Experimental Validation of the Reliability-Aware Multi-UAV Coverage Path Planning Problem

Mickey Li*
Autonomous Manufacturing Laboratory, University College London, United Kingdom

Arthur Richards†
Bristol Robotics Laboratory and University of Bristol, Bristol, United Kingdom

Mahesh Sooriyabandara‡
Bristol Research & Innovation Laboratory, Toshiba Research Europe Ltd, Bristol, United Kingdom

Unmanned aerial vehicles (UAVs) have become crucial for various applications, necessitating reliable and time-constrained performance. Multi-UAV solutions offer advantages but require effective coordination. Traditional coverage path planning methods overlook uncertainties and individual UAV failures. To address this, reliability-aware multi-UAV coverage path planning methods optimise task allocation to maximise mission completion probabilities given a failure model. This paper presents an experimental validation of the reliability-aware approach, specifically an approach using a Greedy Genetic Algorithm (GGA). We evaluate the GGA performance in real-world environments, comparing mission reliability to computed reliability and comparing it against a traditional multi-UAV methods. The experimental validation demonstrates the practical viability and effectiveness of the reliability-aware approach, showing significant improvement in mission reliability despite the inevitable mismatch between real and assumed failure models.

I. Introduction

Uncrewed aerial vehicles (UAVs)* have gained significant attention in recent years for various applications, including aerial robotics, surveillance, and infrastructure inspection [1, 2]. With the increasing deployment of UAVs in real-world scenarios, ensuring their reliability and performance within time constraints has become a critical requirement. Multi-UAV solutions offer advantages such as flexibility, scalability, and fault tolerance compared to single-UAV systems. However, effective coordination and task allocation among multiple UAVs pose challenges that need to be addressed for reliable and fastest operation [3]. Simultaneously, as the complexity and scale of missions involving multiple UAVs have grown, the need for reliability-aware planning approaches has become increasingly evident. Real-world applications often involve dynamic environments, time-varying mission requirements, and the potential for individual UAV failures. Consequently, traditional coverage path planning methods that solely aim to minimise cost or time fail to account for the inherent uncertainties and risks associated with UAV operations [4, 5].

Resilience to failures for multi-robot coverage is currently an active area of research. Early work [6, 7] defined a given plan as *robust* if the mission will eventually complete as long as at least one robot remains alive. This, however, is often the worst case, with no graceful degradation, likely providing conservative strategies in practice. More recently, Song and Gupta [8] consider MCPP in an unknown environment with failure-prone robots by applying distributed game theoretic decision methods in order to cooperatively decide task re-allocations on failure. Similar to our work, task allocations are evaluated by their probabilities of success given a model of battery reliability. While this addresses the issue of failure-prone robots, it should be noted that the approach was not validated through real-life experimentation. On the other hand, Ramachadran et al.[9] combine a reliability constraint on the optimisation of the number of UAVs required to complete an area coverage task, with a local re-planner to actively compensate if a robot is lost. Both of these methods are online reactive methods which seek to find actions which minimise the effect of robot failure on

*Senior Research Fellow in Robotics and AI for Aerospace Applications
†Professor of Robotics and Control, Senior Member AIAA
‡Managing Director of Bristol Research & Innovation Laboratory, Toshiba Europe Limited
*This paper primarily uses the term “Uncrewed aerial vehicle" (UAV) when discussing general applications. The term “Drone" refers to the smaller quad-rotors used in our experiment. When discussing general planning approaches, the vehicle may also be referred to as an “Agent"

the overall objective. Importantly, their work also demonstrates the effectiveness of their approach through real-world testing. Inspired by the insights provided by both Song's and Ramachandran's work, our research focused on the fundamental effect of planning initial paths with a reliability-aware objective function based on a priori analysis of known environments. This is similar to [10–12], where the authors' a-priori design strategies for non-coverage tasks are robust to a fixed number of failures during execution. However, our strategy optimisation also takes into account the failure of any number of robots.

To tackle these challenges, researchers have recognised the importance of developing coverage path planning approaches that explicitly consider reliability metrics. Reliability-aware multi-UAV coverage path planning methods seeks to optimise the allocation of tasks among UAVs, accounting for their individual failure rates, and ensuring high mission completion probabilities within specified time constraints [13]. By promoting overlapping robot-task allocations, these methods provide improved mission reliability, even at the expense of nominal-case time. Previous work focused on the development of these reliability-aware coverage path planning methods for multi-UAV systems. A brief introduction of the reliability evaluation framework along with a Linear Program solver for discrete time RA-MCPP is found in [14]. The next work [15] introduces the Reliability-Aware Multi-Agent Coverage Path Planning (RA-MCPP) problem, which extends the traditional Multi-Agent Coverage Path Planning (MCPP) framework by explicitly optimising the Probability of Completion (PoC) metric which incorporating agent failure rates into the path planning process. It also presents a path-based genetic algorithm for finding PoC optimal RA-MCPP path plans. However, in this previous work it is assumed that the environment can be discretised into a unit lattice with all moves occurring synchronously. This led to the introduction of the Greedy Genetic Algorithm [13] where, these two assumptions are relaxed, in order to apply RA-MCPP to larger, more complex and realistic environments.

