DOI 10.31256/UKRAS19.13

Exploiting System Capacity with a Distributed Routing Strategy for UAVs*

W. Bonnell, *Member, UKRAS*

Abstract — **This paper presents a routing strategy for UAVs that can be applied in conjunction with lower level navigation and collision avoidance methods. The strategy presented draws inspiration from traditional road vehicle routing where some cars are directed down less busy routes even if it results in a longer path. Our strategy will allow individual UAVs to route themselves in 2D space in order to avoid areas of high-density traffic.**

The proposed approach is then explored in simulation. Details of the simulation set-up are provided. The results demonstrate that the traffic is safer when the routing strategy is used compared with just a simple collision avoidance method.

I. Introduction

This paper demonstrates that by implementing higher level routing on top of simple navigation algorithms, it is possible to ensure Unmanned Aerial Vehicles (UAVs) more fully exploit the airspace capacity, providing system-wide benefits. Here we investigate the BeeJamA routing algorithm [1], previously applied to road traffic networks, represented as graphs where intersections are nodes and roads between them are links. At each node, cars score neighbouring nodes according to a weighted sum of their distance from the intended destination and a penalty related to traffic congestion. Simulations showed that by proceeding to the node with the lowest score, cars achieve higher levels of stable throughput than by other routing methods.

In contrast to cars, UAVs have a greater degree of freedom in their maneuverability. Although they can fly in three dimensions, the airspace in which they operate may be restricted to a 2D plane or several such layers. This is due to 1. restrictions on how close UAVs can fly to traditional airspace, 2. that some minimum vertical separation is required between UAVs and 3. that UAVs will have some minimum operating altitude. Therefore, building on the idea of using routing as a means to exploit system capacity, this paper explores using the BeeJamA algorithm to route UAVs through a 2D airspace.

Navigation through 2D spaces is a well studied problem and a number of methods have been developed for large multi-agent systems that may be applicable to UAVs. Examples include those based on the Social Force Model (SFM) [2] or using Reciprocal Velocity Obstacles (RVO) [3]. These paradigms incorporate robust collision avoidance. However, collision avoidance becomes more computationally expensive and/or less effective/efficient when agents are densely packed. For example, depending on the set-up of an RVO-based navigation strategy, the entire system can become gridlocked if agents are too close, see [3].

The routing strategy presented in this paper is intended to be implemented in conjunction with a navigation method, such as SFM or RVO, in order to ensure that extreme traffic densities do not arise. While these scenarios may seem unlikely to happen due to the inherent extra capacity afforded by relatively unrestricted 2D airspace, there are some obvious bottlenecks such as landing zones. It is also likely that in the near future there will be a significant increase in the density of UAV traffic, especially in and around large population centers. According to the Single European Sky ATM Research Programme (SESAR) in 2016 [4], there were an estimated 1–1.5 million leisure drones and 10,000 commercial drones in use in Europe. These numbers are expected to rise to near 7 million and 400,000 respectively by 2050, exacerbating existing pinch points.

A further problem with the roll-out of UAVs is the lack of an appropriate air traffic management infrastructure. Thus, distributed traffic management strategies are desirable. The routing method presented here is therefore implemented so that each UAV generates its route dynamically based on the environment. This way UAVs can be flown autonomously without the expense of developing bespoke communication and control infrastructure.

The rest of this paper is divided in to three sections. Section II will provide details on how the routing method is implemented along with some details of the underlying navigation mechanisms. Section III will then describe how this method has been implemented in simulation. Section IV will present the results from a representative traffic scenario for a variety of simulation parameters. Finally, Section V will provide some concluding remarks, including possible future work based on this paper.

II. Methods

While the main contribution of this paper is the routing method described below, this section will begin with a brief overview of the underlying navigation and collision avoidance

*This work was supported by an EPSRC grant EP/R0047571 and by Thales SA.

