CEAS-GNC-2024-016

EuroGNC Bristol, UK June 11th-13th 2024

Airspace Routing and Automation for Vertiports

Hirad Goudarzi Research Associate, Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1TR, UK. hirad.goudarzi@bristol.ac.uk

Mickey Li Research Associate, Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1TR, UK. mickey.li@bristol.ac.uk

Duncan Hine Technical Specialist, Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1TR, UK. duncan.hine@bristol.ac.uk

Thomas Richardson Professor of Aerial Robotics, Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1TR, UK. thomas.richardson@bristol.ac.uk

Arthur G. Richards Professor of Robotics and Control, Bristol Robotics Laboratory, Bristol, BS16 1QY, UK. arthur.richards@bristol.ac.uk

ABSTRACT

An approach is proposed for the design and partial automation of arrival and departure routing at vertiports. Multiple independent routes are provided to promote multiple simultaneous take-off and landing operations. The goal is to resolve conflicts spatially in the vertiport airspace, reducing conflicts in scheduling, en-route and ground-handling phases. A human-in-the-loop automation protocol inspired by railway signaling is introduced to prevent conflicting arrival and departure movements. For this, the concept of fixed 'routes' between fixed waypoints which require the reservation of intermediate waypoints or 'blocks' provide guaranteeable interlocking functionality. Two vertiport topologies are presented: a handcrafted example, and a scalable, parameterized vertiport based on counter-threaded helices. Both topologies are demonstrated in simulation and in reality with multiple drones where consistent spatial separation is shown.

Keywords: drone; vertiport; semi-automated; deconfliction

1 Introduction

Drones[1] are predicted to grow in commercial use in the coming years [1]. Vertical take-off and landing (VTOL) drones are especially popular for their ability to hover and their independence from runways, but scaling up will inevitably demand some ground infrastructure, especially at emerging hubs. This motivates renewed interest in VTOL airports or *vertiports* [2], especially as their availability is anticipated to be one of the top three constraints on future drone scalability [3]. Fig. 1 illustrates the proposed concept of operations. In particular, the landing and take-off paths are to be spatially separated such that paths with non-overlapping endpoints can be used simultaneously. For example, landings A1→H3, A2→H1 and A3→H2 could all happen independently. This independence has been identified as a strong enabler of capacity of the vertiport as a whole [4]. Vertiport research is varied, encompassing architecture, economics, design and regulations [2]. NASA's High Density Vertiplex (HDV) project [5]

[1]This paper will use the term "drone" for consistency and familiarity, noting that other terms such as "uncrewed air system," "aerial robot" and "remotely piloted air system" are also applicable and capture important aspects of the subject.

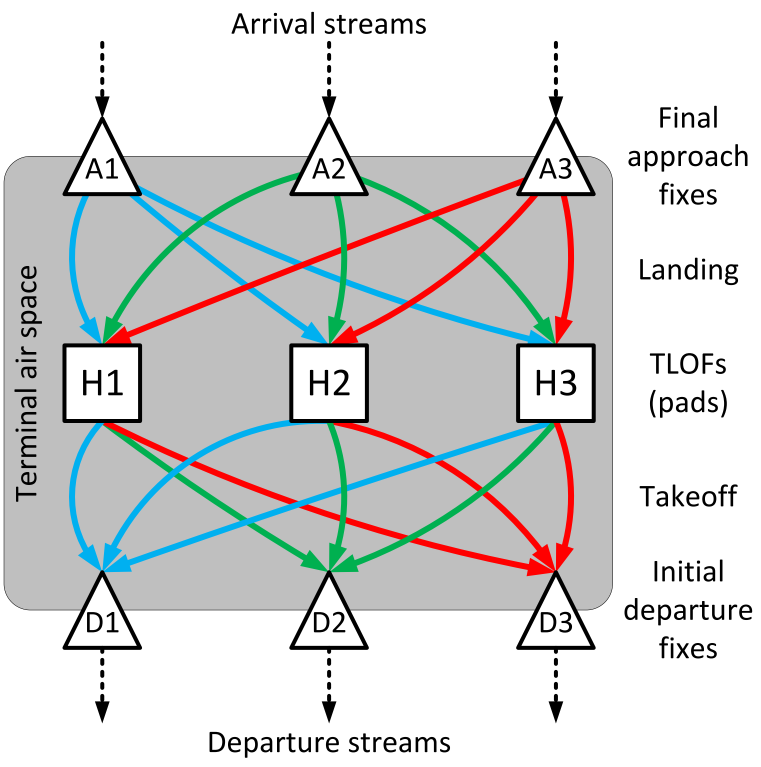


Fig. 1 Proposed vertiport concept of operations

has looked at architectures and human factors for vertiport operations, also including hardware-in-the-loop trials with surrogate drones[6]. EASA work in this area gives broad principles for design [7] building on established principles for heliport design [8]. In this context, dependent on scaling, our work here could be considered as complex routing within a take-off/climb volume; a combination of separate, complex take-off/climb volumes; or even a 3D assembly of hover taxi-routes between a virtual Final Approach and Take Off airspace region and associated Touchdown ad Lift Off (TLOF) points on the ground. However, it must be noted that none of these would meet the draft requirements [7] and thus look ahead to a far future with much higher automation. Note that this paper does not consider transitions between hover and forward flight and only looks at (near) vertical flight.

Closest to this paper are works on designing the flight paths for final descent and initial climb. The *vertidrome* concept [9] includes detailed design of descent and landing profiles for multiple gates sharing one landing TLOF and one take-off TLOF between many parallel gates. Ref. [10] adds concentric rings for holding traffic around a central vertiport control zone. Meanwhile, Ref. [4] shows that independent simultaneous landings and take-offs could dramatically improve vertiport performance – but without designing the exact paths needed. The contribution of our work is a proposal for detailed design of these paths and the associated control system. We envisage considerably more TLOFs than most published works and focus on designing independently usable paths.

2 Principles of Operation

This section outlines the airspace principles upon which the proposed vertiport would operate, including the design of the paths and the associated control logic. The vertiport has a dedicated terminal airspace [9] into which drones enter via predefined initial approach fixes [4] from the surrounding en-route airspace, which is assumed to be supported by a separate traffic management system to ensure separated arrivals. Such an arrangement can readily fit into the EASA U-space framework [11], with drone-only geozones for terminal airspace and predefined routes to them. On the ground, a number of Touchdown and Lift Off (TLOF) areas [8] denote the physical points on which drones can land. Between arrival fix and TLOF, aircraft fly along fixed, pre-designed paths, much like the published procedure maps for airports. On departure, drones again fly fixed paths from TLOFs to departure fixes, where they depart the terminal airspace. Fig. 1 summarizes the operation in conceptual form. Unlike many other works, a large number of TLOFs is assumed. The design of the paths aims to maximize spatial separation, *i.e.* paths joining different fixes and TLOFs do not overlap at all and can be used independently without real-time temporal coordination between different drones.