To validate the RA-MCPP, we present in this paper the results of a carefully designed experimental validation of our RA-MCPP approach. Our aim is to bridge the gap between theory and practice, testing if reliability improvements can be observed, especially given the challenge of estimating the true agent failure model. In this experimental validation, we built upon our previous work, which introduced the concept of the Greedy Genetic Algorithm for reliability-aware coverage path planning. The GGA algorithm leverages both genetic algorithms and greedy heuristics to iteratively optimise the allocation of tasks among UAVs, considering their reliability characteristics. While our previous work showed promising results in theoretical scenarios, the experimental validation allows us to assess the performance and applicability of the GGA algorithm in real-world environments. We sought to evaluate the performance of the GGA algorithm by comparing the real-life mission reliability to the computed reliability. Additionally, we aim to compare the performance of our reliability-aware approach with traditional multi-UAV coverage path planning methods.

However care was required in the design of the experiment in order to reduce bias of the results. The experiment is carefully conducted in an indoor arena with three Coex Clover drones and a fixed set of batteries and chargers. All the methods were carefully designed and pre-registered before data collection. Special attention went to pre-registering [16] the protocols for operator intervention and maintenance in response to faults, e.g. when to push the stop button, as these could introduce real or perceived bias in the outcome.

The remainder of this paper is organised as follows. Section II provides a review of the RA-MCPP problem and the GGA method used in this validation. Section III presents the methodology and experimental setup employed for the validation of our reliability-aware approach. Section IV presents and discusses the experimental results, offering insights into the performance and effectiveness of our proposed methods. Finally, Section V concludes the paper by summarising our findings, discussing the implications of our work, and outlining future research directions.

II. Reliability-Aware Multi-Agent Coverage Path Planning

The Reliability-Aware Multi-Agent Coverage Path Planning Problem seeks, for a given set of agents and a set of connected tasks, to find the multi-agent path plan which maximises the probability of mission completion (PoC). The method adopts a reduced state representation of the drones $x = (t_1, \ldots, t_n) \in [0, \bar{t}]^n = \mathbf{S}$, capturing only the useful time worked by each drone $t_i^t \in \mathbf{R}$, with a mission deadline $\bar{t}$. The environment graph $G(\mathbf{J}, E)$ defines a set of m individual tasks, described by the nodes $\mathbf{J} = (j_1, \ldots, j_m)$, with the edges, E describing valid traversable paths between two tasks. For an agent i, the failure probability density is denoted $f_i(t)$, cumulative density $F_i(t)$, The probability of agent i surviving at time t is the *reliability* $R_i(t) = 1 - F_i(t)$.

Then, given a set of n agents, each following failure distributions $f_i(t)$ and environment graph $G(\mathbf{J}, E)$ containing a set of m task nodes and a set of traversable edges E between them. Let $\psi_i \in \Psi$ be a finite ordered subset of connected tasks $j \in \mathbf{J}$ (*i.e.* a path through G). A path for each agent forms a strategy $\psi = \{\psi_1, ..., \psi_n\} \in \Psi^n$. Given a strategy ψ, a subset of the agent work done states $\mathbf{S}$ can be defined to be *completion states*, such that reaching a completion state

implies completion of all tasks, and hence the mission. The probability of reaching a *completion state* can be analysed based on the reliability functions $R_i(t)$, yielding the $PoC(\psi, \bar{t})$. For full details, the reader is encouraged to refer to Section 3 of [13]. Then the optimisation objective is to find the n paths in the strategy $\psi = \psi_1, ..., \psi_n$ which maximises the reliability metric of probability $PoC(\psi)$ by a deadline $\bar{t}$, whilst ensuring all tasks have been visited.

$$\max_{\psi \in \Psi^n} PoC(\psi, \bar{t}) \tag{1}$$

subject to,

$$\bigcup_{i \in 1..n} \psi_i = \mathbf{J} \tag{2}$$

$$(\psi_{i,k}, \psi_{i,k+1}) \in E \quad \forall k < |\psi_i|, \forall i \in 1...n \tag{3}$$

This optimisation problem has been previously solved using a *Greedy-Genetic Algorithm* [13], which employs a meta-heuristic optimisation approach for finding high reliability multi-agent path plans. In this work it was found that the PoC is submodular, which implies that a greedy approach can give a bounded approximation of the optimal. This approach takes advantage of this finding and uses a genetic algorithm to greedily generate single candidate trajectories ψ_i which maximise PoC when combined with trajectories $\psi_1, ..., \psi_{i-1}$ found in previous iterations. This method was found [13] to find strategies with greater theoretical PoC than any other optimisation method, and provide significantly greater reliability compared to existing environment partition based multi-agent coverage methods.

III. Experimental Validation of Reliability-Aware Planning

The experimental method was to fly three drones in an indoor flying arena so as to visit a set of task locations. Flights were performed using two sets of paths, one set optimised for reliability by GGA, and the other optimised for time to completion. The real drones suffered from real failures, so the real completion rates can be analysed. Given the potential for bias in an experiment such as this, the experiment design, protocols and analysis described in this paper were *Pre-Registered* to the Open Science Framework[†] prior to conducting the experiment [16]. The intent of Pre-Registration is to demonstrate transparency and rigour in the application of the scientific process, ensuring that the experiment is not biased accidentally or otherwise by operator judgement. The following describes the core elements of the experimental methodology. Further details are given in the Pre-Registration [16], including details of the data collected, and the specific order of operations which is not included here for brevity.

