacle. Fig. 3 (b) shows the intensities of the constructed environment field. The left peak is the virtual leader of the swarm while the right peak is the obstacle. The area within D_{obs} around the obstacle has the maximum intensity and is forbidden for UAVs to enter. The slope of the potential field increases as q getting close to the center of the forbidden area. With such an APF model, each UAV detects higher field intensity and feels stronger threat when the swarm gets closer to the obstacle.

From Eqs. (9), (10) and (12), it can be seen that each UAV needs to know the locations and velocities of the other members and all the detected obstacles to construct the environmental field. We assume each UAV is equipped with a Lidar sensor, such as SICK TIM561 (SICK 2020), for detecting

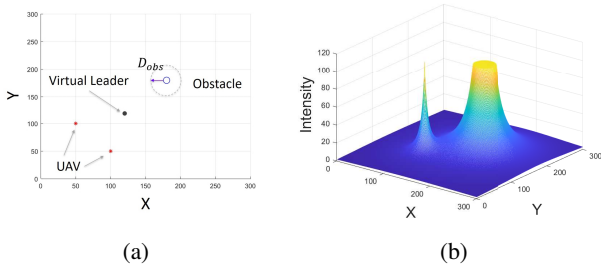


Figure 3: Example of environment field.

locations and velocities of obstacles. Each UAV is mounted with a WiFi transceiver, which is used to broadcast its own location and velocity as well as the locations and velocities for the obstacles it detects to other swarm members. Based on its own measurements and the measurements received from other swarm members, each UAV uses Kalman filter to track the location and velocity of each obstacle.

To avoid collisions with obstacles, the variation of a UAV's intensity on the environment field $|\Phi(C_t(p_1)) - \Phi(C_t(p_2))|, \forall p_1, p_2 \in [\eta_s, \eta_e]$ shall be minimized. Because intensities on the environment field indicate proximity to the obstacles. Minimizing the variations of a UAV's intensity along its trajectories on the environment field keeps it a certain distance away from the obstacle. Moreover, minimizing the variations of the UAVs' respective intensities along their trajectories also decreases the probability of collisions with each other.

In order to easily find the trajectory with minimum variation on intensities in the environmental field for a swarm member, we convert the field defined by Eq. (13) into a binary field using the following equation.

$$\Phi_b(q) = \begin{cases} 1, & \Phi(q) \geq \Phi(p_0), \\ -1, & \Phi(q) < \Phi(p_0), \end{cases} \quad (14)$$

where p_0 is the current position of UAV. Based on this binary field, the trajectory with minimum variations of intensities in the original environment field becomes an edge in the binary field, where the binary field has maximum gradient magnitudes.

Based on the above analysis, we define the second item of the fitness function as follows, aiming to find the edge in the binary field with the minimum variations of intensities.

$$f_s(C_t) = - \int_{C_t} |\nabla \Phi_b(C_t(p))| dp. \quad (15)$$

Now we show how our fitness function can guarantee a safeguard distance D_{obs} between UAVs and a detected obstacle. From Eq. (12), the area within $|qp_o| = D_{obs}$ in the obstacle's potential fields always has the maximum intensities. The influence of the swarm's potential field is negligible as its peak is on the swarm's virtual leader which is far ahead of swarm members. So based on Eq. (14), there is no edge within $|qp_o| = D_{obs}$, when the swarm is approaching the obstacle. Hence, UAVs getting closer to an obstacle than D_{obs} will be attracted to the boundary at $|qp_o| = D_{obs}$. As

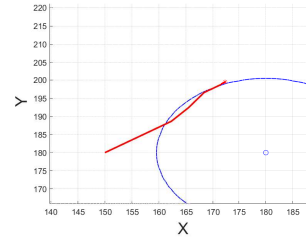


Figure 4: Illustration of a UAV entering the forbidden area being attracted to the boundary at $|qp_o| = D_{obs}$.

illustrated in Fig. 4 where the red curve is the trajectory of the UAV, and the blue circle is the boundary of the forbidden area at $|qp_o| = D_{obs}$. It shows that, once the UAV reaches into the forbidden area, it is attracted out because the gradient along the edge has the largest values. As shown in Fig. 4, a UAV will enter the forbidden area at most one planning step before it can be attracted back to the boundary. Since one planning step is very small, it has little impact on the safety.

