UAV Path Planning for Maximum Visibility of Ground Targets in an Urban Area

Jongrae Kim
Department of Aerospace Engineering
University of Glasgow
Glasgow, UK.
jkim@aero.gla.ac.uk

John L. Crassidis
Dept. of Mechanical & Aerospace Engineering
University at Buffalo
State University of New York
Amherst, NY, U.S.A.
johnc@buffalo.edu

Abstract − *Multiple moving targets in an urban area are to be tracked simultaneously by unmanned aerial vehicles. It is assumed that the moving targets try to avoid the camera field of view of the aircraft by changing their velocities and/or hiding behind buildings. The number of aircraft is much smaller than the number of targets, in general. In order to track as many targets as possible, firstly the targets are grouped into a number of subgroups by maximising the modularity, which is solved efficiently by the power iteration. Secondly, circular optimal paths are assigned to maximise the visibility of the area, given shapes and locations of the ground obstacles, where the computational complexity is reduced using a novel random sampling method. Finally, the aircraft transition paths from the current positions to the desired path are obtained by solving a discrete minimum weighted path length problem.*

Keywords: Target Tracking, Path Planning, UAV

1 Introduction

For the last decade, Unmanned Aerial Vehicles (UAVs) have been used for various purposes in civilian and military operations. Many practical issues in UAV dynamics and control have been resolved. However, many other problems are yet to be solved in order to make UAVs fully autonomous. In addition, mission demands in various operations are more complicated now than before.

Ground moving target searching and tracking using cameras on UAVs are of great interest, especially for operations over an urban area [4, 5, 12]. A receding-horizon cooperative search algorithm using the UAV routes and orientations is presented in [4]. Computational complexity in the tracking problem in an urban area is acknowledged in [5] and [12]. A randomisation based optimisation approach [5] and a genetic algorithm [12] are used to reduce the computational complexity. Based on a nonlinear programming approach a direct collocation method is presented for the path planning in [3] but unsatisfiable computation speed is noted. In addition, an optimal separation strategy for assigning multiple UAVs in a region is presented in [6].

In most of the previous results, tracking has been demonstrated for only a few targets. However, in reality it may be necessary to track all moving targets feeding back from Ground Moving Target Indicator (GMTI) sensors in order to further classify the targets to determine whether or not they need to be continued to be tracked or not [10, 11]. The scenario to be considered is the case that the number of moving targets to be tracked is much larger than the number of aircraft. Note that visible targets are yet to be classified and the objective of the proposed algorithm is a priory algorithm before the classification. The algorithm maximises the visibility of all moving objects given by the GMTI sensor output.

The formal definition of the problem will be presented next. Three main parts of path planning algorithm will be presented: dividing given targets into a few subgroups; design of optimal circular paths for tracking each subgroup; development of an optimal transition guidance law for the aircraft from the current location to the desired circular path. Finally, the performance of the algorithms is demonstrated by a random scenario Monte-Carlo simulation.

2 Scenario

In an area multiple ground targets, of which the number, n_t, is unknown, are moving within a certain range of speeds to avoid observation by UAV cameras. Several UAVs, where the number of UAVs is n_a, are to fly over an area to monitor the targets. The area is densely populated by urban obstacles, e.g., high-rise buildings as shown in Figure 1, where $n_t = 3$ and $n_a = 2$. The shape and the location of each building or obstacle, which may be used for hiding targets, are assumed to be known.