A. Experiment Aim

This experiment aims to compare the mission performance rate of a reliability aware (RA) coverage path against a reliability-unaware, minimum-time (MT) coverage path using three drones in the flight arena. In each trial, the drones will fly one of the two coverage paths and it will be noted whether the mission was successful, *i.e.* all tasks were completed, or whether it failed. Through repeating the trials we will gain a series of successes and failures through which we can analyse the hypothesis. The hypotheses are the following:

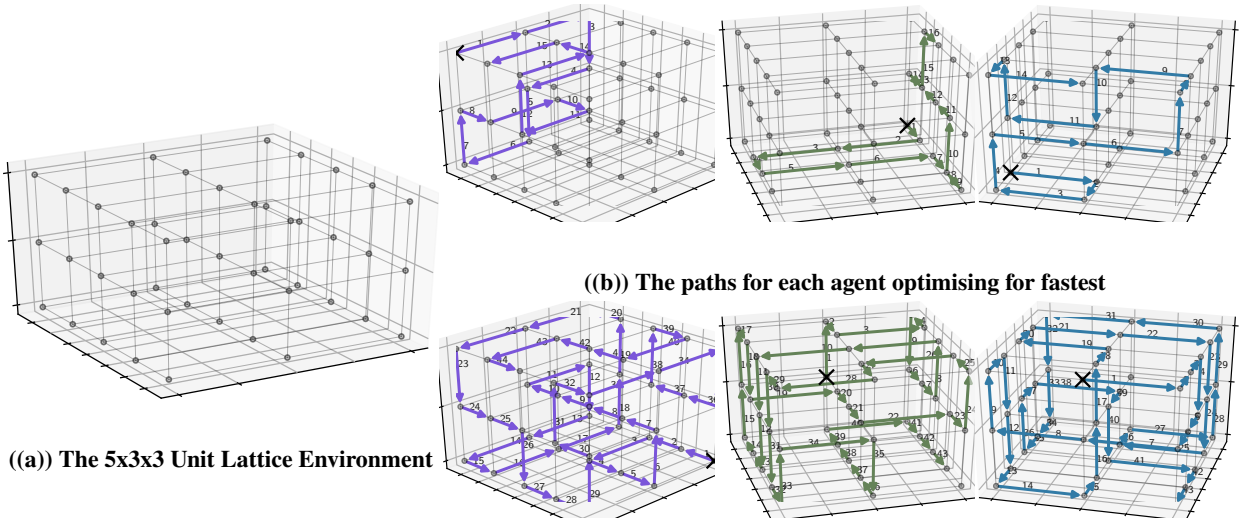
- Null Hypothesis (H_0): Three drones flying trajectories optimised for maximum probability of completion of all tasks provide the same probability of mission completion when compared to the same drones flying trajectories optimised for minimum total flight time.
- Alternative Directional Hypothesis (H_1): **Three drones flying trajectories optimised for maximum probability of completion of all tasks provide a greater probability of mission completion than the same drones flying trajectories optimised for minimum total flight time.**

B. Experiment Design and Methodology

The three drones form our group of study participants which will be tasked with performing a number of trials.

Each trial involved the flight of the drones along one of two possible sets of trajectories which cover a 5x3x3 3D lattice environment graph of physical size 4m x 1.5m x 1.5m, where each node of the graph is a task. The trajectory generation generates an individual trajectory for each vehicle, i.e. 3 individual trajectories. The two sets of trajectories, shown in Figure 1, are generated by optimising for the following:

[†]Open Science Framework `https://osf.io/`



((a)) The 5x3x3 Unit Lattice Environment

((b)) The paths for each agent optimising for fastest

((c)) The paths for each agent optimising for reliability

Fig. 1 The environment and planned trajectory paths for 'fastest' and for 'reliable' used in this experiment.

1) Fastest time - A set of trajectories generated by partitioning the environment into almost-equal connected task points and generating a travelling salesman path for each vehicles partition. This represents the traditional MCPP approach.
2) Highest reliability - A set of trajectories generated by the Greedy Genetic Algorithm [15] which aims to find paths for each robot which maximises the probability of mission completion by a deadline $\bar{t}$.

A key challenge with the method discussed is that the true failure model of each drone is unknown. Methods for deriving a drone's failure model are out of the scope of this paper. In this work, a best guess based on a Bathtub model [13, 17] is used. A Bathtub model is a 3-Weibull Mixture Model, with each model representing three stages of possible failure: (1) Early Failure, (2) Random Failure during operation (3) End-of-life failures. In this work a model is hand-approximated based on the maximum flight time observed from the pilot study and previous flights. This is the same model described in [13]. Note that even though the trajectories are generated within the same 5x3x3 volume, in practice to avoid overflight and collisions, the drone trajectories are shifted relative to each other, such that the drones remain at least 3 meters apart.

This Study can be considered as within-subject design as the *same set of drones* are tasked to fly a pre-computed fastest or reliable trajectory. Each trial was randomly assigned a set of drones to fly either fastest or reliable trajectories. This is inline with real usage as a user would not (presumably) have access to new drones for each mission.

1. Trial Randomisation

For this experiment, the variables are randomise for each trial before starting the experiment. This variable randomisation is strictly adhered to in order to minimise the possibility of introducing bias. Since we are analysing random failure, where the "failure" encompasses any and all possible failure methods, we have to ensure that there is minimum correlation between a particular type of physical failure to the methodology being used. For instance it would be an unfair experiment if defective battery 3 was always accidentally paired with the fast method. Therefore we have identified the various aspects of the experiment in the list below to randomise.