W. Bonnell is with the University of Bristol, Bristol, BS8 1QU, UK (phone: +44117 985 8759; e-mail: Wb17716@bristol.ac.uk).

scheme. The UAVs move using a method inspired by the SFM, comprised of two parts. The first part causes the UAV to move toward its goal, with an acceleration toward the goal that is proportional to the deviation away from a desired velocity in the direction of the goal. The second part is the collision avoidance. When a UAV comes within a certain range of another UAV or is on a collision course within a given time window, then the two UAVs will accelerate away from one another with a magnitude that is inversely proportional to their separation. These two component accelerations are then combined in such a way as to ensure that the UAV does not exceed a maximum acceleration (a physical bound).

The routing developed for this paper was inspired by [1]. The BeeJamA algorithm uses a graph representation of the road network to decide on which node, i.e. intersection, to move to next. In order to extend this to a 2D space, the UAV will generate candidate intermediary goals that we call waypoints, and then choose one to travel toward. This enables the routing method to be layered on top of any individual navigation regime as each waypoint becomes the goal of that particular section of the overall journey. Also, as each UAV generates and ranks its own waypoints, this allows the UAV to operate without centralised control.

The routing method works by first generating a set number of candidate waypoints at a distance R_{wp} away from the UAV, i.e. all the waypoints are on a circle of radius R_{wp}, see Fig. 1. Each candidate waypoint, c, is then assigned a dimensionless score Q_c according to

$$Q_c = N_c + (R_{cg} - R_{UAVg})\,\gamma, \tag{1}$$

where N_c is the number of other UAVs within a distance R_c of the candidate waypoint, R_{cg} and R_{UAVg} are the distances from the candidate waypoint and the UAV to the end goal respectively, and γ is a parameter with units m^{-1}. All distances in (1) have units m. The candidate with the smallest score Q_c

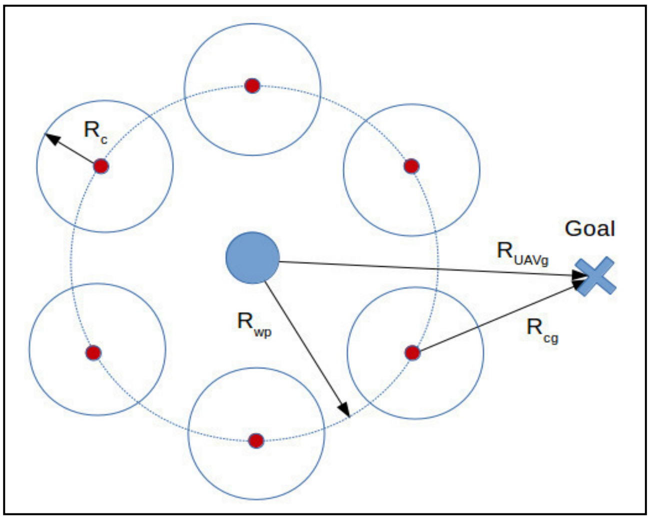

Figure 1. Shows a UAV (blue circle), candidate waypoints (red circle) and the UAV's goal. The radius at which candidate waypoints are generated R_{wp}, the radius out to which the candidate waypoints count other UAVs R_c, the distance from the UAV to its goal R_{uavg} and the distance from a particular candidate waypoint to the goal R_{cg}.

is then chosen as the next waypoint to move toward. Once the UAV reaches that waypoint it then generates new candidate waypoints and chooses again. Also, if the end goal is within R_{wp} of the UAV then it is also added to the list of candidate waypoints.

The waypoint that has the least traffic in its local vicinity and is closest to the end goal is therefore chosen according to (1). The parameter γ can be used to decide which of these two factors is more important. A UAV with a high γ will prioritise heading toward their end goal while a UAV with a small γ will prioritise avoiding areas with lots of other UAVs present. For this paper, γ is the same for all UAVs throughout a simulation run, but this assumption could be relaxed in future work.