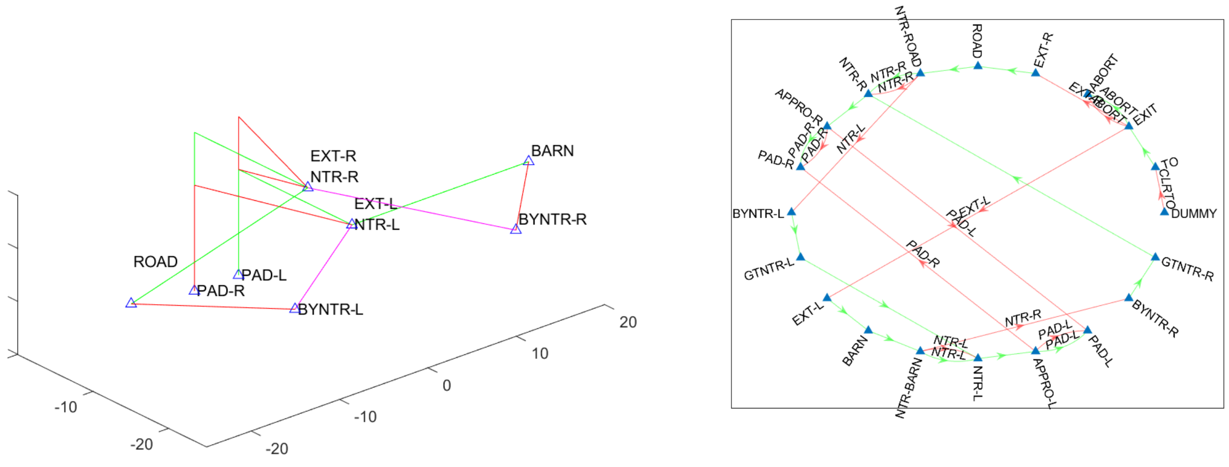
Drones navigate along the paths and may switch between them at crossing points according to (sparse) instructions from the ground, inspired by the air traffic technique of point merge [12, 13]. The control problem is then to avoid issuing conflicting instructions, to which the solution is inspired by railway signalling [14]: commands are restricted such that only one drone can be in a section of airspace at a time. This also draws on the ideas of *procedural control* for air traffic [15], relevant here because extensive surveillance and precise 4D navigation is unlikely to be available in many drone scenarios. The scope of this paper is limited to discussing the interlocking between such commands. These interlocks can form the guards to a human controller in a semi-automated solution, as in our flight trials, or would form the scheduling constraints on a future fully-automated arrival and departure manager [16].

The authors accept that this approach seems antiquated given the emergence drones fully capable of designing trajectories in real-time [17], but it does have a clear route to verification [18] and inherits hard-won experience of safety-critical operation [14]. However, it must be noted that the paths proposed are considerably more intricate than draft regulations [7] foresee and are thus suited to far future operations with a high level of automation.

2.1 Design of Takeoff and Landing Paths

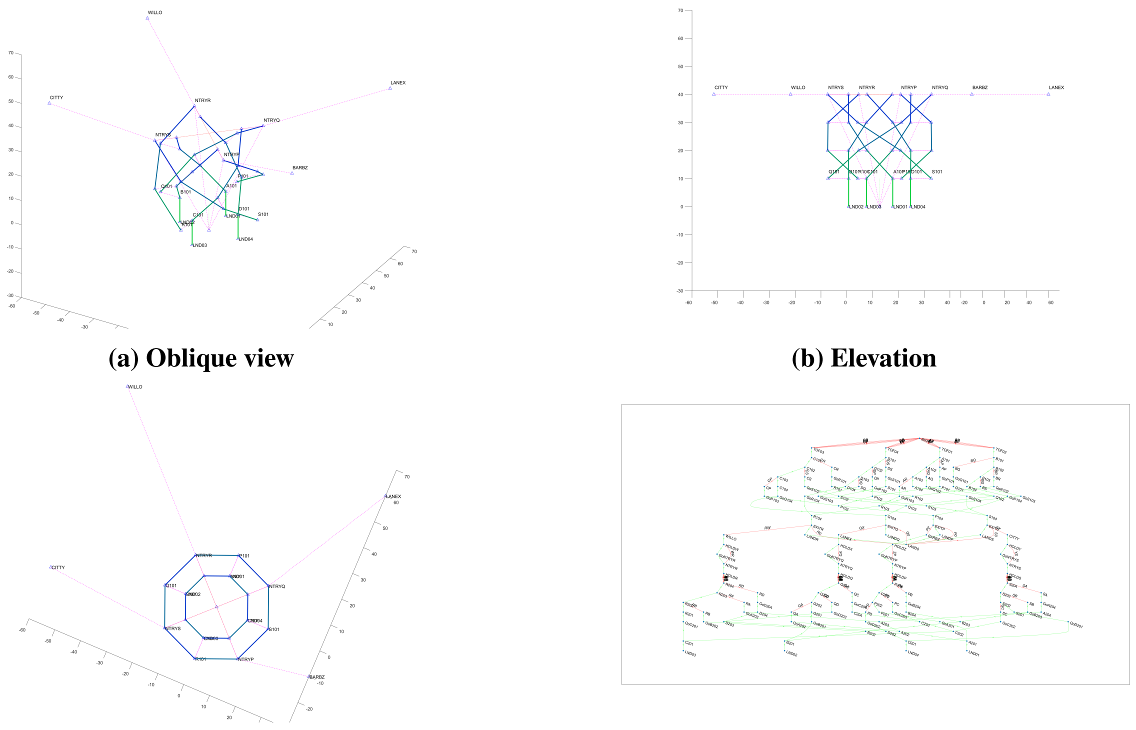
The key requirement for the design of the paths is that any two paths with non-overlapping endpoints must be safely separated. This is the maximal separation available: paths with overlapping endpoints would conflict at those endpoints. The distance of separation will depend on drone size, navigation accuracy, control performance, terrain, wake considerations and wind conditions [4, 7]. Detailed incorporation of this important parameter is left for future work. Two families of designs have been considered:

2.1.1 *Hand-crafted example*



(a) Spatial **(b) Topological**

Fig. 2 Views of the hand-crafted "T" example pattern



(a) Oblique view **(b) Elevation**

(c) Plan **(d) Topological**

Fig. 3 Counter-threaded helical paths

Fig. 2a shows a handcrafted pattern for two TLOFs, PAD-L and PAD-R, and two collocated arrival and departure fixes, *e.g.* “NTR-L" denotes the fix for arrival or entry from the left. By separating the arrival/departure fixes in altitude, since the PAD-L→EXT-R path passes over the opposing PAD-R→EXT-L path. Hence the design goal of independent paths between non-overlapping endpoints is met. The waypoints “ROAD" and “BARN" are token destination/origin locations named for familiarity based on features of our test site. Waypoints “BYNTR-x" provide further separation in the en-route space between crossing routes to different arrival fixes. Fig. 2b will be discussed further in Section 2.2.

2.1.2 Counter-threaded helices

Scaling up demands a more systematic approach than hand-crafting, so Fig. 3 shows a concept for four pads based on a symmetric pattern of helical paths. Each pad is at the base of one of the inner four helices. Three different views of the pattern in 3D space are shown, accompanied by the topological representation of the pattern as a directed graph. A departing drone is placed at one of the four pads at the bottom of the pattern, marked ‘LND01’–‘LND04’. Takeoff occurs when the drone is commanded to a specified departure fix ‘NTRYP’ – ‘NTRYS’. It climbs straight up 10m to the bottom of an associated inner helix, ‘LND01’ to spiral ‘A’ at waypoint ‘A101’, ‘LND02’ to ‘B101’, and so on. At some point, the drone arrives opposite an outer helix, denoted ‘P’ to ‘S’, associated with its departure fix, and moves to the outer ring, climbing then up the outer helix to its departure fix. It then moves to the outer waypoints, ‘CITTY’, ‘WILLO’ etc (named for memorability in relation to features of the test site). To return and land, the drone does the reverse, descending the outer spiral ‘P’–‘S’ from its arrival fix ‘NTRYP’–‘NTRYS’, and then cutting in to the inner spiral leading to its assigned pad. Crucially, paths between non-overlapping pairs are not in conflict, i.e. the route from A to Q is separated from the route from B to S. Four drones can move through the pattern simultaneously, provided no two share a pad or fix. Fig. 3d illustrates the available paths through the pattern, from a common “entry" point at the top, to an assigned takeoff pad (‘TOF01’ etc), then to one of the four fixes (‘EXITP’ etc) to a corresponding outer waypoint (‘WILLO’ etc), and then back through an arrival fix (‘NTRYP’ etc) and down to a pad

('LND01' etc.). Note that not all these points are labeled in Figs. 3a–3c to avoid overlapping labels. For example, 'EXITP' and 'NTRYP' are in the same physical location but are different logical commands in the mission. The implementation of the routing logic is described further in Section 2.2. The graph representation is generated automatically from the mission description and provides a route to verification of the pattern properties.