Each trial will be randomly assigned to one of two groups: group A will fly trajectories optimised for reliability; group B will fly trajectories optimised for time. Therefore, each trial uses a-priori simple randomisation of the following variables per trial:

1) **Method**: [Highest reliability, Fastest time]. The independent variable.
2) **Trajectory Ordering**: [All permutations of (0, 1, 2)]. This is required to randomise the allocation of drone to a particular trajectory.
3) **Drone Position in Physical Space**: [Left, Centre, Right]. This is required to reduce air flow effects on a

particular drone. *e.g.* Clover2 can have trouble flying in the centre location due to excessive prop-wash from both sides.

4) **Battery Allocation to Drone**: 6 Batteries in total, allocated successive groups of three. [Permutation of Tattu batteries (clover1, clover2, clover3), 2300mAh, 4S, 14.8V], [Permutation of Overlander batteries (clover5, clover6, clover8), 2200mAh, 4s, 14.8V]. This is required to ensure that battery characteristics do not negatively affect any one drone in particular. All the batteries are not new and have previously been used an unknown amount.
5) **Battery Allocation to Battery Chargers**: [Permutations of 6]. This is required to ensure that battery charger characteristics do not negatively affect any one battery in particular. The chargers are not new and have been previously used.

The allocation was randomised using the RAND function in Google sheets per trial, and then frozen by copying the cell values. During the experiment, the randomised values were followed for each trial.

2. Sampling and Data Collection

A pilot study of 40 flights was conducted to check procedures for data collection and analysis. This also looked at rough reliability rates to inform sample size and procedures for the main study. It also identified and fixed a few infrastructure problems, specifically network drop-outs and motion capture problems.

From the pilot study, a target sample size of 80-100 sample sizes was estimated, based on a chi squared (χ^2) goodness of fit test with significance level of 0.05, and medium effect size (0.25) and a statistic power of between 0.6 and 0.7. The effect size is estimated from our pilot data. When the experiment was undertaken, the results were close to, but not over the significance threshold after 100 trials ($p = 0.06 > 0.05$, see results). The authors decided to undertake a further 20 trials. This was the only deviation from the pre-registered method for this study.

In addition, flights are constrained by time, since only one set of flights can be performed at a time. Each is five minutes of flying time but takes about 15 minutes including battery fitting and testing, set-up, recording, and charging delays. They are also constrained by maintenance pending any major failures. Therefore the maximum allowed time to conduct this experiment is capped to 2 weeks.

3. Trial Outcomes

A task is considered **complete** if any drone is assigned that task as its next task and it briefly stops and stabilises within 10cm of the task location. When the central task monitor observes that all tasks have been completed, it sends a signal to land the drones and the trial is considered completed and successful.

A trial is considered failed if the system is unable to complete all tasks due to but not limited to the following factors:

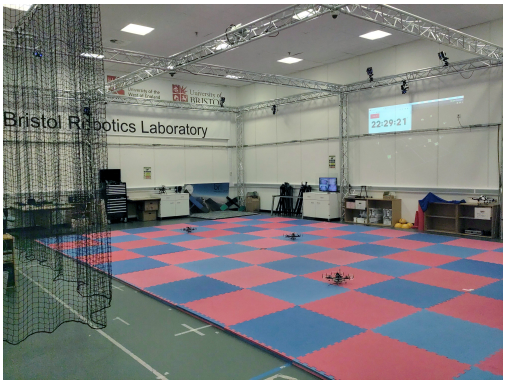
- Hardware or Communications Faults causing the drone to land itself
- Battery capacity reaching minimum threshold causing the drone to land itself
- Operator cancelling flight due to drone flying in an unsafe manner.

If a hardware fault occurs, the Operator will do their best to fix and maintain the drones, including actions such as replacing propeller guards, replacing chipped propellers, re-fixing motion capture balls, and so on. There exists one extra set of core replacement electronics such as the autopilot, power distribution board or companion computer which are to be used with no extra parts being purchased. In the event of a catastrophic non-fixable failure, the study will be terminated.

The operator has the ability to abort (land) and e-stop individual vehicles if they are deemed to be flying in an unsafe manner. Observed from the pilot study, this mainly covers bouts of poor control and lost tracking where the vehicle does following:

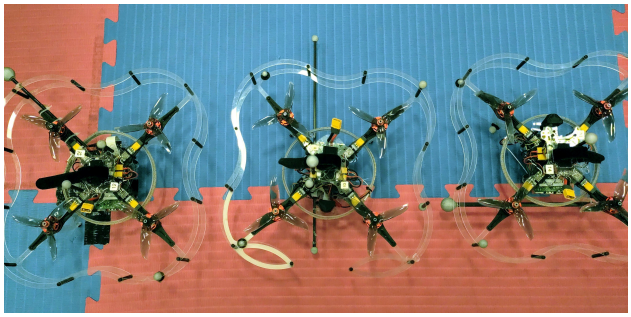
- Drifts slowly out of the safe flying volume marked by the mats
- Accelerates unexpectedly in one direction and appears to have lost tracking and control.
- Loses vertical tracking and accelerates into the ceiling.
- Extreme sudden acceleration in a random direction, identified by sudden motor pitch increase.
- Fails to stabilise into a hover during takeoff or flying to initial position for more than 30 seconds.