Substituting Eq. (2) and Eq. (15) into Eq. (1), our fitness function is in the same form with the active contour model (Kass, Witkin, and Terzopoulos 1988), except that we don't have the first order derivative of a curve in our method, as the trajectory length is fixed by planning step of UAVs. Thus, a trajectory minimizing Eq.(1) is a contour on the environmental field. Therefore, another way of understanding our collision avoidance scheme is: 1) UAVs never collide with obstacles as contours never go through peaks on a potential field; 2) UAVs will never collide if all members are on different contours with safeguard distance guarantee between any two contours.

Trajectory Adjustment

The *Trajectory Adjustment* layer is used to adjust the trajectories of two swarm members to avoid collision when their distance falls below the safeguard distance D_{v2v} . The **key idea** of our trajectory adjustment scheme is to move two UAVs that fall within the safeguard distance to two different contours of the environmental field with at least D_{v2v} from each other. For ease of understanding, this adjustment of trajectories can be imagined as atoms leaping between orbits at different energy levels.

We adjust a trajectory by expanding or shrinking edges of the binary field. Since $f_s(C_t)$ in Eq. (1) always has minima on edges of the binary field, trajectories of UAVs will always be attracted to edges of the binary fields. Expanding the edge of a binary field repels UAVs away from obstacles while shrinking the edge attracts UAVs towards obstacles. Expanding and shrinking the edges of a binary field can be achieved by decreasing and increasing the intensity on the environment field, respectively.

Let $\delta\Phi_{i,j}$ be the intensity that the j^{th} UAV expects the i^{th} UAV to change. We define the magnitude of $\delta\Phi_{i,j}$ as follows:

$$|\delta\Phi_{i,j}| = \left| \frac{D_{v2v} - |p_i p_j|}{\Phi(p_i)} \right|, \quad (16)$$

where $|p_i p_j|$ is the distance between the two UAVs, and $\Phi(p_i)$ is the intensity of the i^{th} UAV in the environmental field. The sign of $\delta\Phi_{i,j}$ indicates the direction of adjustment where negative indicates moving away from obstacles and positive indicates moving towards obstacles. The sign of $\delta\Phi_{i,j}$ is determined based on the following rules:

- $\delta\Phi_{i,j}$ is negative if the i^{th} UAV is further away from the obstacle than the j^{th} UAV; otherwise it is positive.
- if both UAVs have the same distance to obstacle, the one with larger speed moves away from obstacles, generating trajectories that are longer and safer.

- if both UAVs have the same distance to obstacle and the same speed, the one with larger ID moves away from the obstacle.

From Eq. (16), it can be seen that $\delta\Phi_{i,j}$ can be imagined as virtual repulsive force enforced by the j^{th} UAV with amplitude depending on UAV-to-UAV distance and the intensity of the i^{th} UAV in the environmental field. When the distance between the two UAVs is small, $|\delta\Phi_{i,j}|$ is large. When the i^{th} UAV is closer to the obstacle than the j^{th} UAV, $|\delta\Phi_{i,j}| < |\delta\Phi_{j,i}|$, thereby discouraging UAVs from getting closer to obstacles and encouraging UAVs to move away from obstacles.

The net intensity change of the i^{th} UAV is calculated as

$$\delta\Phi_i = \sum_{j=1, j \neq i}^N \delta\Phi_{i,j}. \quad (17)$$

Based on the sign of $\delta\Phi_i$, the i^{th} UAV decides whether to move away or towards the obstacle. The adjustment is terminated once the distance between any two swarm members is no less than the safeguard distance D_{v2v} .

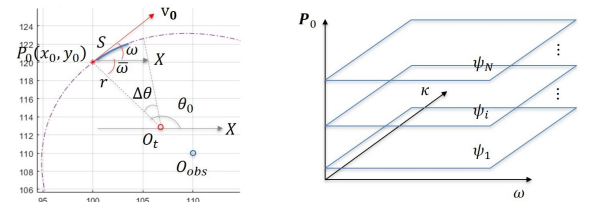
PSO-based Trajectory Planning

In our PSO-based trajectory planning, each particle represents a candidate trajectory. To achieve fast convergence, a compact expression of the trajectory candidates is needed to reduce the dimension of the search space for PSO.

