



LJMU Research Online

Page, V, Dadswell, C, Webster, M, Jump, M and Fisher, M

Towards the Determination of Safe Operating Envelopes for Autonomous UAS in Offshore Inspection Missions

<http://researchonline.ljmu.ac.uk/id/eprint/15339/>

Article

Citation (please note it is advisable to refer to the publisher's version if you intend to cite from this work)

Page, V, Dadswell, C, Webster, M, Jump, M and Fisher, M (2021) Towards the Determination of Safe Operating Envelopes for Autonomous UAS in Offshore Inspection Missions. Robotics, 10 (3). ISSN 2218-6581

LJMU has developed **LJMU Research Online** for users to access the research output of the University more effectively. Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Users may download and/or print one copy of any article(s) in LJMU Research Online to facilitate their private study or for non-commercial research. You may not engage in further distribution of the material or use it for any profit-making activities or any commercial gain.

The version presented here may differ from the published version or from the version of the record. Please see the repository URL above for details on accessing the published version and note that access may require a subscription.

For more information please contact researchonline@ljmu.ac.uk

<http://researchonline.ljmu.ac.uk/>

Article

Towards the Determination of Safe Operating Envelopes for Autonomous UAS in Offshore Inspection Missions

Vincent Page ^{1,*}, Christopher Dadswell ¹, Matt Webster ², Mike Jump ¹ and Michael Fisher ³

- ¹ Department of Mechanical, Materials and Aerospace Engineering, School of Engineering, Faculty of Science and Engineering, University of Liverpool, Liverpool L69 3BX, UK; c.m.dadswell@liverpool.ac.uk (C.D.); mjump1@liverpool.ac.uk (M.J.)
- ² School of Computer Science and Mathematics, Faculty of Engineering and Technology, Liverpool John Moores University, Liverpool L2 2QP, UK; m.p.webster@ljmu.ac.uk
- ³ Department of Computer Science, Faculty of Science and Engineering, School of Engineering, University of Manchester, Manchester M13 9PL, UK; michael.fisher@manchester.ac.uk
- * Correspondence: v.page@liverpool.ac.uk

Citation: Page, V.; Dadswell, C.; Webster, M.; Jump, M.; Fisher, M. Towards the Determination of Safe Operating Envelopes for Autonomous UAS in Offshore Inspection Missions. *Robotics* **2021**, *10*, 97. <https://doi.org/10.3390/robotics10030097>

Academic Editor: Roberto Sabatini

Received: 20 May 2021

Accepted: 21 July 2021

Published: 28 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Abstract: A drive to reduce costs, carbon emissions, and the number of required personnel in the offshore energy industry has led to proposals for the increased use of autonomous/robotic systems for many maintenance tasks. There are questions over how such missions can be shown to be safe. A corollary exists in the manned aviation world for helicopter–ship operations where a test pilot attempts to operate from a ship under a range of wind conditions and provides subjective feedback on the level of difficulty encountered. This defines the ship–helicopter operating limit envelope (SHOL). Due to the cost of creating a SHOL there has been considerable research activity to demonstrate that much of this process can be performed virtually. Unmanned vehicles, however, have no test pilot to provide feedback. This paper therefore explores the possibility of adapting manned simulation techniques to the unmanned world to demonstrate that a mission is safe. Through flight modelling and simulation techniques it is shown that operating envelopes can be created for an oil rig inspection task and that, by using variable performance specifications, these can be tailored to suit the level of acceptable risk. The operating envelopes produced provide condensed and intelligible information regarding the environmental conditions under which the UAS can perform the task.

Keywords: aerial robots/UAV; inspection robots; hazardous environments; autonomous systems; navigation and exploration; oil and gas robots; modelling and simulation; validation and verification

1. Introduction

With more than five decades of production and supply chain experience, exporting around the world, the oil and gas sector is a UK industrial success story [1]. Nevertheless, it is facing a number of significant challenges over both the short and long term, including ageing infrastructure, declining production rates, reduced drilling exploration success rates, and an ageing workforce [2]. More recently, these have been compounded by low oil prices and the demand to shift to more renewable energy sources. All of this has led to a drive to increase competitiveness by reducing the overall cost of the operation of existing energy fields. One of the solutions being explored to achieve this is the increased use of unmanned and autonomous robotic platforms.