Variations with helices are available: consider Fig. 3 instead for two pads, with outer top fixes alternating between arrivals and departures. Different variants of the pattern offer different trades between key factors: the vertical and horizontal extent of the vertiport airspace, the throughput of the vertiport, and the control required.

2.1.3 Free design

A future alternative for the design of the paths is to use a numerical trajectory generation approach to optimize the pattern. A centralized optimization of all paths would demand intensive computation, but one simplifying approach is to assume symmetry, such that the set of paths are constructed by rotating a common path around a vertical axis by a set of uniformly distributed angle θ. If that generating path is constrained to be spatially separated from its own rotation by θ, then rotations by any integer multiple $k\theta \in [0, 2\pi)$ must be separated from all the others, *i.e.* for the tree $\mathbf{p}$ with nodes indexed by $k \in [0, \ldots, N]$:

$$\|\mathbf{p}(\mathbf{k}) - \mathbf{R}_\theta \mathbf{p}(\mathbf{k}')\| \geq D_{\min} \ \forall (k, k') \in [0, \ldots, N] \times [0, \ldots, N] \tag{1}$$

where $\mathbf{R}_\theta$ is the 3×3 matrix rotating by θ about the z-axis and $D_{\min}$ is the minimum separation. Fig. 4 shows some initial results from this method using a rapidly-exploring random tree (RRT) [19] as the generator for eight paths separated by 45^o. The RRT is not ideal for flyable paths given the quality of the paths generated [20] but is simple to implement and constrain. Future work will look at optimizers for smoother, flyable paths in this role. The tree was grown from a starting point at $(20, 0, 0)$ and constrained to remain within the cylinder of radius 30 from the z-axis. Points on the cylinder surface are possible arrival or departure fixes marked by triangles. The resulting paths connect each TLOF to fixes well-spaced around the cylinder offering routes to all directions, which respecting $D_{\min} = 10$ separation throughout.

2.2 Drone Control Logic

Following the principles from Ref. [18], drones are pre-loaded with missions that include all possible routes through the terminal air space, akin to an ARINC 424 database [21]. The choice of the route taken is controlled by advancing the mission waypoint index to select only the desired movements. Waypoint advances are under the control of a discrete set of *routes* issued from a central controller. In particular, the mission is augmented with a list of commanded jumps, $\mathcal{J} := \{s, r, t\}_i$ such that if a drone has been directed along route r_i and reaches source waypoint index s_i, then the drone jumps ahead to target waypoint index t_i. The design of the commands is tightly coupled to the design of the paths, as well as additional considerations to ensure separation.

Figs. 2b and 3d shows a representation of the paths in Figs. 2a and 3, respectively, as directed graphs. Green lines denote automated transitions, either through simple progression from one waypoint to the next, or "GOTO" commands in the mission. Red lines denote the jumps associated with a route, and in this case, the commands simply use the waypoint nicknames.

In Fig. 2b, the initial "DUMMY" waypoint at the right is the starting point. The "CLRTO" route enables a jump from "DUMMY" to "TO" which initiates the take-off, followed in default by "EXIT". As a safety feature, the default next behavior is to land again via waypoint "ABORT". Alternatively, route "EXT-L" will direct the drone to proceed to waypoint "EXT-L" from where it heads to destination

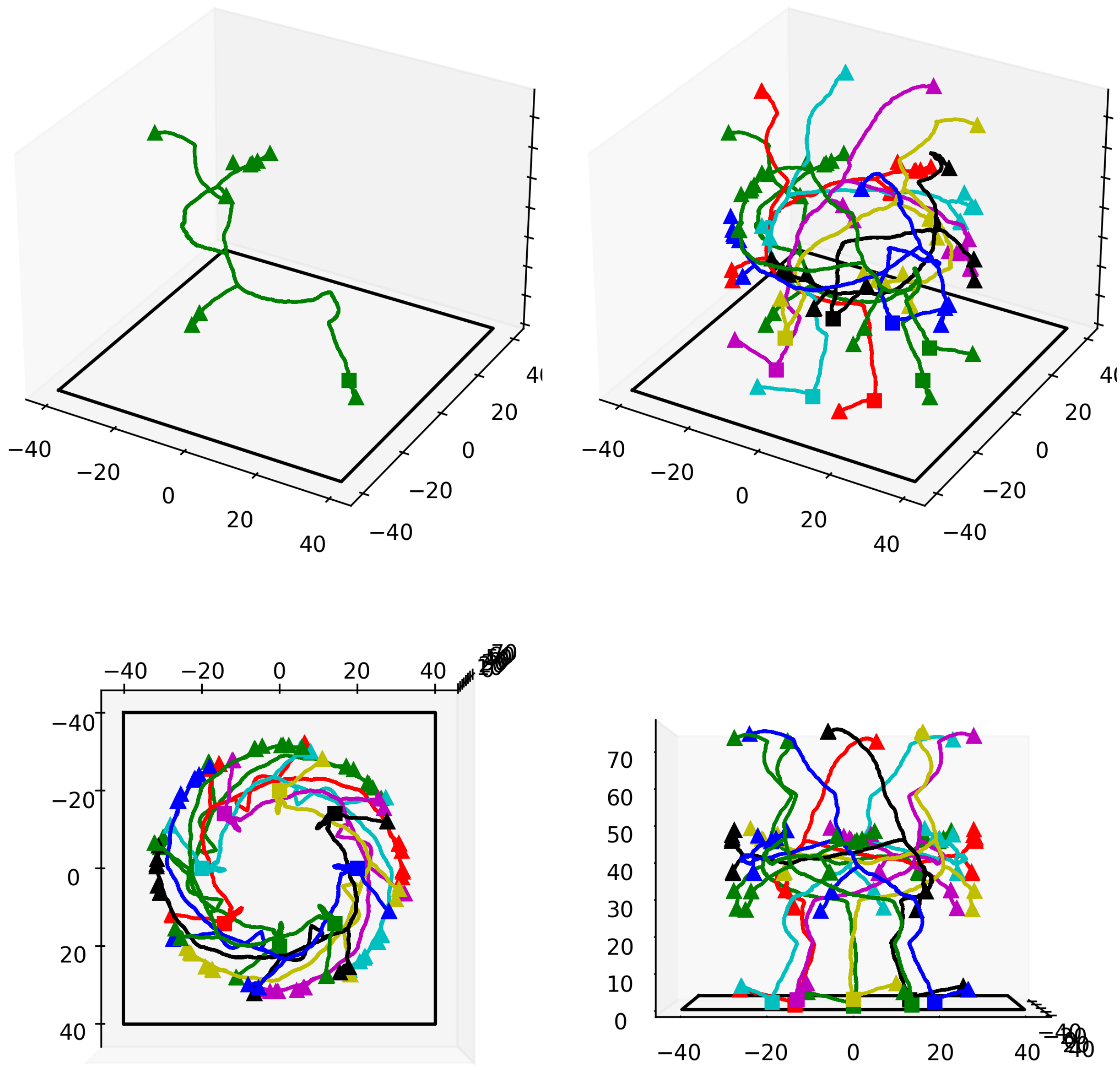


Fig. 4 Example pattern designed by RRT with symmetric deconfliction for eight independent paths. Pads are marked by squares. Triangles indicate arrival or departure fixes.

waypoint “BARN" , or route “EXT-R" will direct it to the waypoint of that name and hence to destination “ROAD".