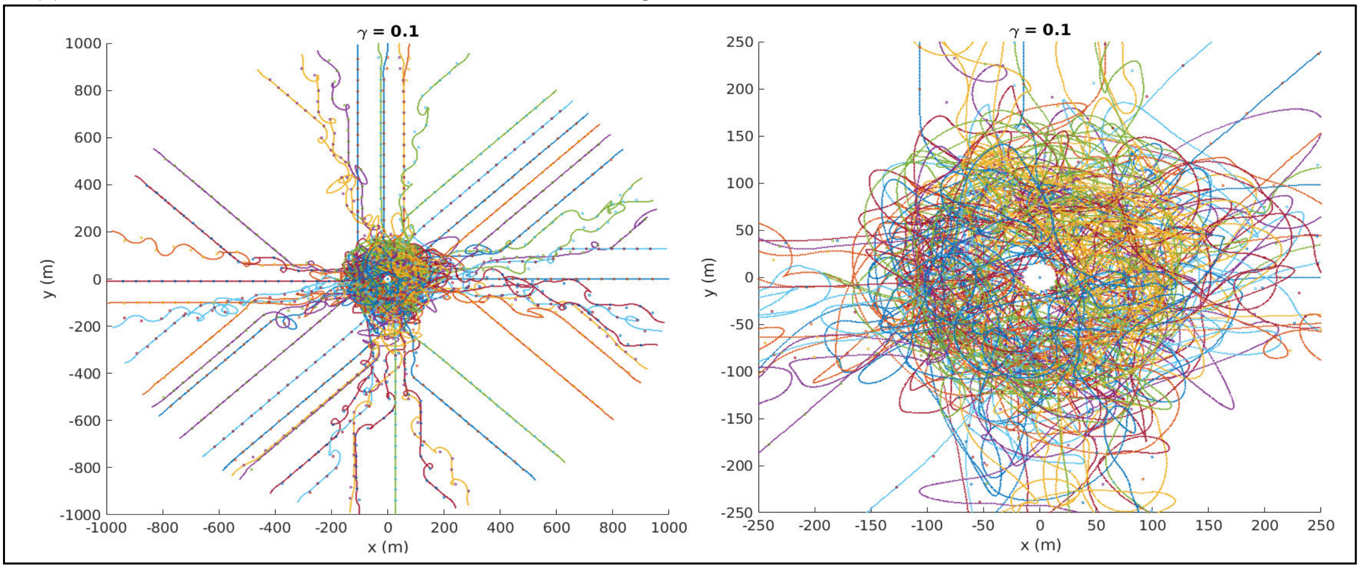

Figure 2. An example simulation run for 50 UAVs where γ is 0.1m^{-1}. Each line is the trajectory taken by a UAV and each point is a waypoint. On the right is a zoomed in view.