Since each step is small, we use an arc to represent a trajectory as it is more natural for turnings of multi-rotor copters and matches their aerodynamics. As illustrated in Fig. 5a, an arc can be expressed as $S(\omega, \kappa)$ with two parameters where $\kappa = d\omega/dS$ is curvature of the arc and ω is the slope of the arc relative to the X -axis of the coordinate system. When $\kappa = 0$, it is a straight line along the UAV's current velocity, which means the UAV has no turning. Any point $P_i = [x_i(\omega, \kappa), y_i(\omega, \kappa)]$ on the arc can be expressed by

$$\begin{aligned} x_i(\omega, \kappa) &= \frac{\cos(\theta_i)}{\kappa} + \frac{\cos(\bar{\omega})}{\kappa} + x_0, \\ y_i(\omega, \kappa) &= \frac{\sin(\theta_i)}{\kappa} + \frac{\sin(\bar{\omega})}{\kappa} + y_0, \\ \theta_i &\in [\theta_0 - \Delta\theta, \theta_0]. \end{aligned} \quad (18)$$

where θ_i is the slope of the vector from the turning center O_t to a point on the arc, and $\Delta\theta$ is the range of θ_i .



(a) Parameterization of trajectory. (b) Search space for PSO.

Figure 5: Parameterization and search space.

It can be seen that an arc is uniquely located based on its $[\omega, \kappa]$ and initial point $[x_0, y_0]$. Hence, a particle can be expressed as $\xi = [\omega, \kappa]_{[x_0, y_0]}$. Therefore, the search space for trajectory planning is reduced to two dimensional. As illustrated in Fig. 5b, solutions for different UAVs exist in orthogonal sub-spaces where ψ_i is the sub-space for the i^{th} UAV. UAVs in a swarm search for solutions simultaneously.

Based on the representation of trajectories in Eq. (18), the particles can be updated as follows:

$$\begin{aligned} v_i^{t+1} &= \mu_0 v_i^t + \mu_1 (p_i^t - \xi_i^t) + \mu_2 (p_g^t - \xi_i^t) \\ \xi_i^{t+1} &= \xi_i^t + v_i^{t+1} \end{aligned} \quad (19)$$

where v_i^t and ξ_i^t are the velocity and position of particle i at time t , p_i^t is the personal best experience of particle i at time t and p_g^t is the global best experience of the swarm at time t . μ_0 is an inertia weight, μ_1 and μ_2 are acceleration coefficients uniformly distributed in $[0, 1]$ independently.

Performance Evaluation

Simulation Setup

In our simulations, an obstacle is detected when the distance between the obstacle and any UAV is smaller than the detection range of the Lidar sensor $R_d = 50m$. A swarm starts to avoid the obstacle when the distance of any UAV to the obstacle is smaller than the avoiding distance $D_a = 30m$. Since industrial UAVs usually have a maximum ground speed of $10 m/s$ to $17 m/s$ (DJI 2020a) and consumer level UAVs commonly have a ground speed of $5 m/s$ to $14 m/s$ (DJI 2020b), we set $v_s = 10 m/s$ in all simulations. An obstacle can be either static or dynamic with velocity v_{obs} varying from 0 to $10 m/s$. In our simulations, the swarm members are equally spaced in a circle with a radius $\tau = 20 m$, and the number of UAVs in the swarm N is varied from 2 to 10 .

We compare E^2Coop with the following two schemes:

- **FFPSO** (Parker, Butterworth, and Luo 2019): A new term repelling particles from one another is introduced to the velocity update equation in PSO. The fitness function of a particle is simply the negative of the distance between the particle and its destination.
- **PPSO** (Cai and Yang 2014): A smoothing field in APF is introduced in order to smooth the trajectories of UAVs. Then, PSO is adopted to find the optimal trajectories on APF. The fitness value of particles is just the field intensity.

All the three schemes (E^2Coop , FFPSO and PPSO) are tested in two scenarios: 1) *Obstacle in Front* where the initial distance between the swarm and the obstacle is $200 m$, and 2) *Obstacle in Side* where the obstacle appears either on the left-front or right-front side of the swarm, the initial distance between the swarm and the obstacle is $200 m$, and v_{obs} is perpendicular to v_s . In our simulations, we observed that the minimum distance between obstacles and trajectories generated by FFPSO and PPSO is $26 m$ and $16 m$, respectively, and the minimum UAV-to-UAV distance in the trajectories generated by FFPSO and PPSO is around $0.5 m$.