Fig. 3d also starts at the top with a dummy initial waypoint. For brevity, a summary of the available routes is provided in Table 1. Drones can be routed from any pad to any fix, and vice versa, and then for testing, always to the nearest “outside world" waypoint ‘WILLO’, ‘CITTY’ etc.

The graph representation of the pattern permits some useful analysis using off-the-shelf tools. The absence of cycles ensures that the mission will complete at some point, rather than get stuck in a loop. Further, an all-pairs graph search for shortest paths can verify (i) that all the desired paths exist, *e.g.* that every TLOF (pad) can be reached from every arrival fix, and (ii) that the system is passively safe, meaning that in the event of a loss of communication that prevents commands being issued or enacted, drones will slowly progress towards safe landing points.

Routes	Summary	Effects
AP, AQ, AR, AS, BP, BQ, BR, BS, CP, CQ, CR, CS, DP, DQ, DR, DS	Departure routes from each pad A–D to each fix P–Q	Jumps from initial dummy waypoint to corresponding takeoff, then up the inner spiral. Jumps across to outer spiral connected to departure fix.
RW. QX, PZ, SY	Routes outside the pattern to “outside world" waypoints. Only routes to nearest waypoint provided.	Jumps from departure fix waypoints ‘EXITP’ etc to corresponding outside world waypoint.
WR, XQ, ZP, YS	Reverse of the above	Jumps from outside world points to nearest entry waypoints, ‘NTRYP’ etc.
PA, PB, PC, PD, QA, QB, QC, QD, RA, RB, RC, RD, SA, SB, SC, SD	Arrival routes from each fix P–S to each pad A–D. Reverse of top row.	Jumps from Arrival waypoints ‘NTRYP’ etc to tops of outer spirals ‘P204’, ‘Q204’ etc. Also jump to corresponding inner spirals when they cross.

Table 1 Summary of All Possible Routes for Four-pad, Four-fix Helical Pattern

For multiple drones, separation demands some interlocking between issuing the commands. For this, we propose an approach inspired by the *block system* for railway signaling, over a century old but established as safety critical and still the basis for much rail safety today [14]. The paths (like the rail tracks) are divided into non-overlapping *blocks* with the goal of ensuring that only one drone can be in a block at any given time. The routes are issued (like the signals are cleared) permitting a drone to enter a block only when that block is known to be clear, and cannot be re-issued until the block has been vacated. This functionality is typically enacted by means of a token or mutex, some device in hardware or software that can only be ‘possessed’ by one user at a time. The blocks prevent conflicts are overlapping waypoints, for example, the collocated ‘NTRYP’ and ‘EXITP’ waypoints in Fig. 3d are in the same ‘PP’ block, ensuring that no two drones can be routed through them simultaneously.

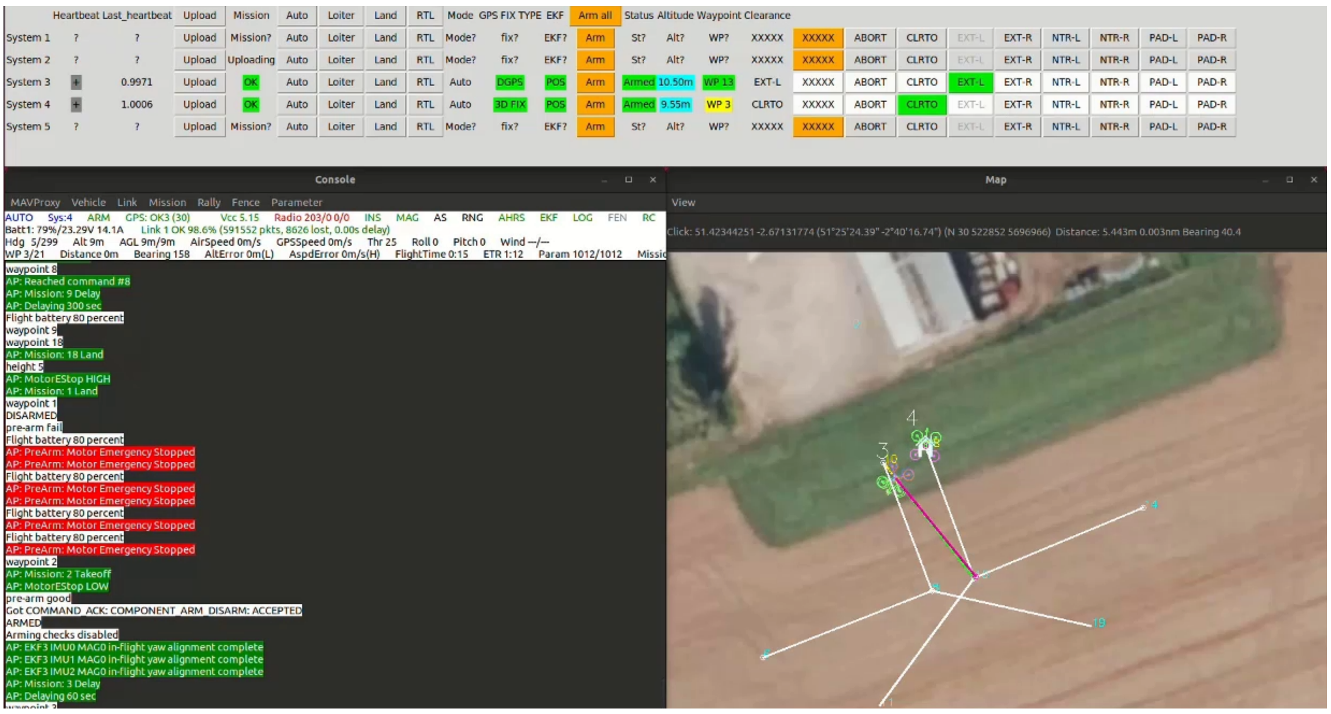
There is flexibility in how routes and blocks are allocated, even across the same pattern geometry. In this initial study, each inner spiral and each outer spiral is its own block. In extremes, each waypoint could be its own block, and each level could have its own routes, to enable drones to be ‘stacked’ in the pattern. The authors expect there to be trade-offs between complexity and performance, measured in terms of throughput or capacity.

In this work, the blocks are arranged such that each arm of the inner or outer helix froms its own block. This forms a set of Blocks AA, BB, CC and DD with respect to pads ‘A’ - ‘D’, and similarly for fixes ‘P’ - ‘S’ and Unmanned Aircraft System Traffic Management (UTM) fixes ‘Q’ - ‘Z’. Crucially if a drone wishes to take off from ‘A’ to go to departure fix ‘P’ on route ‘AP’, both the ‘AA’ and ‘PP’ blocks must be reserved by the drone, and for the drone to stay put otherwise.

3 Results

The tools described have been implemented in a suite of software. A design toolbox for developing patterns including commanded jumps and graph interpretation has been written in Matlab, including the facility to export the mission in MAVLINK format [22]. For flight testing, a control application has been written in Python using the Pymavlink drone communication library [23] and combining the jump controls, command logic, and a graphical user interface for multiple drone monitoring and command.