If the operator spots any of these patterns occurring, the operator will attempt to individually abort. Unfortunately, depending on the nature of the failure, the abort may not safely land the vehicle, or the vehicle will get stuck on a wall or netting and attempt to accelerate out. The operator will then attempt to e-stop that individual vehicle, which should stop the motors. Sometimes the individual e-stop commands have been observed to not immediately stop the motors on button press. Therefore in dangerous or unresponsive situations, the arena e-stop button will be pressed which stops all

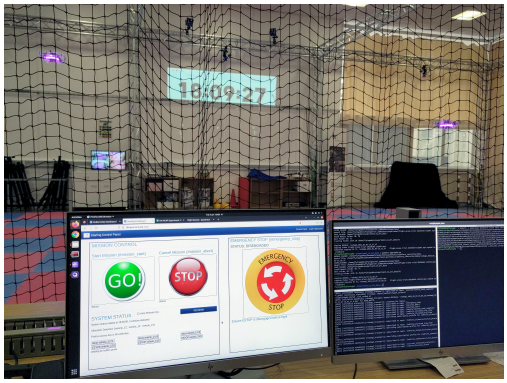


((a)) The flying arena with VICON cameras.

((b)) The battery charging setup with 6 batteries



((c)) The 3 Coex Clover quadrotors used for this experiment



((d)) Image of the experiment in progress from the operators view. The GUI is visible for vehicle monitoring.

Fig. 2 The facilities and equipment used within this experiment.

vehicles.

If the mission can still be completed after the failure of a single vehicle (due to operator abort or vehicle itself), the trial will still be successful. If all vehicles fail before mission completion due to the vehicles themselves, or operator conducted arena wide abort or estop, the trial will be recorded as a failure.

C. Experiment Implementation

1. System Hardware

The experiment uses three Coex Clover drones. These Clovers are of an X-shaped quadcopter configuration and run the PX4 1.12.3 autopilot firmware on top of a Coex Pix modified PixRacer autopilot. The PX4 cascading PID control stack for each drone has been manually tuned to the best of our ability prior to the experiment. Each Clover carries a Raspberry Pi 4B as a companion computer, connected by serial to the autopilot. The Pi performs the onboard control described in the next subsection.

The drones are localised using a 12-camera VICON motion capture system using the Tracker v9.0.1 software. VICON uses infrared to detect pre-defined rigid shapes composed of small reflective balls which have been rigidly attached to each drone. It calculates the position and orientation of each vehicle relative to the arena origin. These are sent to each drone's companion computer via a wifi connection, and these packets are then forwarded to the autopilot controller. The VICON motion capture system was calibrated at the start of each experiment day.

Figure 2 shows a number of pictures showing the parts of the system. This includes pictures of the Coex Clover drone in detail, a picture of the 3 Coex Clover drones used in the experiment (numbered 12,13,14), the flying arena where the motion capture cameras and timing can be seen at the top, and the batteries and battery chargers used.

	Highest reliability	Fastest time	Total
Completed	54	46	100
Not completed	6	14	20
>=1 failure	23	14	37
Total	60	60	120
PoC	90%	76.67%	83.33%

Table 1 The total number of completed and not completed trials for each method, as well as the number of trials in which more than one drone failed. The Proportion of Completions is reported.

Trajectories	Highest reliability	Fastest time	Total
t0_t1_t2	10	12	22
t0_t2_t1	15	9	24
t1_t0_t2	11	10	21
t1_t2_t0	11	7	18
t2_t1_t0	9	10	19
t2_t0_t1	4	12	17
Total	60	60	120

Table 2 The total number of trials for each trajectory allocation permutation for each method.

2. System Architecture and Software

As mentioned before, this experiment is built upon the Starling infrastructure [18]. The details pertinent to the description of this system is that it primarily utilises the Robot Operating System (ROS2 Foxy) framework for implementation and for communication between the different functions within this system [19]. Onboard the drones, the companion computer runs several ROS nodes. A ROS node controls the onboard sensors and LED lights for signalling and control. Another node runs MAVROS [20] which is the bridge between our onboard ROS controller and the MAVLINK commands required by the autopilot. Finally there is an experiment controller which performs the trajectory following required. It accepts the trajectory from the allocator and inputs it into a linear interpolator to ensure the vehicle flies at a set speed from task to task. Once loaded, the controller flashes its lights to signal that it is ready to start. The onboard controller runs a state machine which runs through the following states: *(1) Init*, *(2) Takeoff*, *(3) GoToMissionStart*, *(4) Execute*, *(5) Stop*, *(6) Land* and *(7) MakeSafe*.

The user has access to a physical set of buttons which send either a *Mission Go*, *Mission Abort* or *Emergency Stop*. Each sends a corresponding ROS message over the network. States (2), (3) and (4) require a Mission Go before continuing onto the next state. If the user presses Abort or the E-Stop or there is an onboard error such as loss of communications with the ground or the flight controller, the controller goes into the Stop state (5) followed by landing (6) and disarming (7). This interface can be seen in Figure 2

If a drone recognises that it has completed a task, it will send out a notification message to the central offboard mission controller. The offboard mission controller also runs on the central arena server and is responsible for monitoring the status of the mission. It also receives the current set of trajectory allocations which is parsed for a list of tasks. If all tasks have been registered by the drones as completed, it will send automatically send a *Mission Abort* to signal the end of the mission.

IV. Analysis and Results

Overall 120 trials were conducted and recorded of which 60 trials used the reliability aware method, and 60 trials used the fastest time method. The analysis described in this section follows the pre-registered procedure. A summary of trials is given in tables 1 and 2. Table 1 shows the total number of mission completions and failures for each method. It also shows the number drone failures that were observed. This experimental data has been released as an open dataset[‡] [21].