The main objective is to position UAVs in order to maximise the visibility of all targets to cameras or

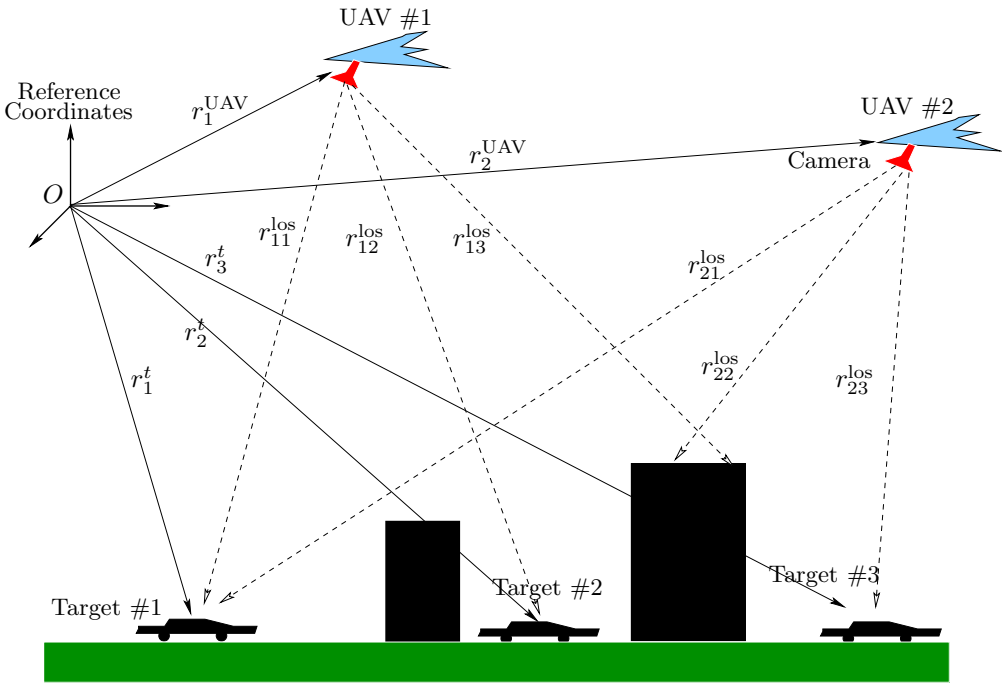

Figure 1: Two UAVs tracking three targets using cameras

equivalently minimise the length of blackout time for each target. The blackout time is the continuous time interval length, measured from the target, lost from all cameras. The corresponding cost function can be formulated as follows:

$$\underset{r_i^{\text{UAV}} \in \mathcal{X}}{\text{minimise}} \; J\left(r_{ij}^{\text{los}}\right) = \int_{t_0}^{t_f} n_{\text{unobv}}(t)dt \tag{1}$$

where r_i^{UAV} is the i-th UAV position vector with respect to the reference coordinates for $i = 1, 2, \ldots, n_a$, $\mathcal{X}$ is an operation area, r_{ij}^{los} is the line of sight vector from the i-th UAV to the j-th target for $i = 1, 2, \ldots, n_a$ and $j = 1, 2, \ldots, n_t$, and $n_{\text{unobv}}(t)$ is the current number of unobservable targets, which can be defined by

$$n_{\text{unobv}}(t) = \sum_{j=1}^{n_t} I_j(t) \tag{2}$$

for $j = 1, 2, \ldots, n_t$, where $I_j(t)$ is an indicator function whether a target is blocked or not by any obstacles, which is defined as follows:

$$I_j(t) = \begin{cases} 1, \; r_{ij}^{\text{los}}(t) \text{ blocked } \forall i \in \mathbb{N}_a \\ 0, \; \text{otherwise} \end{cases} \tag{3}$$

where $\mathbb{N}_a = \{1, 2, \ldots, n_a\}$.

The number of targets, n_t, is unknown in general. The cost function to be minimised is as follows:

$$\underset{r_i^{\text{UAV}} \in \mathcal{X}}{\text{minimise}} \; J\left(r_{ij}^{\text{los}}\right) = \int_{t_0}^{t_f} \hat{n}_{\text{unobv}}(t)dt \tag{4}$$

where

$$\hat{n}_{\text{unobv}}(t) = \sum_{j=1}^{\hat{n}_t} I_j(t) \tag{5}$$

and $\hat{n}_t$ is the number of targets that are identified, which can be larger or smaller than n_t as it may include false targets.