3.1 Flight Test of T-Shaped Pattern



	Heartbeat	Last_heartbeat	Upload	Mission	Auto	Loiter	Land	RTL	Mode	GPS FIX TYPE	EKF	Arm all	Status	Altitude	Waypoint	Clearance									
System 1	?	?	Upload	Mission?	Auto	Loiter	Land	RTL	Mode?	fix?	EKF?	Arm	St?	Alt?	WP?	XXXXX	XXXXX	ABORT	CLRTO	EXT-L	EXT-R	NTR-L	NTR-R	PAD-L	PAD-R
System 2	?	?	Upload	Uploading	Auto	Loiter	Land	RTL	Mode?	fix?	EKF?	Arm	St?	Alt?	WP?	XXXXX	XXXXX	ABORT	CLRTO	EXT-L	EXT-R	NTR-L	NTR-R	PAD-L	PAD-R
System 3	+	0.9971	Upload	OK	Auto	Loiter	Land	RTL	Auto	DGPS	POS	Arm	Armed	10.50m	WP 13	EXT-L	XXXXX	ABORT	CLRTO	EXT-L	EXT-R	NTR-L	NTR-R	PAD-L	PAD-R
System 4	+	1.0006	Upload	OK	Auto	Loiter	Land	RTL	Auto	3D FIX	POS	Arm	Armed	9.55m	WP 3	CLRTO	XXXXX	ABORT	CLRTO	EXT-L	EXT-R	NTR-L	NTR-R	PAD-L	PAD-R
System 5	?	?	Upload	Mission?	Auto	Loiter	Land	RTL	Mode?	fix?	EKF?	Arm	St?	Alt?	WP?	XXXXX	XXXXX	ABORT	CLRTO	EXT-L	EXT-R	NTR-L	NTR-R	PAD-L	PAD-R

```
AUTO    Sys:4    ARM    GPS: OK3 (30)        Vcc 5.15    Radio 203/0 0/0    INS    MAG    AS    RNG    AHRS    EKF    LOG    FEN    RC
Batt1: 79%/23.29V 14.1A      Link 1 OK 98.6% (591552 pkts, 8626 lost, 0.00s delay)
Hdg  5/299    Alt 9m    AGL 9m/9m    AirSpeed 0m/s    GPSSpeed 0m/s    Thr 25    Roll 0    Pitch 0    Wind ---/---
WP 3/21    Distance 0m    Bearing 158    AltError 0m(L)    AspdError 0m/s(H)    FlightTime 0:15    ETR 1:12    Param 1012/1012    Missic
waypoint 8
AP: Reached command #8
AP: Mission: 9 Delay
AP: Delaying 300 sec
Flight battery 80 percent
waypoint 9
waypoint 18
AP: Mission: 18 Land
height 5
AP: MotorEStop HIGH
AP: Mission: 1 Land
waypoint 1
DISARMED
pre-arm fail
Flight battery 80 percent
AP: PreArm: Motor Emergency Stopped
AP: PreArm: Motor Emergency Stopped
Flight battery 80 percent
AP: PreArm: Motor Emergency Stopped
AP: PreArm: Motor Emergency Stopped
Flight battery 80 percent
AP: PreArm: Motor Emergency Stopped
Flight battery 80 percent
AP: PreArm: Motor Emergency Stopped
waypoint 2
AP: Mission: 2 Takeoff
AP: MotorEStop LOW
pre-arm good
Got COMMAND_ACK: COMPONENT_ARM_DISARM: ACCEPTED
ARMED
Arming checks disabled
AP: EKF3 IMU0 MAG0 in-flight yaw alignment complete
AP: EKF3 IMU1 MAG0 in-flight yaw alignment complete
AP: EKF3 IMU2 MAG0 in-flight yaw alignment complete
AP: Mission: 3 Delay
AP: Delaying 60 sec
```

```
Click: 51.42344251 -2.67131774 (51°25'24.39" -2°40'16.74") (N 30 522852 5696966)  Distance: 5.443m 0.003nm Bearing 40.4
```

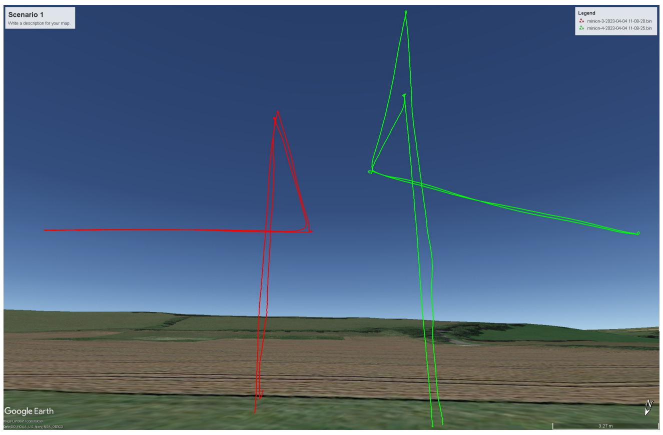
Fig. 5 Screen capture of semi-automated control system during flight test

Fig. 5 shows a screen capture from the ground control station during a flight test of the basic "T-shaped" pattern described in Section 2.1.1. The upper window shows two drones, with IDs 3 and 4, under the control of the system. The matrix of buttons at the right of this window enables the controller to issue commands to the drones. Observe that drone 3 has been issued with the "EXT-L" command, denoted by the green coloring of the associated button. Following the "endpoint as token", the other buttons in that column are disabled. This denotes that the "EXT-L" token is locked and no other drone can be given the "EXT-L" command until drone 3 has been issued and responded to another command.

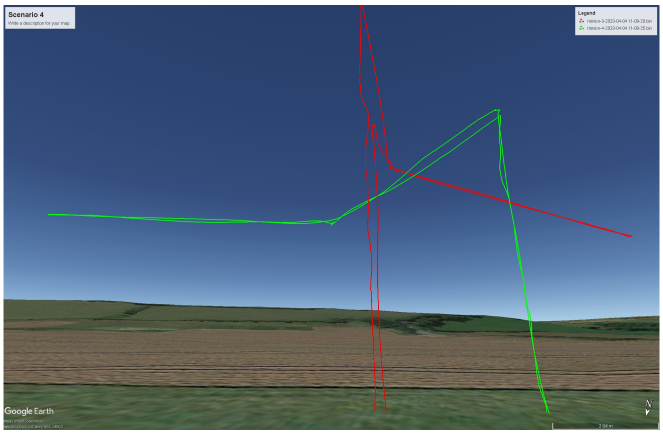
Fig. 6 visualize trajectories from three scenarios within these flight tests, extracted from the log files of the drones. The first scenario in Fig. 6a shows obvious spatial separation as the two drones go in opposite directions. The second scenario in Fig. 6b shows the drones crossing, drone 3 going from left pad to right fix, and drone 4 *vice-versa*. Drone 4 passed well under drone 3's path. Finally, Fig. 6c shows a case where the drones swap pads, but again were well-separated at the point of crossing. All of these scenarios were conducted purely under the control of the control logic described in Section 2.2, with commands issued by button pushes on the ground station. Safety pilots were present for both drones but made no interventions during the flights.

3.2 Multi-drone Flight Test of Helical Pattern

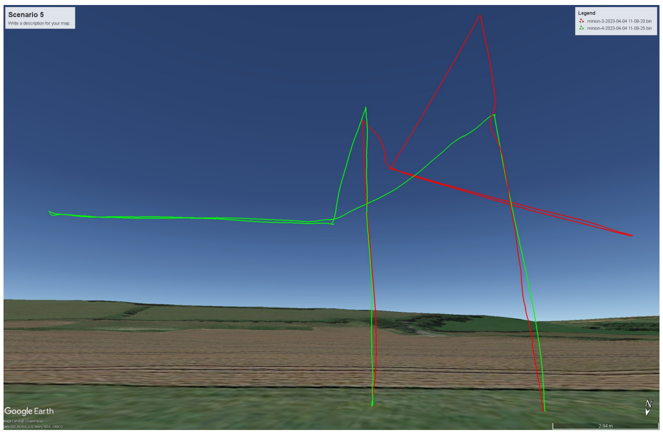
The introduction of the routing and block system during the design of the helical pattern, shown in section 2.1.2, allowed for flight tests with greater number of vehicles in a safer manner. A flight test of 4 drones on a 4-armed helical pattern was performed at Snowdonia Aerospace Center with one of the resulting trajectories shown in Fig. 8. A flight consists of a takeoff from a landing pad to a departure fix, a transition to a UTM fix, a transition back to an arrival fix, and then a landing pad request. In each of these stages, the drone awaits the reservation of its requested blocks through the users route request. All runs were first verified in Software-In-The-Loop simulation to ensure all the safety-critical parameters were maintained before flight testing in reality.