Autonomous systems are systems which decide for themselves what to do [3]. Typically, these decisions are made using computer software, which controls the system(s) in question and performs operations that might otherwise be performed by a human—in

essence, the cognitive element of performing a task. For example, an autonomous unmanned aerial system (UAS, aka “drone”), the focus of this paper, will need to contain computer systems that can replace the function of a human pilot operating the UAS by remote control when operating beyond visual line of sight [4].

In this paper, an autonomous system is defined as the following:

An unmanned system (UMS) wherein the UMS receives its mission from the human and accomplishes that mission with or without further human–robot interaction [5]

Traditionally, autonomous robotic systems are considered to be well suited for tasks carried out in hazardous environments; the so-called dull, dirty, and/or dangerous missions (commonly referred to as the “three Ds”). More recently, however, the need to use such systems within demanding, distant, and distributed missions has also been established [6]. This is a result of their increasing capability and desirability in terms of reduced cost and increased safety. Offshore energy field operations, such as unmanned inspection and maintenance of oil platforms and wind farms, are prime examples of these latter “three Ds”. This makes unmanned aerial autonomous systems strong candidates for these missions.

In all environments, but in particular in hazardous environments, autonomous systems must operate safely and be safe to operate. Operating safely in this context means that the system should not carry out acts or behave in a manner that would be considered to be hazardous, such as colliding with an asset or a person. An autonomous system that is safe to operate is one where operating procedures and/or onboard software/systems have been developed to minimize the risk of the system causing damage or harm. This paper illustrates a method intended to contribute to both of these aspects of safety.

One means to ascertain where it is safe to operate the system would be by the definition of an operational envelope for the offshore asset-autonomous system combination in question (e.g., UAS and oil rig), inside of which the confidence in the safety of the mission is, in some sense, “high”, although perhaps not “guaranteed”.

In one sense, this is not a new problem. A closely related task is the clearance of aircraft operating onto and from a ship. Here, aerial vehicles are operating in the presence of bluff body structures that give rise to local unsteady flows which can be hazardous to the operation. In the UK military, this maritime problem is addressed as follows. To provide helicopter operators with guidance as to which environmental conditions are safe for take-off and landing operations from ships, a ship–helicopter operating limit (SHOL) is created for each helicopter–ship combination [7]. This is generally constructed via what are known as first of class flight trials (FOCFTs). The SHOL indicates to the ship/helicopter operator both the wind speeds and directions (relative to the ship) for which it has been demonstrated that helicopter operations are safe to carry out to that class of ship during the FOCFT. The larger the SHOL envelope, the greater the operational capability of a specific helicopter type operating from a given ship. An example of such an envelope can be seen in Figure 1.

First of class flight trials are performed at sea. They are, inevitably, expensive and it can typically take weeks to construct a full SHOL envelope. The full range of wind and sea conditions may not be available during the trials, resulting in the published SHOL being more conservative than would otherwise be necessary [8].

In addition to the operational piloting limits being established during the FOCFT, pilots use the Deck Interface Pilot Effort Scale (DIPES) [9]. This is a 5-point rating scale where the test pilot awards a rating for the landing. It is based on the amount of effort the pilot has to expend to remain in control based on workload (or pilot compensation), performance, accuracy, and consistency. The pilot then makes a subjective assessment of the landing as to whether or not the average pilot would be able to make the landing. This assessment then appears in the DIPES scale as a numerical value, where a 3 or less indicates a safe landing. A rating that is higher therefore indicates the contrary. The SHOL

envelope thus indicates the transition in environmental conditions between the safe and unsafe landing of the helicopter to a particular ship.

As per the oil and gas sector, the global military community is under pressure to reduce the costs of their operations. To try to achieve this for the FOCFT, they have turned to modelling and simulation [10–12], not as a means to completely replace at-sea trials, but as a means to inform the most effective conditions for live testing. In its most common form, the use of modelling and simulation for SHOL development has been the use of pilot-in-the-loop testing to derive helicopter/ship operational guidelines and to construct preliminary simulated SHOL envelopes [13,14]. Of particular note, piloted flight simulation was used extensively to prepare for the F-35B Lightning II FOCFTs on the UK's new aircraft carrier, HMS Queen Elizabeth [15].