The cost function, (4), could be minimised by each UAV independently, i.e. decentralised control, and some communication laws could be introduced to ensure that better minimisation would occur, i.e. cooperative control. However, in order to obtain the best performance guideline, full information is shared by all UAVs for the first two steps of the algorithms, i.e. centralised control: grouping and optimal circular path design. On the other hand, a centralised optimal path transition for each UAV will restrict the practical applicability of the algorithms and it will need to be performed in each UAV separately.

3 UAV Path Planning

Three main parts of the UAV path planning algorithm, i.e. grouping, optimal circular paths and optimal transition to the paths, are derived.

The number of identified targets, $\hat{n}_t$, is greater than the number of UAVs, n_a, in general. The targets have to be grouped into a certain number, which could be equal to or less than n_a.

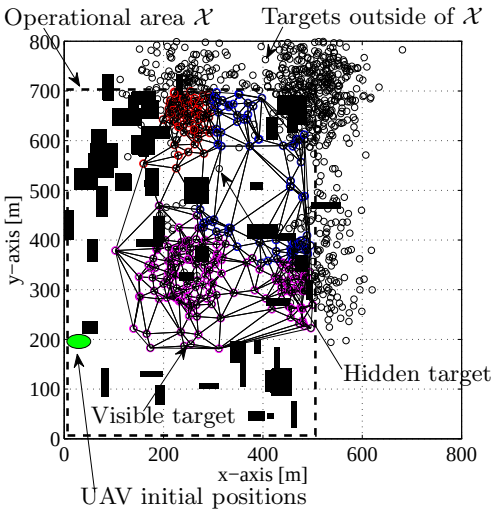

Figure 2: Three UAVs find 155 targets among 424 targets inside $\mathcal{X}$. Total number of targets is 1,000.

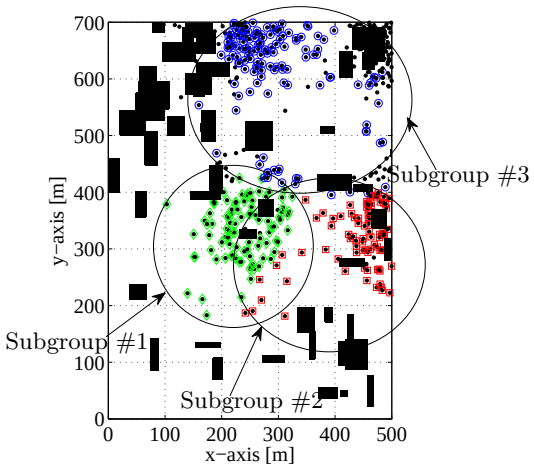

Figure 3: Visible targets are divided into three subgroups.

3.1 Grouping

Figure 2 shows one random scenario, where 1,000 ground targets are present and 424 of them are inside the operational area, $\mathcal{X}$, which is defined by

$$\mathcal{X} = \{(x, y) | 0 \leq x \leq 500 \text{ m and } 0 \leq y \leq 700 \text{ m}\} \quad (6)$$

and the three UAVs current locations are approximately $x = 0$ m, $y = 200$ m and the altitude $z = 100$ m. The visible targets are connected by Delaunay triangulation [9].

Based on the triangulation, the Laplacian matrix, A, can be constructed whose elements a_{ij} are 1 if two targets, the i-th and j-th targets, are connected to each other by the triangulation or 0 if they are not directly connected. This is a bi-directional graph, i.e. no directionality in the connections and the Laplacian matrix, A, is symmetric matrix whose elements are 1 or 0, for $i, j = 1, 2, \ldots, \hat{n}_t$. In order to divide the groups into subgroups or modules, the following modularity (Q) definition for graphs is adopted [8]:

$$Q = \frac{1}{4m}\mathbf{s}^T B \mathbf{s} \quad (7)$$

where m is the total number of edges in the graph,

$$B = A - \frac{1}{2m}\mathbf{k}\mathbf{k}^T, \quad (8)$$

Each element of the column vector $\mathbf{k}$ is the degree of each target, i.e. the number of edges connected to each target, and $\mathbf{s}$ is a column vector whose dimensions is equal to $\hat{n}_t$ and element is 1 or -1.