(a) **Drone 3 (red): PAD-L→EXT-L→NTR-L→PAD-L Drone 4 (green): PAD-R→EXT-R→NTR-R→PAD-R**

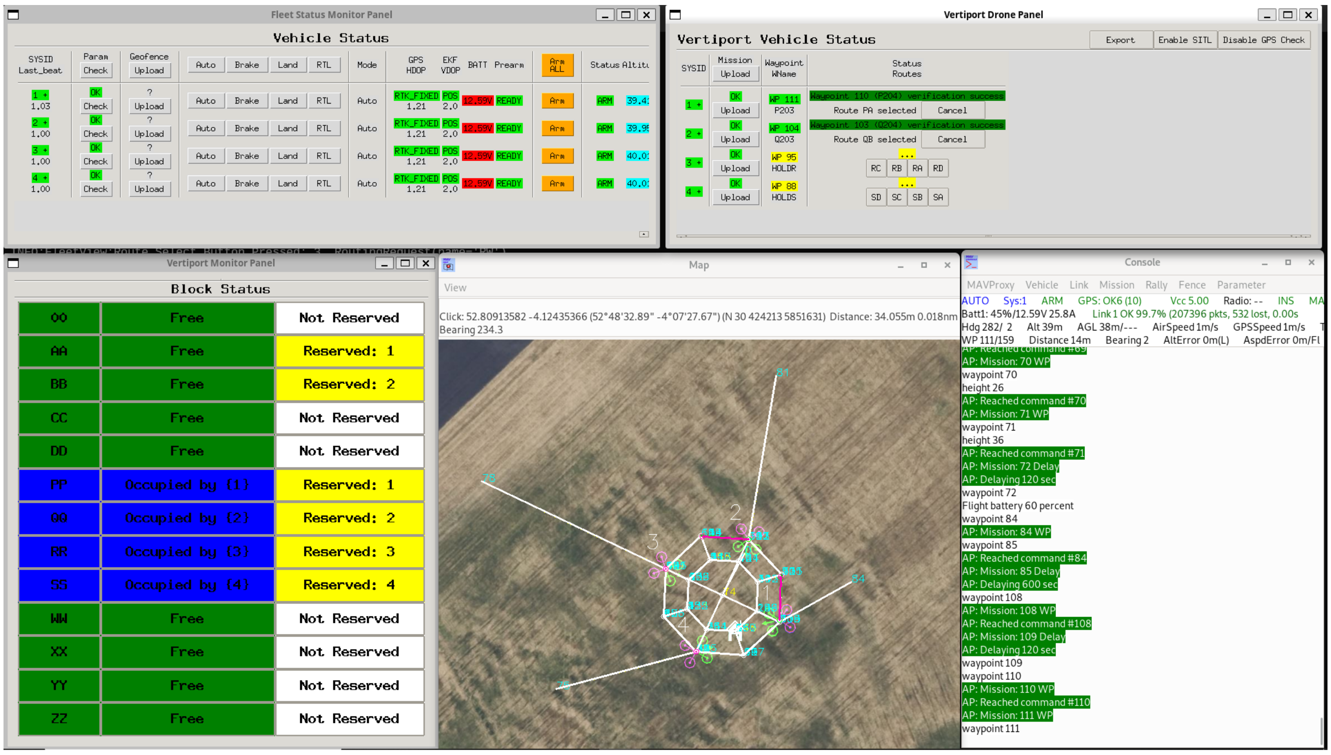


(b) **Drone 3 (red): PAD-L→EXT-R→NTR-R→PAD-L Drone 4 (green): PAD-R→EXT-L→NTR-L→PAD-R**



(c) **Drone 3 (red): PAD-L→EXT-R→NTR-R→PAD-R Drone 4 (green): PAD-R→EXT-L→NTR-L→PAD-L**

Fig. 6 Flight test results for three scenarios involving two drones, two pads and two collocated arrival/departure fixes.



Fleet Status Monitor Panel

Vehicle Status

SYSID Last_beat	Param Check	Geofence Upload	Auto	Brake	Land	RTL	Mode	GPS HDOP	EKF VDOP	BATT	Prearm	Arm ALL	Status	Altitu
1 + 1.03	OK Check	? Upload	Auto	Brake	Land	RTL	Auto	RTK_FIXED 1.21	POS 2.0	12.59V	READY	Arm	ARM	39.4
2 + 1.00	OK Check	? Upload	Auto	Brake	Land	RTL	Auto	RTK_FIXED 1.21	POS 2.0	12.59V	READY	Arm	ARM	39.9
3 + 1.00	OK Check	? Upload	Auto	Brake	Land	RTL	Auto	RTK_FIXED 1.21	POS 2.0	12.59V	READY	Arm	ARM	40.0
4 + 1.00	OK Check	? Upload	Auto	Brake	Land	RTL	Auto	RTK_FIXED 1.21	POS 2.0	12.59V	READY	Arm	ARM	40.0

Vertiport Drone Panel

Vertiport Vehicle Status

Export | Enable SITL | Disable GPS Check

SYSID	Mission Upload	Waypoint WName	Status Routes
1 +	OK Upload	WP 111 P203	Waypoint 110 (P204) verification success; Route PA selected; Cancel
2 +	OK Upload	WP 104 Q203	Waypoint 103 (Q204) verification success; Route QB selected; Cancel
3 +	OK Upload	WP 95 HOLDR	...; RC RB RA RD
4 +	OK Upload	WP 88 HOLDS	...; SD SC SB SA

Vertiport Monitor Panel

Block Status

OO	Free	Not Reserved
AA	Free	Reserved: 1
BB	Free	Reserved: 2
CC	Free	Not Reserved
DD	Free	Not Reserved
PP	Occupied by (1)	Reserved: 1
QQ	Occupied by (2)	Reserved: 2
RR	Occupied by (3)	Reserved: 3
SS	Occupied by (4)	Reserved: 4
WW	Free	Not Reserved
XX	Free	Not Reserved
YY	Free	Not Reserved
ZZ	Free	Not Reserved

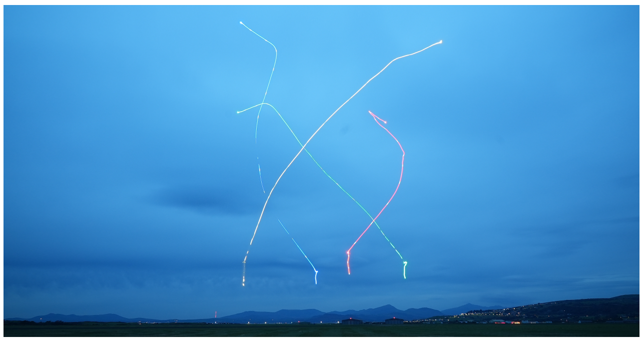
Map

```
Click: 52.80913582 -4.12435366 (52°48'32.89" -4°07'27.67") (N 30 424213 5851631) Distance: 34.055m 0.018nm
Bearing 234.3
```