A. Contingency Table Analysis

Both the Fisher and Pearson test require the data in 2x2 contingency table form before analysis begins. Table 3 shows the contingency table for our experiment, with each observation being $O_{i,j}$ for row i and column j. Both tests seek to determine whether the two categorical variables are independent in influencing their respective test statistics. It is often regarded that the Pearson chi-squared test is used for large sample sizes such that the null hypothesis is valid when the test statistic is distributed under the chi-squared, χ^2 distribution. The Fisher's Exact Test is used for small

[‡]Data are available at the University of Bristol data repository, `data.bris`, at `https://doi.org/10.5523/bris.1577dh6ttrv3a2kadjorbrwsjv`.

	Highest Reliability	Fastest Time	Total
Mission Completed	54	46	100
Mission Not Completed	6	14	20
Total	60	60	120

Table 3 The 2x2 Contingency Table

sample sizes where one or more entries in the contingency table have a frequency of 5 or fewer and tests against the hypergeometric distribution. Since the frequencies in our table are somewhere in between the two recommended sample sizes, we use both tests to determine whether there is a significant difference between the expected frequencies and the observed frequencies in our experiment.

To start off with the Pearson's χ^2 test for statistical Independence between the variables. The test functions by comparing the observed frequencies to the theoretical frequencies for a cell, given the hypothesis of independence. I.e. that each data cell in the table has expected entry of $E_{i,j} = Np_ip_j$, where N is the total sample size (120). $p_i = \frac{O_{i*}}{N}$ and $p_j = \frac{O_{*j}}{N}$, where they are the fraction of observations of a row or column respectively for each variable combination. For example in our table, $i = 1, j = 1$ would have $p_i = 100/120$ and $p_j = 60/120$ with $E_{1,1} = 50$. Then the χ^2 distributed test statistic is given by the following:

$$\chi^2 = \sum_{i=1}^{2}\sum_{j=1}^{2}\frac{\left(O_{i,j} - E_{i,j}\right)^2}{E_{i,j}} = 3.84 \tag{4}$$

The test is then conducted by computing the probability of getting such a statistic value on a χ^2 distribution of degrees of freedom $k = N_r + N_c - 1$, where N_r and N_c are the number of rows and columns respectively. In our 2x2 example this gives us dof $k = 1$. The one tailed test can therefore be formulated as:

$$p = Pr\left(\chi_1^2 \geq 3.84\right) = 0.05 \leq 0.05 \tag{5}$$

which rejects the null hypothesis at the $\alpha = 95\%$ level in favour of the alternative hypothesis of a significant difference in completion likelihood between using the Reliable and Fastest methods. The effect size of this test can be calculated for our sample using the ϕ coefficient which shows the impact of using reliability over efficiency. The effect size is calculated to be $\phi = 0.178$ which is regarded as a small to medium impact from using reliability over efficiency on probability of mission completion for coverage path planning in practice.

As secondary analysis, the Fisher's Exact Test is conducted which calculates the exact significance of the deviation from the null hypothesis. It achieves this by showing that any entry in the contingency table, conditioned on the marginals, is distributed as a hypergeometric distribution on the marginals where the probability of obtaining such a set of values out of the set of all possible tables with given marginals is the following:

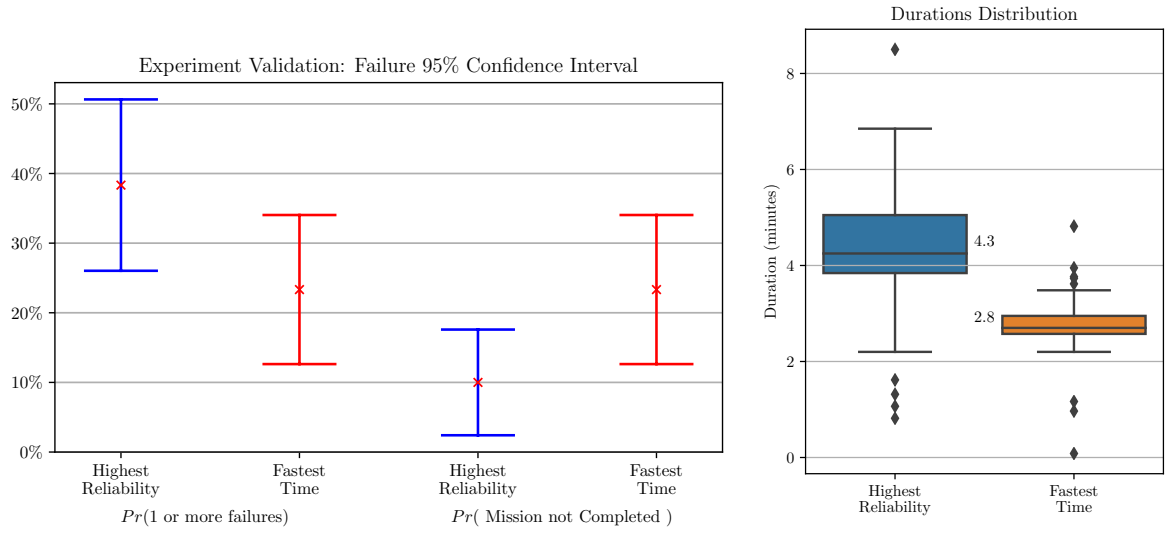
The P-value for a one tailed test is then calculated as the sum of the probabilities for all tables with the same marginals having a probability equal to or greater than $O_{11} = 54$. This gives the following if X is our test statistic following the hypergeometric distribution:

$$p = Pr(X \geq 54) = 0.042 \leq 0.05 \tag{6}$$

Which also provides enough justification to reject the null hypothesis at the $\alpha = 95\%$ level in favour of the alternative hypothesis of a significant difference in completion likelihood between using the Reliable and Fastest methods.