To maximise the modularity, Q, $\mathbf{s}$ must be chosen such that its direction is aligned to the eigenvector corresponding to the maximum eigenvalue [8]. Although the power iteration has certain limitations [7], for the modularity maximisation problem it gives solutions reasonably fast. The power iteration algorithm is implemented as follows:

1. Choose random vector $\mathbf{x}_k \in \mathbb{R}^{n_t}$
2. Update $\mathbf{x}_k$ using $a\mathbf{x}_{k+1} = B\mathbf{x}_k$
3. Normalise $\mathbf{x}_{k+1}$ as follows: $\mathbf{x}_{k+1} = \mathbf{x}_{k+1}/\|\mathbf{x}_{k+1}\|$
4. If $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq \epsilon$, then stop the iteration, otherwise go to step 1, where ϵ is a tolerance value.

The above algorithm only splits the given graph into two. In order to split more than two subgroups, each group must be split into two until the number of subgroups is equal to the desired number, where the subgroup whose number of targets is the largest is always chosen to be split further into two smaller subgroups. However, as it is pointed out in [8], the subgroup matrix B^g, whose dimension is $n_g \times n_g$, is not obtained by just simply removing rows and columns of the original B, which are not in the same group, but it should be modified as follows:

$$\left[b^g_{ij}\right] = [b_{ij}] - \text{diag}\left[\sum b_{1j} \quad \sum b_{2j} \quad \ldots \quad \sum b_{n_g j}\right] \quad (9)$$

where b^g_{ij} and b_{ij} are the i-th row and the j-th column element of B^g and B matrix, respectively, and (i, j) belongs to the subgroup. More details can be found in [8].

Using the power iteration, for example, dividing 155 targets into 3 subgroups takes less than 0.8s on an Intel Core 2 Quad-Core, 2.4GHz, 4GB RAM, ubuntu linux 9.10, MATLAB 7.9. The result is shown in Figure 3.

3.2 Optimal Circular Path

The maximum visibility of a target may be obtained by minimising the distance between a UAV and the

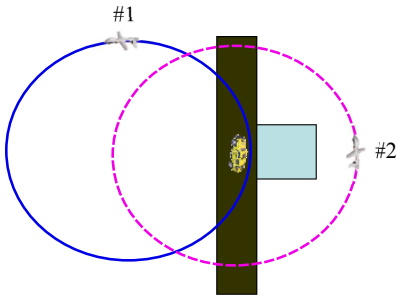

Figure 4: Optimal circular path centre

target on the x-y plane and maximising the altitude, i.e. the z-coordinate. Therefore, the altitude can be fixed to the highest possible altitude, where it can be determined by the UAV operational altitude, the performance of the camera on UAV, weather conditions, etc.

The UAV is assumed to be a fixed-wing type and it does not have any hovering capability, i.e. the minimum speed of the UAV is nonzero. Therefore, keeping the distance on the x-y plane is not straightforward. However, one obvious choice would be the minimum radius turn, where the minimum radius is determined by the UAV flying capability and/or some operational reasons. Intuitively, it seems that placing the centre of the circular path on the target location would minimise the average distance between the target and the UAV as shown in Figure 4, which is the circle that #2 UAV is following. But, the line of sights around the location of #2 UAV would be blocked by the building on the left hand side of the UAV. Therefore, it would be better to move the centre of the circular path to the left as shown in the figure, where #1 UAV follows. Hence, the problem to maximise the visibility for a fixed target is not equivalent to just minimising the distance between the UAV and the target on the x-y plane. The information about the ground obstacles must be exploited.