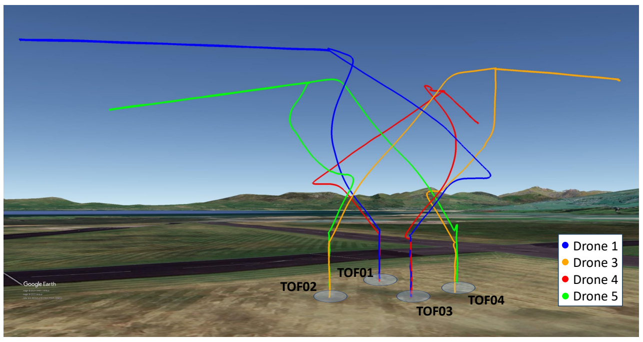
Console

```
AUTO    Sys:1    ARM    GPS: OK6 (10)       Vcc 5.00    Radio: --     INS    MA
Batt1: 45%/12.59V 25.8A     Link 1 OK 99.7% (207396 pkts, 532 lost, 0.00s
Hdg 282/ 2    Alt 39m    AGL 38m/---    AirSpeed 1m/s    GPSSpeed 1m/s    T
WP 111/159    Distance 14m    Bearing 2    AltError 0m(L)    AspdError 0m/Fl
AP: Reached command #69
AP: Mission: 70 WP
waypoint 70
height 26
AP: Reached command #70
AP: Mission: 71 WP
waypoint 71
height 36
AP: Reached command #71
AP: Mission: 72 Delay
AP: Delaying 120 sec
waypoint 72
Flight battery 60 percent
waypoint 84
AP: Mission: 84 WP
waypoint 85
AP: Reached command #84
AP: Mission: 85 Delay
AP: Delaying 600 sec
waypoint 108
AP: Mission: 108 WP
AP: Reached command #108
AP: Mission: 109 Delay
AP: Delaying 120 sec
waypoint 109
waypoint 110
AP: Mission: 110 WP
AP: Reached command #110
AP: Mission: 111 WP
waypoint 111
```

Fig. 7 Screen Capture of the semi-automated control system for the helix pattern during operation.



(a) **A long-exposure shot showing the simultaneous departure of 4 drones from the vertiport.**



(b) **Plotted drone position logs showing their combined departure and arrival.**

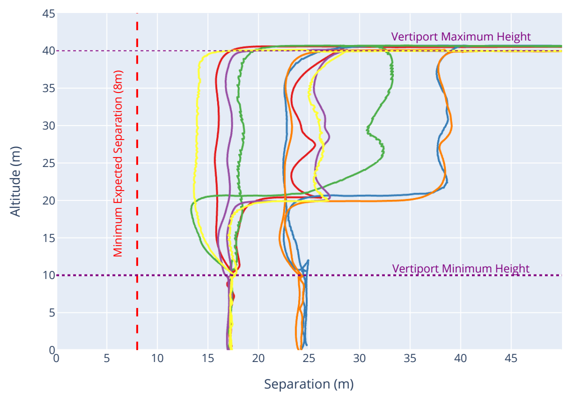
Fig. 8 Results of the helical pattern flight test 17 at Snowdonia Aerospace Center.

The users route request is facilitated with a redesign of the user interface. The screenshot shown in Fig. 7 shows the live interface during the flight test. The top two windows of the GUI are the primary control panels. The left is a safety critical panel monitoring the status of the drone and is run in its process to avoid system failure. The panel on the right allows a user to select routes for a specific drone to take. The selectable routes displayed are dynamic based on the current waypoint of the drone. The selection of waypoint first checks with the block manager to ensure that the route reservation is valid as previously described. Messages are shown when a route is completed, or if a route is unable to be reserved. The status of the block manager is shown in the bottom left window, with a number of blocks PP, QQ, RR and SS shown as occupied with drones in them. You can also see how 'AA' and 'BB' have been reserved as the drones are arriving in this snapshot from PA and QB respectively by drones 1 and 2. Drones 3 and 4 are awaiting a routing to a landing pad and thus remain unreserved. The remaining bottom right windows are from the mavproxy utility which shows a birds eye view of the drones with a visualization of the stored mission representing the vertiport, and a console for monitoring the internal status of a single drone.

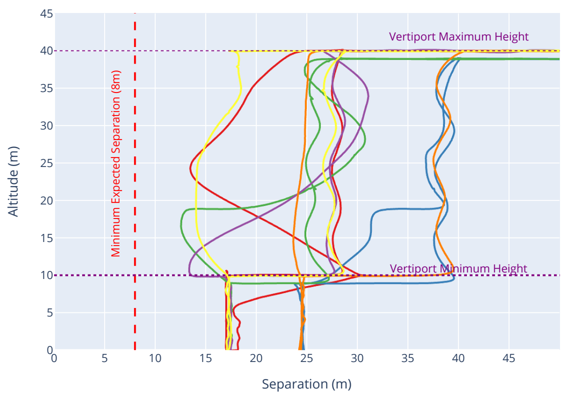
Table 2 shows the routes selected for flight test 17. This flight was intentionally designed such that all of the drones take off and ascend following the outer helix in order to visually demonstrate the vertiport layout. Fig. 8 shows the trajectories traversed during test flight 17. The helical paths can clearly be seen in the long-exposure photo of the departure. Plotting all of the trajectories, the spatial separation of the trajectories demonstrates how conflicts are avoided. The table shows the blocks which were automatically reserved when routes were selected during the departure and arrival flights. Note how the arrival routes and requested landing pads differ from the departure ones.

Drone ID	Routes	Departure	Arrival
Drone 1 (BLUE)	AS, SY, YS, SC	Takeoff: TOF01 Blocks: AA, SS, YY	Landing: TOF03 Blocks: YY, SS, CC
Drone 3 (AMBER)	BP, PZ, ZP, PD	Takeoff: TOF02 Blocks: BB, PP, ZZ	Landing: TOF04 Blocks: ZZ, PP, DD
Drone 4 (RED)	CQ, QX, XQ, QA	Takeoff: TOF03 Blocks: CC, QQ, XX	Landing: TOF01 Blocks: XX, QQ, AA
Drone 5 (GREEN)	DR, RW, WR, RB	Takeoff: TOF04 Blocks: DD, RR, WW	Landing: TOF02 Blocks: WW, RR, BB

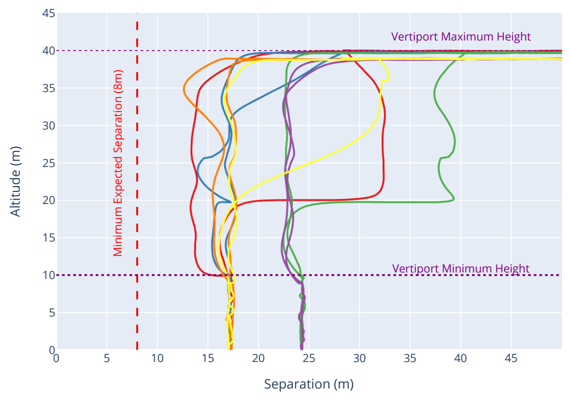
Table 2 The routes selected by the operator for each drone during flight test 17. The reserved blocks for departure and arrival are shown. These actions result in the trajectories logged in Fig. 8.



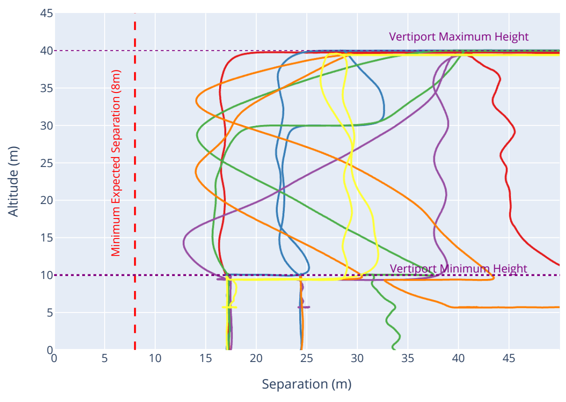
(a) Flight Test 17



(b) Flight Test 18



(c) Flight Test 19



(d) Flight Test 20

Fig. 9 This plot shows the minimum separation distance at a given altitude between every pair of drones for the 4 drone helical flight tests, for each of the separate tests conducted. The separation never goes below the minimum expected separation in any of the flight tests showing consistent spatial temporal separation within the vertiport.