B. Confidence Interval Analysis

Another common test is to form confidence intervals for each category of data. A k% confidence interval (CI) around, say, the mean gives you the likelihood that the true mean is within the interval is k%. In our scenario we can



((a)) 95% Confidence Intervals **((b)) Duration**

Fig. 3 (a) The 95% Confidence Intervals representing from left to right (1) The probability of drone failure using Reliable (2) The probability of drone failure using fastest (3) The probability of non-completion using reliable (4) The probability of non-completion using fastest. (b) The distribution of mission durations.

form two 95% CIs, representing the probability of non-completion using the reliable or fastest method respectively. A test is formed by determining whether the difference between the means is statistically significant.

$$(\hat{p}_L, \hat{p}_U) = \hat{p} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \qquad \hat{p} = \frac{\#\text{failures}}{n} \tag{7}$$

where again $z = 1.96$ is 95th percentile of the standard normal distribution.

One simple method of testing is to plot the confidence intervals and observe whether the two intervals overlap. If those intervals overlap, they conclude that the difference between methods is not statistically significant. If there is no overlap, the difference is significant. The interval bounds can be calculated using the Wald Binomial Proportion Interval for each method and plotted in Figure 3(a).

Observe in Figure 3(a) that the rate of failure is higher for Reliable. This is likely due to the slightly longer flight durations on average as seen in Figure 3(b) where Reliability takes on average an extra 2 minutes compared to the Minimum Time paths. Despite this, the rate of mission failure is still much lower than that of the Fastest path. This shows that the mission can succeed even if drone failures occur, hence why the CIs for Reliable differ, whereas they are identical for Fastest. The figure shows overlap between the CIs for non-completion between the two methods, therefore not confirming statistical significance of the difference in probabilities. However, it has been found that simply observing overlap between confidence intervals is an extremely conservative metric and does not necessarily imply that the difference between the two means are statistically significant. An alternative method of analysis is to form a confidence interval from the *difference in means* between the two groups. Given the frequencies shown in Table 3, Let us form the observed proportions $\hat{p}_r = 6/60$ and $\hat{p}_e = 14/60$. The observed difference is then given by $\hat{d} = \hat{p}_e - \hat{p}_r$ with total sample size of $N = n_1 + n_2 = 120$. For a 95% confidence interval of the mean difference, the Wald interval can be formed from the following [22]:

$$(\hat{p}_L, \hat{p}_U) = \hat{d} \pm z\sqrt{\frac{\hat{p}_r(1-\hat{p}_r)}{n_1} + \frac{\hat{p}_r(1-\hat{p}_r)}{n_2}} \tag{8}$$

where again $z = 1.96$ is 95th percentile of the standard normal distribution. Putting in the values, this outputs the 95% confidence interval from $[0.21, 26.45]$. This interval contains the range of possible probability differences. Since zero

is not in the interval, this implies that it is unlikely for there to be no difference between the means and hence the result is also statistically significant.

V. Conclusion

This paper presents a real world validation study on the efficacy of our Reliability-Aware Coverage Path Planning Methods in comparison to a reliability-unaware, minimum-time coverage path plan. In this study, 3 real world drones are flown on a randomised mission of a set of trajectories generated from either plan, and the mission completion status is reported. Overall 120 trials were collected and the results statistically analysed, identifying that the Reliability-Aware methods made a significant improvement ($\alpha = 0.05$) to the likelihood of mission completion compared to minimum time plans. Despite the discussed limitations of sample size and failure detection, these findings provide real world validation for the simulated and theoretical results described in previous chapters. It also demonstrates that considering reliability as a valid method of resilient multi-agent planning and lays the groundwork for further research into reliability-aware planning.

This validation also exposes a number of key future works. This work requires the knowledge of individual agent failure distributions. As seen in this work, using an approximate failure distribution can still generate reliable paths, but an investigation is needed into how to both derive an accurate failure model for an agent (for example from the failure data collected in this experiment), and to quantify the presumed improvement in reliability if the experiment was repeated with more accurate failure models. In addition it would provide interesting results if this experiment was repeated using different vehicles and in different environments to determine how much the reliability can be improved by in practice.

Acknowledgments

This work has been funded by the Engineering and Physical Sciences Research Council (EPSRC) iCASE with Toshiba Research Europe Ltd and FARSCOPE Centre of Doctoral Training at the Bristol Robotics Laboratory, United Kingdom

References

[1] Gupte, S., Mohandas, P. I. T., and Conrad, J. M., "A survey of quadrotor Unmanned Aerial Vehicles," *2012 Proceedings of IEEE Southeastcon*, IEEE, 2012, pp. 1–6. https://doi.org/10.1109/SECon.2012.6196930, URL http://ieeexplore.ieee.org/document/6196930/.

[2] Chung, S.-J., Paranjape, A. A., Dames, P., Shen, S., and Kumar, V., "A Survey on Aerial Swarm Robotics," *IEEE Transactions on Robotics*, Vol. 34, No. 4, 2018, pp. 837–855. https://doi.org/10.1109/TRO.2018.2857475, URL https://ieeexplore.ieee.org/document/8424838/.

[3] Prorok, A., Malencia, M., Carlone, L., Sukhatme, G. S., Sadler, B. M., and Kumar, V., "Beyond Robustness: A Taxonomy of Approaches towards Resilient Multi-Robot Systems," 2021. https://doi.org/10.48550/arxiv.2109.12343, URL https://arxiv.org/abs/2109.12343v1.