Full optimisation, including shapes and sizes of ground obstacles, increases the computational time to be unreasonably large. This is overcome efficiently by using a novel random sampling method presented in [5]. The main idea is as follows. Firstly, a finite number of points are sampled on the ground and buildings. Secondly, whether each line of sight vector from the UAV to the ground sampling points is blocked by any sampled points on buildings or obstacles is checked. This point cost value at a certain location of a circular path is integrated along the circular path. Thirdly, the worst direction where the line of sight vectors are blocked the most is found by selecting the direction corresponding to the maximum value of the point cost. Finally, the centre of the flight path is moved into the opposite direction of the worst direction until the integration cost is minimised. More details about the algorithm can be found in [5].

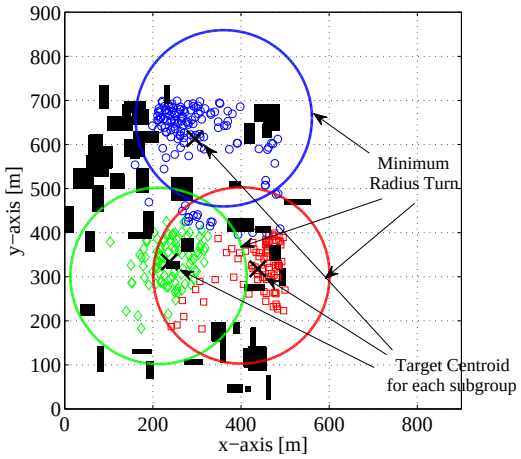

Figure 5: Three centroids for each subgroup is indicated are black crosses and three optimised minimum radius turns are shown by three circles.

Figure 5 shows an example of three circular paths for a given scenario. The optimal centres of the three circular paths do not coincide with the centre of the target centroid, which are indicated in the black crosses. The calculation time for obtaining the three circular paths are about 2.14s on the same computer as described in the previous section.

3.3 Optimal Transition

The last part of the algorithm involves directing each UAV to arrive at each desired circular flight path. For a given current location and velocity of the UAV, the shortest path to the desired circular path, which is designed in the previous step or was received from a commanding centre, is to be calculated. The problem can be formulated as a two-point boundary value problem as follows:

$$\underset{u(t)\in U}{\text{minimise}}\, J_s = \int_{t_0}^{t_f} ds \tag{10}$$

subject to the following inequality constraints

$$0 < v_{\min} \leq \sqrt{\mathbf{v}\cdot\mathbf{v}} \leq v_{\max} \tag{11a}$$

$$u_{x_{\min}} \leq u_x \cos\phi + u_y \sin\phi \leq u_{x_{\max}} \tag{11b}$$

$$u_{y_{\min}} \leq -u_x \sin\phi + u_y \cos\phi \leq u_{x_{\max}} \tag{11c}$$

$$\frac{|\mathbf{v}\times\mathbf{u}|}{\left(v_x^2+v_y^2\right)^{3/2}} \leq \frac{1}{r_{\min}}, \tag{11d}$$

an equality constraint, which is the aircraft dynamics,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u}) \tag{12}$$

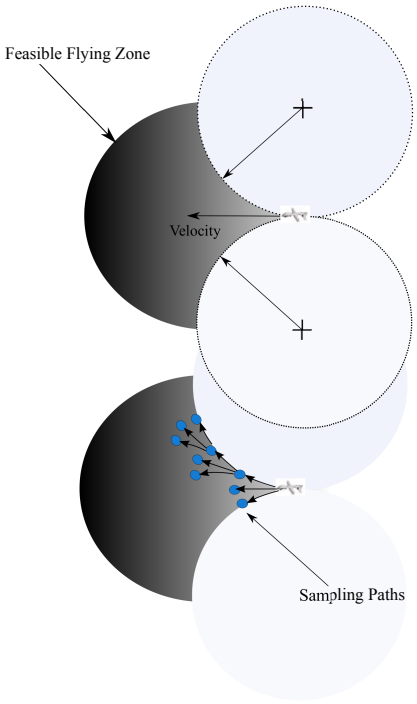

Figure 6: Feasible flying zone and discretisation

and the final equality constraints

$$\|\Delta\mathbf{r}\| = \|\mathbf{r}^{\mathrm{UAV}} - \mathbf{r}_c\| = r_{\min} \tag{13a}$$