4 Discussion of Results

A key validation is to ensure that there always exists an accepted amount of separation between any two pairs of drones within the vertiport itself. This is important in verifying the separation constraints enforced during the design of the routes. In total, four flight tests (17, 18, 19, and 20) were conducted in a helical pattern, each involving four drones. For each flight, Fig. 9 shows the minimum separation distance between each pair of drones for the flight test described above following routes in and out, generated a-priori from the drone logs. The minimum separation distance is plotted against the altitude for a visualization of spiral traversal. The routes in and out are manually controlled by the experiment operator but are intended to represent nominal operation. A minimum expected separation distance can be calculated at 8m as the difference between the inner spiral radius (12m) and the outer spiral radius (20m), with the minimum occurring if a drone is departing as another is arriving. The figure illustrates that no flown pair of routes results in a smaller separation than this value. This therefore demonstrates the safety factor gained by spatial separation through using the helical pattern generated through the method previously described.

5 Conclusion

An approach to the design and control of air operations around vertiports has been proposed. It is based on controlled switching between pre-determined routes, designed to support multiple simultaneous arrivals and departures. The goal is to resolve conflicts in flight in the terminal airspace, such that ground and en-route operations may be simplified. Several designs of paths and associated controllers have been discussed. Paths are designed to ensure spatial separation of independent routes. Control logic is designed along an airspace block principle, with no more than one drone allowed in a block at any time. Initial results have investigated the concept validity. Numerical simulations have shown potential performance of a fully automated system. Flight tests have validated the operability of a semi-automated scheme using the control logic to provide safety guards on commands from a human controller. Future iterations will explore routing for heterogeneous vehicles and handling contingency scenarios.

Acknowledgments

This work was supported by UKRI through Innovate UK Future Flight Challenge Phase 3 Strand 2 Project PORTAL (number 10023177).

References

[1] PwC. Skies without limits v2.0. Technical report, PwC, July 2022. `https://www.pwc.co.uk/intelligent-digital/drones/skies-without-limits-2022.pdf`.

[2] Karolin Schweiger and Lukas Preis. Urban air mobility: Systematic review of scientific publications and regulations for vertiport design and operations. *Drones*, 6(7):179, jul 2022. DOI: 10.3390/drones6070179.

[3] Parker D. Vascik, R. John Hansman, and Nicholas S. Dunn. Analysis of urban air mobility operational constraints. *Journal of Air Transportation*, 26(4):133–146, oct 2018. DOI: 10.2514/1.d0120.

[4] Parker D. Vascik and R. John Hansman. Development of vertiport capacity envelopes and analysis of their sensitivity to topological and operational factors. In *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, jan 2019. DOI: 10.2514/6.2019-0526.

[5] Jeffrey Homola. Aam project: High density vertiplex (hdv) research and capabilities. In *AAM WG's Community Integration Subgroup Tag Up*, 2023.

[6] Robert G McSwain, Jacob R Schaefer, Louis J Glaab, James Unverricht, Jeffrey Homola, Quang Dao, Faisal Omar, and Bryan Petty. High density vertiplex: Scalable autonomous operations flight test. In *AIAA SCITECH 2024 Forum*, page 0874, 2024. DOI: 10.2514/6.2024-0874.

[7] European Union Aviation Safety Agency (EASA). Vertiports prototype technical specifications for the design of vfr vertiportsfor operation with manned VTOL-capable aircraft. Available at `https://www.easa.europa.eu/downloads/136259/en`, accessed 22nd May 2023, Mar. 2022.

[8] Federal Aviation Administration, U.S. Department of Transportation. Advisory circular No. 150/5390-2C: Heliport design. Available at `https://www.faa.gov/documentLibrary/media/Advisory_Circular/150_5390_2c.pdf`, accessed 22nd May 2023., Apr. 2012.

[9] Karolin Schweiger, Franz Knabe, and Bernd Korn. An exemplary definition of a vertidrome's airside concept of operations. *Aerospace Science and Technology*, 125:107144, jun 2022. DOI: 10.1016/j.ast.2021.107144.

[10] Kyowon Song, Hwasoo Yeo, and Jung-Ho Moon. Approach control concepts and optimal vertiport airspace design for urban air mobility (UAM) operation. *International Journal of Aeronautical and Space Sciences*, 22(4):982–994, mar 2021. DOI: 10.1007/s42405-020-00345-9.

[11] EASA. Commission implementing regulation (eu) 2021/664 of 22 april 2021. Available at https://www.easa.europa.eu/en/document-library/regulations/commission-implementing-regulation-eu-2021664, accessed 09 May 2024.

[12] EUROCONTROL. Significant implementation progress as top airports increasingly turn to point merge, eurocontrols innovative arrival sequencing technique. https://www.eurocontrol.int/news/significant-implementation-progress-top-airports-increasingly-turn-point-merge-eurocontrols, Oct. 2021. https://www.eurocontrol.int/news/significant-implementation-progress-top-airports-increasingly-turn-point-merge-eurocontrols.

[13] Özlem Sahin Meric and Oznur Usanmaz. A new standard instrument arrival: the point merge system. *Aircraft Engineering and Aerospace Technology*, 85(2):136–143, mar 2013. DOI: 10.1108/00022661311302742.

[14] G Kichenside and A Williams. *Two Centuries of Railway Signalling*. Crecy, second edition, Sept. 2016. ISBN: 9780860936725.

[15] Skybrary. Procedural control. https://www.skybrary.aero/articles/procedural-control.

[16] Imke C. Kleinbekman, Mihaela A. Mitici, and Peng Wei. eVTOL arrival sequencing and scheduling for on-demand urban air mobility. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. IEEE, sep 2018. DOI: 10.1109/dasc.2018.8569645.

[17] Marco Tranzatto, Frank Mascarich, Lukas Bernreiter, Carolina Godinho, Marco Camurri, Shehryar Khattak, Tung Dang, Victor Reijgwart, Johannes Loeje, David Wisth, Samuel Zimmermann, Huan Nguyen, Marius Fehr, Lukas Solanka, Russell Buchanan, Marko Bjelonic, Nikhil Khedekar, Mathieu Valceschini, Fabian Jenelten, Mihir Dharmadhikari, Timon Homberger, Paolo De Petris, Lorenz Wellhausen, Mihir Kulkarni, Takahiro Miki, Satchel Hirsch, Markus Montenegro, Christos Papachristos, Fabian Tresoldi, Jan Carius, Giorgio Valsecchi, Joonho Lee, Konrad Meyer, Xiangyu Wu, Juan Nieto, Andy Smith, Marco Hutter, Roland Siegwart, Mark Mueller, Maurice Fallon, and Kostas Alexis. Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge. *arXiv preprint arXiv:2201.07067*, 2022. DOI: https://doi.org/10.48550/arXiv.2201.07067.

[18] Hirad Goudarzi and Arthur Richards. Semi-autonomous drone control with safety analysis. *Drone Systems and Applications*, 11:1–13, jan 2023. DOI: 10.1139/dsa-2022-0031.

[19] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[20] Oren Nechushtan, Barak Raveh, and Dan Halperin. *Sampling-Diagram Automata: A Tool for Analyzing Path Quality in Tree Planners*, pages 285–301. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN: 978-3-642-17452-0. DOI: 10.1007/978-3-642-17452-$0_1$7.

[21] Federal Aviation Administration. Airborne navigation databases. Available at https://web.archive.org/web/20090824075852/http://www.faa.gov/library/manuals/aviation/instrument_procedures_handbook/media/appndxA1.pdf, accessed 14 May 2024., 2009.

[22] MAVLINK. File formats. Available at https://mavlink.io/en/file_formats/, accessed 23 May 2023.

[23] MAVLINK. Using pymavlink libraries. Available at https://mavlink.io/en/mavgen_python/, accessed 23 May 2023.