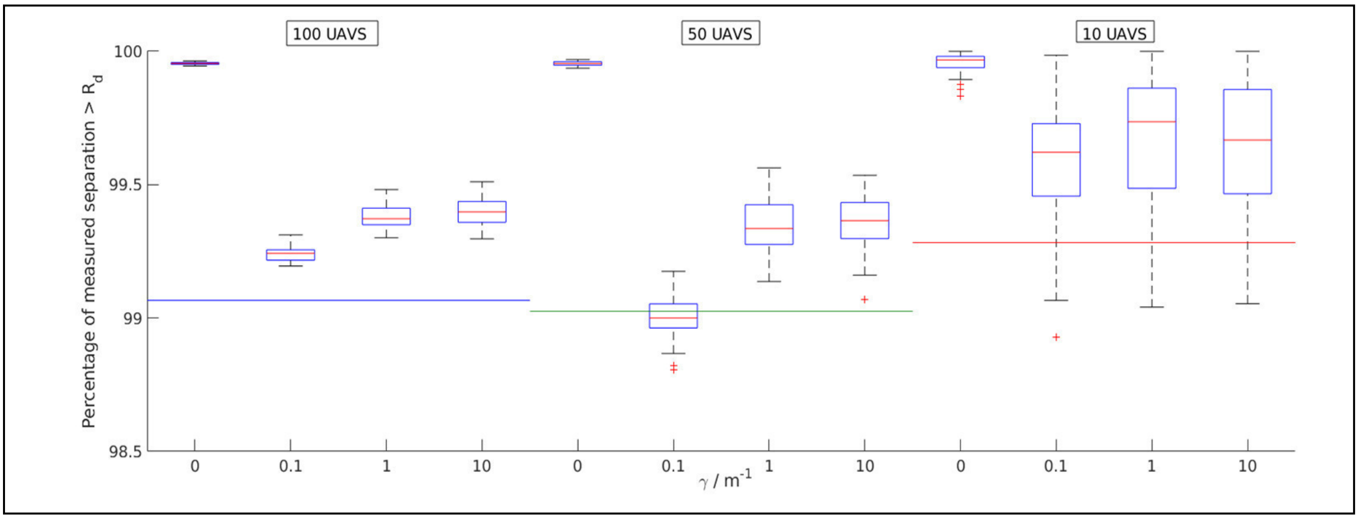

Figure 3. The percentage of UAV displacements that are greater than or equal to R_d for different values of γ and numbers of UAVs. The solid lines show the average percentage for that number of UAVs when there is no routing. The red line in each box shows the mean value.

III. Simulation Procedure

This section provides details about the simulation. In order to ensure that the routing method was tested in a scenario with high traffic density, all the UAVs share an end goal in the center of the simulation. Thus, the scenario is an approximation of a depot or similar landing area that all the UAVs are converging on.

The number of UAVs is set before the simulation starts and all UAVs are instantiated at the first time step at random points along the circumference of a circle with radius 1000m. Each UAV's starting velocity is $20ms^{-1}$ pointing toward the central goal. Two simplifications for this work are that the simulation is restricted to 2D and that UAVs are considered to have landed if they come within a set radius, 15m, of the end goal. This same criterion is used to determine when a UAV has reached its next waypoint and therefore when to generate the next one.

The UAVs simulated in this paper are also quadcopter-like as they can hover. As such there is a certain minimum desired separation that UAVs should try to maintain in order to not adversely affect one another, referred to in this paper as R_d. This defines the idea of a conflict, when two UAVs have a separation that is less than the desired minimum.

See Fig. 2 for example trajectories. Each simulation run is for 500s and statistical results are derived by averaging over 50 runs. Some of the UAVs exhibit an unexpected behaviour where they loop around waypoints. This was not intended and hopefully can be eliminated in future work through refinements to simulation parameters.

IV. Results

This section will provide results from simulation and explore the effect that the routing method has on the safety of the system. Using the concept of the conflict as described in the previous section, the percentage of recorded UAV separations that are greater than or equal to the desired minimum separation, R_d, provides a metric to understand the safety of each simulation. If the percentage is 100%, then no conflicts are recorded for that particular run of the simulation (perfect safety).

This metric is recorded in Fig. 3 for a variety of values of γ and number of UAVs. The solid line shows the average percentage for that number of UAVs when there is no routing. There are two main results to note. Firstly, in almost all scenarios the average percentage of UAV separations compliant with the minimum separation increases for all values of γ compared with the average for simulations where no routing is used. Thus, the system is safer when the routing method is implemented. Secondly, the system becomes safer as gamma increases from $0.1m^{-1}$ to $10m^{-1}$. This is the opposite of the expected behaviour as smaller values of γ should correspond to traffic that prioritises avoiding areas of higher density.