$$\mathbf{v} \cdot \Delta\mathbf{r} = 0 \tag{13b}$$

$$\mathbf{v} \times \Delta\mathbf{r} = \gamma\mathbf{k} \tag{13c}$$

where s is the arc length, U is the control input set, which is defined by (11b), (11c) and (11d), t_0 is the current time, the final time t_f is free, the control input constraints are in the aircraft body coordinates assuming that the x-axis of aircraft is aligned with the current velocity, $\phi = \tan^{-1}(v_y/v_x)$, v_x and v_y are the x or y directional aircraft velocity, respectively, $\mathbf{u}$ is the control input, $\mathbf{r}_c$ is the centre of the flight path designed in the previous section. Equation (11d) is the curvature constraint for the minimum radius turn, γ is a constant and $\mathbf{k}$ is the unit vector towards the z-axis. Equation (13c) is the flying direction constraint. The aircraft must fly along the circular path in the counter clockwise direction ($\gamma < 0$) or the clockwise direction ($\gamma > 0$). The flying direction must be consistent among aircraft when multiple UAVs are assigned to a same circular path with some separation angle.

However, solving the optimisation problem may not be feasible as it should be solved by the on-board computer of the aircraft in realtime. In order to reduce the computational burden the continuous problem is transformed into a discrete problem. Figure 6 shows a feasible flying zone for an aircraft, where two dotted lines are circles with each radius equal to the aircraft minimum radius turn. The grey area is the feasible flying zone for the current direction of flight during a certain length of time interval. Then, the feasible flying zone is sampled as shown in the lower picture in Figure 6.

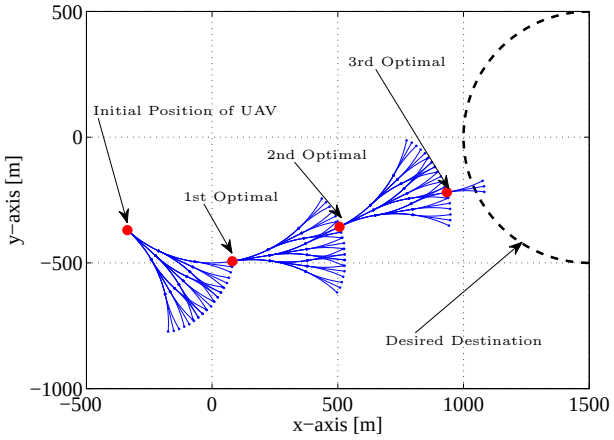

Figure 7: Discrete path sampling

The closest three dots from the current aircraft location are the three possible flights from t_0 to $t_0 + \Delta t$, where Δt is a control variable for the path planning algorithm. The line in the middle is the straight flight path and the other two correspond to the minimum radius turn towards up and down, respectively. The number of points to be sampled at each stage, n_s, is also a control variable, where n_s is equal to three in Figure 6. For each sampled point another n_s is sampled in a similar way as demonstrated in the figure. This is repeated until $t_0 + N\Delta t$ reaches the desired length of time, where N is another control variable, whose value is an integer and $T = N\Delta t$ is the length of optimisation horizon.