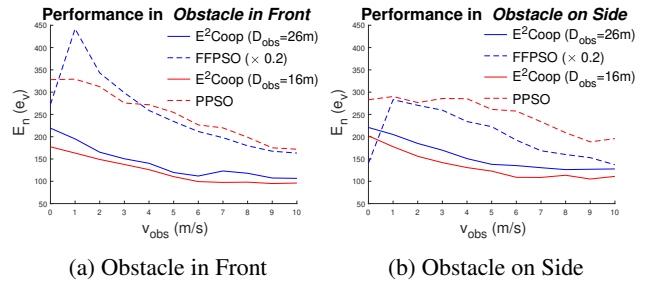


Figure 6: Energy consumption.

For a fair comparison on energy consumption, we set D_{obs} in E^2Coop to $26 m$ and $16 m$, and D_{v2v} in E^2Coop is set to $0.5 m$. Each combination is repeated 10 times.

Simulation Results

Fig. 6 shows the energy consumption for three schemes in both scenarios with the variation of obstacle speed. The energy consumption of FFPSO is scaled by 0.2 for a clear comparison. It can be seen that E^2Coop has the lowest energy consumption among all schemes in both scenarios. This is because the fitness function we designed aims to minimize energy consumption by minimizing trajectory's curvature. Moreover, the environment field constructed by the swarm provides a global coordination among swarm members for collision avoidance. Hence, the trajectories for individual UAVs do not conflict with each other. FFPSO performs the poorest because there is no such a scheme that smooths the trajectories. Although PPSO has a smoothing field in its APF method, each UAV plans its own trajectory without a global coordination. As a result, the trajectories generated by PPSO are still very zigzag.

For all three schemes, energy consumption decreases as the obstacle's speed increases. This is because that, when the obstacle is at a high speed, the avoidance of UAVs starts early and finishes quickly. This causes their trajectories to be short and have less turnings. An abrupt change can be spot on the curve of FFPSO in both scenarios at $v_{obs} = 1 m/s$. This is because the repelling force of obstacles pushes the particles to the direction of v_{obs} , and hence causes a very sharp turning on their trajectories. So the energy consumption of FFPSO drastically increases when v_{obs} changes to $1 m/s$. Comparing Fig. 6a and Fig. 6b, the energy consumption of FFPSO is smaller when the obstacle comes from side of the swarm, whereas the other two schemes remain roughly the same. This is because an obstacle coming from a side of the swarm pushes the trajectories slightly deviate from v_s . Overall, the results show that E^2Coop can save up to 51% energy in *Obstacle in Front* and 46% in *Obstacle on Side* in comparison with the other two schemes.

Parameter Analysis

Important parameters of E^2Coop include D_{obs} , D_{v2v} , λ_1 and λ_2 . Given a setting on v_s and v_{obs} , large D_{obs} is expected to prolong trajectories, and large D_{v2v} is also expected to prolong and curve trajectories. λ_1 and λ_2 are linear

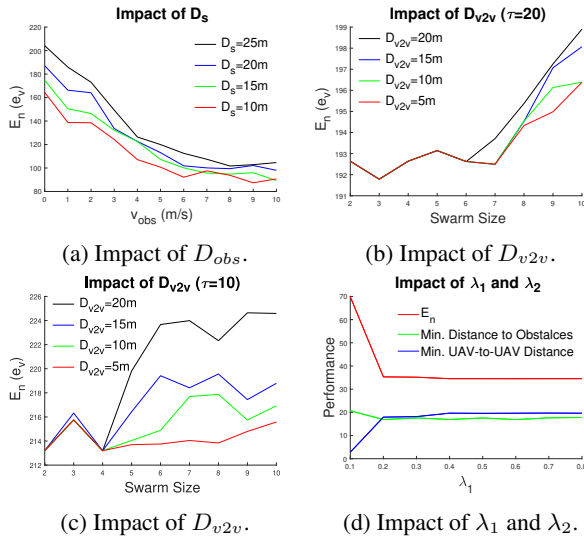


Figure 7: Parameter analysis results.

weights of energy efficiency and safety concerns in our fitness function (1). With $\lambda_1 + \lambda_2 = 1$, energy efficiency and safety appear to be a pair of trade-off. We now study the impact of these parameters on the performance of E^2Coop using the scenario of *Obstacle in Front*. Each combination is repeated 3 times. The simulation results are shown in Fig. 7.

Impact of D_{obs} From Fig. 7a, it can be seen that energy consumption of E^2Coop increases with the increase of D_{obs} under all obstacle speeds. This is because larger D_{obs} forces the trajectories to go further away from the obstacle. Therefore, the curvature and length of the trajectories are larger, and hence more energy consumption is required.