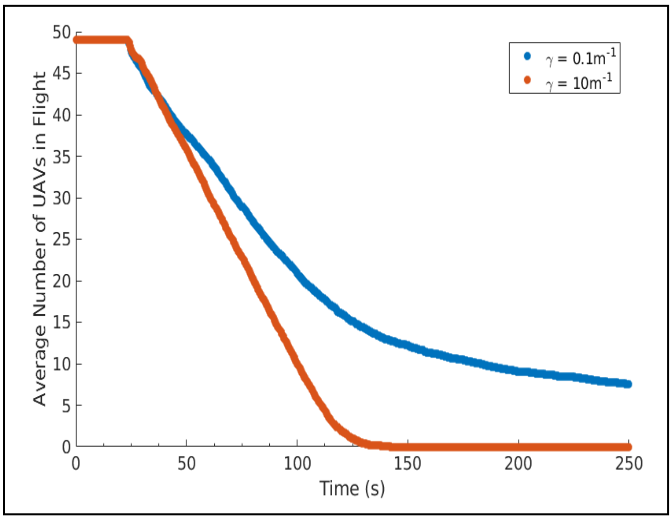

Figure 4. The number of UAVs still in flight versus time step for $\gamma = 10m^{-1}$ (orange) and $\gamma = 0.1m^{-1}$ (blue).

It is also worth considering how γ can affect other characteristics of the system. Fig.4 shows how the average number of UAVs still in flight changes over time. It shows that for a larger value of γ, the number of UAVs in flight decreases more rapidly in time, as traffic prioritises reaching the landing zone more strongly.

V. Conclusion

This paper has presented a new routing strategy for UAVs that attempts to improve performance by avoiding areas of high traffic density. The routing strategy has been implemented in a distributed manner so that it can be used on UAVs without centralised control or the development of new infrastructure. The results presented are for a simulation scenario inspired by a UAV depot or other landing zone in order to force UAVs to form an area of high density traffic. Despite this bottleneck, the safety of the system has been shown to increase when the routing strategy is applied on top of some simple navigation rules.

As was expected, the value of γ for which there were the fewest conflicts on average was $0m^{-1}$. This is when UAVs will always pick a waypoint based on which waypoint has the least neighbouring UAVs. However, the relationship between γ and the system safety did reveal an unexpected behaviour.

As γ increases from $0.1m^{-1}$ to $10m^{-1}$, the average safety of the system increases as well, see Fig. 3. Also, from Fig. 4, it can be seen how the rate of UAVs landing is higher for higher values of γ. Since no analysis of the severity of each conflict has been conducted, i.e. a conflict of 25m is not as severe as a conflict of 1m, it is possible that though the rate of conflicts is lower for larger γ, the severity of those conflicts is worse. In other words, for large values of γ, the UAVs converge quickly on the central goal and rapidly increase the local density. While this results in severe conflicts, they also manage to land quickly which in turn decreases the local density allowing other UAVs to move more safely.

There are several directions that future work might take. The strategy presented here is agnostic about what method the UAV uses to detect other UAVs and the landing mechanics are extremely simplified. Both aspects could be modelled more accurately in future iterations. As for the routing strategy itself, there are two adaptations that might be of interest to explore. These are a stochastic version where Q_c is used as a weight and a version where γ is dynamic. By introducing a stochastic element to (1), we obtain a more extreme version of the routing strategy by ensuring that some amount of the UAVs pick sub-optimal routes, ensuring a lower local density around the goal. Alternatively, a dynamic γ that is related to the total number of UAVs in the simulation could allow UAVs to change their risk appetite, by prioritising avoidance when there are many UAVs still flying, and prioritising landing when there are fewer.

With conflicts as defined in Section III, it is acceptable for a traffic system to have some conflicts provided they are short lived and are close to the desired minimum separation. However, this paper recognizes that actual collisions will need to be entirely prevented before any routing strategies can see real world aviation applications. Although we have shown how routing strategies can achieve performance gains over simple collision avoidance and navigation methods, our future aims are to establish how these gains might still be achieved, with the minimisation of conflicts and elimination of undesirable behaviours such as spiralling.

References

[1] H. F. Wedde et al., "A novel class of multi-agent algorithms for highly dynamic transport planning inspired by honey bee behavior," in *2007 IEEE Conference on Emerging Technologies and Factory Automation* (EFTA 2007), 2007, pp. 1157–1164.

[2] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, May 1995.

[3] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," *Springer Tracts Adv. Robot.*, vol. 70, no. STAR, pp. 3–19, 2011.

[4] "European Drones Outlook Study, Unlocking the value for Europe," *SESAR*, 2016.