As n_s and N increase and Δt decreases, the solution will approach the original continuous problem. The total number of sampling points is equal to

$$n_s^{\mathrm{total}} = \sum_{i=1}^{N} n_s^k \tag{14}$$

which increases exponentially with N or $T/\Delta t$. Hence, n_s and δt could be large but T must be kept small. Figure 7 shows an example path for $n_s = 3$, $\Delta t = 5$s and $T = 15$s. Therefore, the total number of sampling points for each optimisation is equal to 36.

To solve the optimisation problem a point cost for each sampling point is to be assigned. By the definition of a feasible flying zone, all sampling points satisfy the inequality constraints given in (11). Therefore, the point cost could be set to a value proportional to the final constraint violation, (13). To do this, the point cost is defined as follows:

$$J_i = \|\mathbf{c}_i\| \tag{15}$$

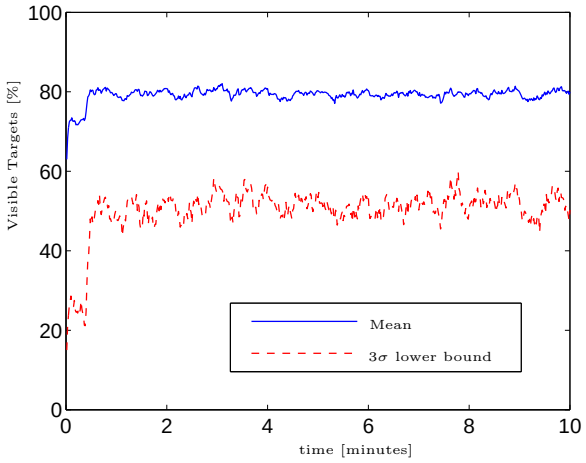

Figure 8: Mean and 3σ lower bound for the proportion of the number of targets inside the camera field of view

for $i = 1, 2, \ldots, n_s^{\text{total}}$, where J_i is the cost for each sampling point,

$$\mathbf{c}_i = \begin{bmatrix} \|\Delta\mathbf{r}_i\|/r_{\min} - 1 \\ (\mathbf{v}_i/\|\mathbf{v}_i\|) \cdot (\Delta\mathbf{r}_i/\|\Delta\mathbf{r}_i\|) \\ \mathbf{g}_i \end{bmatrix}, \tag{16}$$

$$\mathbf{g} = \begin{cases} \mathbf{0}, & \text{for } \gamma < 0 \\ \alpha \left\| \dfrac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \times \dfrac{\Delta\mathbf{r}_i}{\|\Delta\mathbf{r}_i\|} \right\|, & \text{otherwise} \end{cases} \tag{17}$$

for the counter clockwise direction constraint, $\Delta\mathbf{r}_i = \mathbf{r}_i - \mathbf{r}_c$, $\mathbf{r}_i$ is the coordinate of each sampling point, $\mathbf{v}_i$ is the velocity for each sampling point, and α is a constant to indicate the relative importance of the rotational direction. Note that for some cases if α is too small, then the path may converge to the one whose rotational direction is opposite from the desired one, i.e. it may converge to a local minimum. Therefore, it is safer to set α at least greater than 2.

4 Simulation

The performance of the algorithm is tested for 100 random maps, where the average number of targets (n_t) inside the operational area is around 200, the number of UAVs (n_a) is three, the targets are grouped as 3 subgroups, the targets are re-grouped every minute, maximum and minimum velocities of the UAVs are 20 m/s and 30 m/s, respectively, the minimum radius turn of the UAVs is 200 m, n_s, Δt, N and α in (17) for discrete sampling are equal to 3, 1 s, 10 and 10, respectively, and the maximum altitude of aircraft is 100 m. The size of the operational area is 500 m in width and 700 m in length, which is defined in (6), populated by buildings whose height is between 5 m to 50 m. Finally, the maximum target velocity is 20 m/s.