Impact of D_{v2v} From Fig. 7b and Fig. 7c, it can be seen that energy consumption of E^2Coop increases with swarm size. This is because UAV members must remain a certain distance away from each other during avoidance. Trajectories of individual UAVs have to be longer and more curved when more members are in a swarm. Fig. 7b and Fig. 7c show the energy consumption when the swarms are in a formation of circle with a radius of 20 and 10 m, respectively. In Fig. 7b, energy consumption of E^2Coop starts to increase with D_{v2v} from $N = 6$. While in Fig. 7c, energy consumption of E^2Coop starts to increase with D_{v2v} from $N = 4$. This is because the impact of D_{v2v} on energy is small when swarm size is small. The impact of D_{v2v} on energy is large when UAVs are close to each other.

Impact of λ_1 and λ_2 From Fig. 7d, we observe that in extreme cases when $\lambda_1 \leq 0.1$, E^2Coop is still able to find trajectories to the destination. However, the trajectories have many turnings as the smoothing term in our fitness function has a small weight. Hence, energy consumption is high. What's more, the minimum UAV-to-UAV distance is very small, while the minimum distance to obstacles is relatively large. This implies that in such extreme cases, UAVs can avoid obstacles, but UAV members may have the risk to

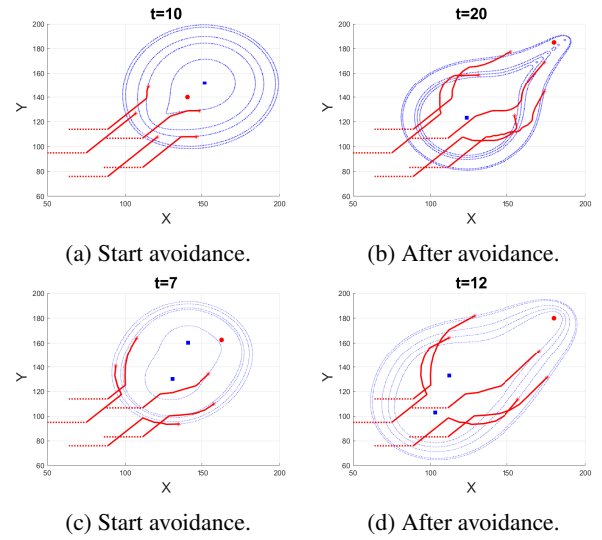


Figure 8: Demonstration of generated trajectories.

collide with each other. When λ_1 is larger than 0.4, energy consumption, minimum distance to obstacles, and minimum UAV-to-UAV distance become stable and converge to a certain level. When $\lambda_1 = 0.9$ ($\lambda_2 = 0.1$), we observe that E^2Coop fails to find trajectories to destination. This is because the second term in our fitness function has a small weight, so UAVs can't find contours on the environment field. It can be seen that the performance of E^2Coop is not very sensitive to the selection of λ_1 and λ_2 .

ODA Demonstration

Fig. 8 shows the trajectories generated by E^2Coop , where $\lambda_1 = \lambda_2 = 0.5$. Five UAVs are in a swarm with $v_s = 5m/s$ and $v_{obs} = 10m/s$. D_{v2v} and D_{obs} are set to 10m and 5m, respectively. Fig. 8a and 8b are single-obstacle scenario, while Fig. 8c and 8d are double-obstacle scenario. The virtual leader of the swarm and geometrical center of the obstacle are shown in red circle and blue square, respectively. Planned trajectories are shown in red solid curves. Contours are shown in blue solid curves to help reflect the deformation of the environment field.

Conclusions

In this paper we investigate the problem of trajectory planning for UAV swarms to avoid collisions in an energy efficient manner. E^2Coop , a hybrid scheme that combines APF and PSO is proposed. E^2Coop is designed to take advantage of the search ability of PSO and the environmental representations of APF to coordinate the trajectory planning of swarm members in a global perspective. Results demonstrate that E^2Coop can generate safe trajectories and consume much less energy than the compared schemes. Regarding large-scale aerial applications, future work can be done to extend it for three dimensional trajectory planning.

Acknowledgments

The authors would like to thank the Air Traffic Management Research Institute, Nanyang Technological University, Singapore for helping conduct the field trials.