[4] Korsah, G. A., Stentz, A., and Dias, M. B., "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, Vol. 32, No. 12, 2013, pp. 1495–1512. https://doi.org/10.1177/0278364913496484, URL http://journals.sagepub.com/doi/10.1177/0278364913496484.

[5] Galceran, E., and Carreras, M., "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, Vol. 61, No. 12, 2013, pp. 1258–1276. https://doi.org/10.1016/J.ROBOT.2013.09.004, URL https://www.sciencedirect.com/science/article/pii/S092188901300167X.

[6] Hazon, N., and Kaminka, G. A., "On redundancy, efficiency, and robustness in coverage for multiple robots," *Robotics and Autonomous Systems*, Vol. 56, No. 12, 2008, pp. 1102–1114. https://doi.org/10.1016/J.ROBOT.2008.01.006, URL https://www.sciencedirect.com/science/article/pii/S0921889008000122.

[7] Fazli, P., Davoodi, A., Pasquier, P., and Mackworth, A. K., "Complete and robust cooperative robot area coverage with limited range," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 5577–5582. https://doi.org/10.1109/IROS.2010.5651321, URL http://ieeexplore.ieee.org/document/5651321/.

[8] Song, J., and Gupta, S., "CARE: Cooperative Autonomy for Resilience and Efficiency of robot teams for complete coverage of unknown environments under robot failures," *Autonomous Robots*, Vol. 44, No. 3-4, 2020, pp. 647–671. https://doi.org/10.1007/s10514-019-09870-3, URL https://doi.org/10.1007/s10514-019-09870-3.

[9] Ramachandran, R. K., Preiss, L. Z. J. A., and Sukhatme, G. S., "Resilient Coverage: Exploring the Local-to-Global Trade-off," 2019. URL http://arxiv.org/abs/1910.01917.

[10] Zhou, L., Tzoumas, V., Pappas, G. J., and Tokekar, P., "Resilient active target tracking with multiple robots," *IEEE Robotics and Automation Letters*, Vol. 4, No. 1, 2019, pp. 129–136. https://doi.org/10.1109/LRA.2018.2881296.

[11] Saulnier, K., Saldana, D., Prorok, A., Pappas, G. J., and Kumar, V., "Resilient Flocking for Mobile Robot Teams," *IEEE Robotics and Automation Letters*, Vol. 2, No. 2, 2017, pp. 1039–1046. https://doi.org/10.1109/LRA.2017.2655142.

[12] Guerrero-Bonilla, L., Prorok, A., and Kumar, V., "Formations for Resilient Robot Teams," *IEEE Robotics and Automation Letters*, Vol. 2, No. 2, 2017, pp. 841–848. https://doi.org/10.1109/LRA.2017.2654550.

[13] Li, M., Richards, A., and Sooriyabandara, M., "Asynchronous Reliability-Aware Multi-UAV Coverage Path Planning," *IEEE International Conference on Robotics and Automation (ICRA)*, 2021. https://doi.org/10.1109/ICRA48506.2021.9560770.

[14] Li, M., Richards, A., and Sooriyabandara, M., "Reliability-Aware Multi-UAV Coverage Path Planning Using Integer Linear Programming," *UKRAS20 Conference: "Robots into the real world" Proceedings*, EPSRC UK-RAS Network, 2020, pp. 15–17. https://doi.org/10.31256/Cy5Ej9K, URL https://www.ukras.org/publications/ras-proceedings/UKRAS20/pp15-17.

[15] Li, M., Richards, A., and Sooriyabandara, M., "Reliability-aware multi-UAV coverage path planning using a genetic algorithm," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, Vol. 3, 2021.

[16] Li, M., and Richards, A., "Reliability-aware Multi-robot Coverage Path Planning - Drone Validation Study," 2022. https://doi.org/10.17605/OSF.IO/72JB4, URL https://osf.io/72jb4.

[17] Lai, C., Xie, M., and Murthy, D., "A modified Weibull distribution," *IEEE Transactions on Reliability*, Vol. 52, No. 1, 2003, pp. 33–37. https://doi.org/10.1109/TR.2002.805788, URL http://ieeexplore.ieee.org/document/1179794/.

[18] Li, M., Clarke, R., and Richards, A., "Starling: Containerisation Architecture for Scalable Local Development, Deployment and Testing of Multi-UAV Systems," *Robotic Science and Systems 2022 Workshop on Envisioning an Infrastructure for Multi-Robot and Collaborative Autonomy Testing and Evaluation*, 2022. URL http://raaslab.org/rss2022/assets/contributed_papers/RSS2022_Li_Clarck_Richards.pdf.

[19] Maruyama, Y., Kato, S., and Azumi, T., "Exploring the performance of ROS2," *Proceedings of the 13th International Conference on Embedded Software, EMSOFT 2016*, 2016. https://doi.org/10.1145/2968478.2968502, URL http://dx.doi.org/10.1145/2968478.2968502.

[20] Ermakov, V., "mavlink/mavros: MAVLink to ROS gateway with proxy for Ground Control Station," , 2022. URL https://github.com/mavlink/mavros.

[21] Li Mickey, and Richards Arthur, "Drone Reliability Experiment," , 2023. https://doi.org/10.5523/BRIS.1577DH6TTRV3A2KADJORBRWSJV.

[22] Laud, P. J., and Dane, A., "Confidence intervals for the difference between independent binomial proportions: Comparison using a graphical approach and moving averages," *Pharmaceutical Statistics*, Vol. 13, No. 5, 2014, pp. 294–308. https://doi.org/10.1002/PST.1631.