The average proportions of the targets visible from the aircraft at each instant are shown in Figure 8. The mean is about 80% and the 3σ lower bound is about 50%. At near the initial time between 0 and 0.5 minutes, the mean and the lower bound are worse than the other intervals. This is mainly caused by the initial velocity directions of the aircraft, which are chosen randomly and sometimes the velocity directions are opposite from the region where most targets are located. However, as shown in the figure these are quickly recovered by subsequent manoeuvres in less than a minute. Finally, note that although the objective flight path is given by the minimum radius turn circular path, because of dynamic target relocation and the limitation from the aircraft manoeuvrability the actual paths for each aircraft is not always circular path unless all targets are completely stationary for long time interval.

5 Conclusions

A mission planning algorithm for a group of UAVs to track multiple moving ground targets in an urban area is presented. By employing an optimal grouping of targets, based on maximising modularity, each UAV is assigned to each subgroup. The centre of the circular flight paths are optimised by minimising the worst case scenario. UAV path transition from one path to another is formulated as a nonlinear optimisation problem and approximated as a finite horizon discrete optimisation problem. The algorithm is tested on random scenarios and it shows that over 160 targets among 200 targets are tracked simultaneously most of the operation time by only using three UAVs.

Collision avoidance, optimal communication scheduling among UAVs and commanding centres, designing a low-level controller for path following, and theoretical convergence properties of the algorithm are yet to be studied. Extending the current algorithm to a 3-dimensional problem including altitude optimisation would be straightforward. For assigning multiple UAVs on the same circular path, each UAV has to either communicate to the other UAVs in order to receive the position information or estimate it using a vision navigation approach as shown in [1, 2].

Acknowledgements

This is funded by an EPSRC under the grant EP/G036195/1.

References

[1] A. M. Hyslop, and J. S. Humbert, *Autonomous navigation in three-dimensional urban environments using wide-field integration of optic flow*, Journal of Guidance, Control, and Dynamics, Vol. 33, No. 1, pp. 147–159, January-February 2010.

[2] A. M. Fosbury, and J. L. Crassidis *Relative navigation of air vehicles*, Journal of Guidance, Control, and Dynamics, Vol. 31, No. 4, pp. 824-834, Jury-August 2008.

[3] B. R. Geiger, J. F. Horn, A. M. DeLullo, and L. N. Long, *Optimal path planning of UAVs using direct collocation with nonlinear programming*, AIAA-2006-6199, AIAA GN&C Conference, August 2006.

[4] J. R. Riehl, G. E. Collins, and J. P. Hespanha, *Cooperative graph-based model predictive search*, Proceeding of the 46th Conference on Decision and Control, December, 2007.

[5] J. Kim, and Y. Kim, *Moving target tracking in dense obstacle areas using UAVs*, the 17th IFAC World Congress, Seoul, Korea, July 6-11, 2008.

[6] J. Kim, and Y. Kim, *Aerial Vehicles: Chapter 17, Optimal circular flight of multiple UAVs for target tracking in urban areas*, InTech-online, 2009.

[7] L. N. Trefethen, and D. Bau III, *Numerical Linear Algebra*, SIAM, 1997.

[8] M. E. J. Newman, *Modularity and community structure in networks*, Proceedings of the National Academy of Sciences, Vol 103, No. 23, pp. 8577–8582, June 2006.

[9] M. Yvinec, *2D Triangulations*, In CGAL User and Reference Manual. CGAL Editorial Board, 3.6 edition, 2010.

[10] R. J. Dunn III, P. T. Bingham, and C. A. "Bert" Fowler, *Ground moving target indicator radar and transformation of U.S. warfighting*, Analysis Center Papers, Northrop Grumman, 2004.

[11] Roke Manor Research, *Visual Moving Target Indicator (VMTI)*, `www.roke.co.uk/sensing /vmti.html`.

[12] V. Shaferman, and T. Shima, *Unmanned Aerial Vehicle Cooperative Tracking on Moving Ground Target in Urban Environments*, Journal of Guidance, Control, and Dynamics, Vol. 31, No. 3, pp. 1360–1371, September-October 2008.