References

- ATMRI. 2020. Field trials full video. <https://youtu.be/EyZzIUlmAII>. (Accessed on 03/05/2021).
- Cai, Y.; and Yang, S. X. 2014. A potential field-based PSO approach for cooperative target searching of multi-robots. In *Proceeding of the 11th World Congress on Intelligent Control and Automation*, 1029–1034. IEEE.
- DJI. 2020a. MATRICE 300 RTK - Specifications - DJI. <https://www.dji.com/nz/matrice-300/specs>. (Accessed on 03/17/2021).
- DJI. 2020b. Mavic Air 2 - Specifications - DJI. <https://www.dji.com/nz/mavic-air-2/specs>. (Accessed on 03/17/2021).
- Huang, S.; and Low, K. 2018. A Path Planning Algorithm for Smooth Trajectories of Unmanned Aerial Vehicles via Potential Fields. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 1677–1684. IEEE.
- Kass, M.; Witkin, A.; and Terzopoulos, D. 1988. Snakes: Active contour models. *International journal of computer vision* 1(4): 321–331.
- Liang, X.; Qu, X.; Wang, N.; Li, Y.; and Zhang, R. 2019. Swarm control with collision avoidance for multiple under-actuated surface vehicles. *Ocean Engineering* 191: 106516.
- Liu, Z.; Jiang, Z.; Xu, T.; Cheng, H.; Xie, Z.; and Lin, L. 2018. Avoidance of High-Speed Obstacles Based on Velocity Obstacles. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 7624–7630. IEEE.
- Parker, L.; Butterworth, J.; and Luo, S. 2019. Fly safe: Aerial swarm robotics using force field particle swarm optimisation. *arXiv preprint arXiv:1907.07647*.
- Pradhan, B.; Nandi, A.; Hui, N. B.; Roy, D. S.; and Rodrigues, J. 2019. A Novel Hybrid Neural Network Based Multi Robot Path Planning with Motion Coordination. *IEEE Transactions on Vehicular Technology*.
- Shan, F.; Luo, J.; Xiong, R.; Wu, W.; and Li, J. 2020. Looking before Crossing: An Optimal Algorithm to Minimize UAV Energy by Speed Scheduling with a Practical Flight Energy Model. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, 1758–1767. IEEE.
- SICK. 2020. TiM561-2050101 — Detection and ranging solutions — SICK. <https://www.sick.com/us/en/detection-and-ranging-solutions/2d-lidar-sensors/tim5xx/tim561-2050101/p/p369446>. (Accessed on 03/17/2021).
- Snape, J.; Van Den Berg, J.; Guy, S. J.; and Manocha, D. 2009. Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5917–5922. IEEE.
- Snape, J.; Van Den Berg, J.; Guy, S. J.; and Manocha, D. 2011. The hybrid reciprocal velocity obstacle. *IEEE Transactions on Robotics* 27(4): 696–706.
- Song, B.; Wang, Z.; Zou, L.; Xu, L.; and Alsaadi, F. E. 2019. A new approach to smooth global path planning of mobile robots with kinematic constraints. *International Journal of Machine Learning and Cybernetics* 10(1): 107–119.
- Stolaroff, J. K.; Samaras, C.; O'Neill, E. R.; Lubers, A.; Mitchell, A. S.; and Ceperley, D. 2018. Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. *Nature communications* 9(1): 409.
- Tan, Q.; Wang, Z.; Ong, Y.-S.; and Low, K. H. 2019. Evolutionary optimization-based mission planning for UAS traffic management (UTM). In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 952–958. IEEE.
- Tharwat, A.; Elhoseny, M.; Hassanien, A. E.; Gabel, T.; and Kumar, A. 2019. Intelligent Bézier curve-based path planning model using Chaotic Particle Swarm Optimization algorithm. *Cluster Computing* 22(2): 4745–4766.
- Tsang, K. F. E.; Ni, Y.; Wong, C. F. R.; and Shi, L. 2018. A Novel Warehouse Multi-Robot Automation System with Semi-Complete and Computationally Efficient Path Planning and Adaptive Genetic Task Allocation Algorithms. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 1671–1676. IEEE.
- Wallar, A.; and Plaku, E. 2013. Path planning for swarms by combining probabilistic roadmaps and potential fields. In *Conference Towards Autonomous Robotic Systems*, 417–428. Springer.
- Xia, G.; Han, Z.; Zhao, B.; and Wang, X. 2020. Local Path Planning for Unmanned Surface Vehicle Collision Avoidance Based on Modified Quantum Particle Swarm Optimization. *Complexity* 2020.