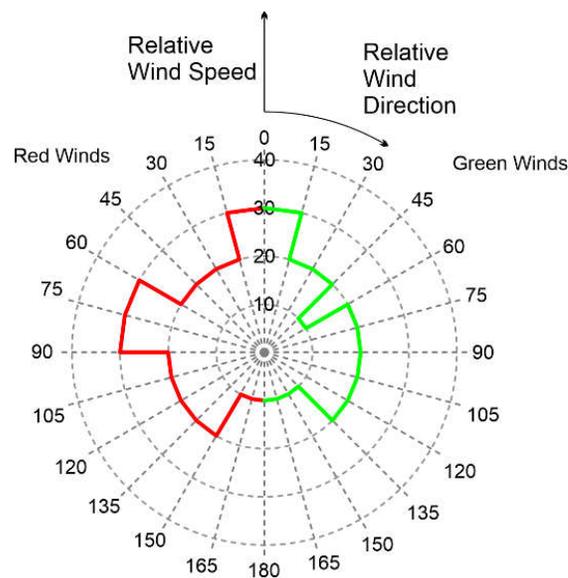


Figure 1. Example SHOL for a manned helicopter ship landing maneuver.

The autonomous systems problem, of course, differs in that there is no pilot on board to provide either the pilotage or the subjective feedback as to any deficiencies encountered that might provide information pertinent to the construction of an operational limit. This has started to be addressed, for example, in [16] where a pilot model is used. Whilst the model developed was demonstrated to show that it could predict a SHOL accurately, the limitation of this approach is that no equivalent to the DIPES rating could be provided, i.e., it was not able to diagnose why the limit was reached.

For an autonomous system to be used in a real-world environment, the safety of its operation needs to be agreed with the regulator of that environment [17]. In the UK, there is not yet a standard method to assess whether or not an autonomous UAS operation is safe. Each request for a particular operation is reviewed by the Civil Aviation Authority (CAA) on a case-by-case basis, using a submitted safety case/risk assessment for the planned operation [4].

To demonstrate this safety case, for an autonomous system, therefore means that the decisions being made by the system, the reasons why they have been made, and the actions that result from these decisions need to be verified for all practical operating conditions. For a safety case, this must be demonstrable and underpinned by evidence. This evidence also needs to be understandable, holistic, and repeatable. It is argued that a system is not verified, or proven to be safe, if only a few physical or simulation tests have been carried out; or if only part of the system's autonomy has been tested. It is expected that a rigorous and systematic method for assessing the autonomous behavior is required.

It is, however, clear that if the tools and techniques developed for and implemented by the manned aviation SHOL testing community (both simulated and live) can be adapted to the needs of the autonomous oil and gas sector, then there is the potential for it to assist in achieving the goal of cost savings whilst maintaining safe operational capability. This paper describes the first attempt at the development of such a simulation system to demonstrate how this can be achieved.

Thus, the question asked in this paper is as follows:

Using manned SHOL simulation techniques as the inspiration, how can an autonomous UAS be analyzed to determine the conditions under which it fails and to also indicate why it failed?

The paper is arranged as follows:

- The simulation environment is described and followed by the method to analyze the response.
- The experimental setup, including the cases simulated and under what condition, is given.
- The resultant operating envelopes are shown for a single case and for a range of performance specifications.
- Extracted responses for a selection of points on the operating envelope are shown.
- The discussion and conclusion are given, drawing out the implications and future works.

2. Method

2.1. The Mission

For the purposes of this paper, “the offshore environment” means the environment around an offshore energy generation asset, such as an oil rig or a wind turbine. As for the manned case, a potentially significant source of operational difficulty for such tasks will be when flying in the disturbed/turbulent air flow near such structures, as characterized in Figure 2. A typical hazard might be where the vehicle does not possess sufficient control authority to maintain a desired position. This could lead to a collision with the asset or its associated personnel [17].

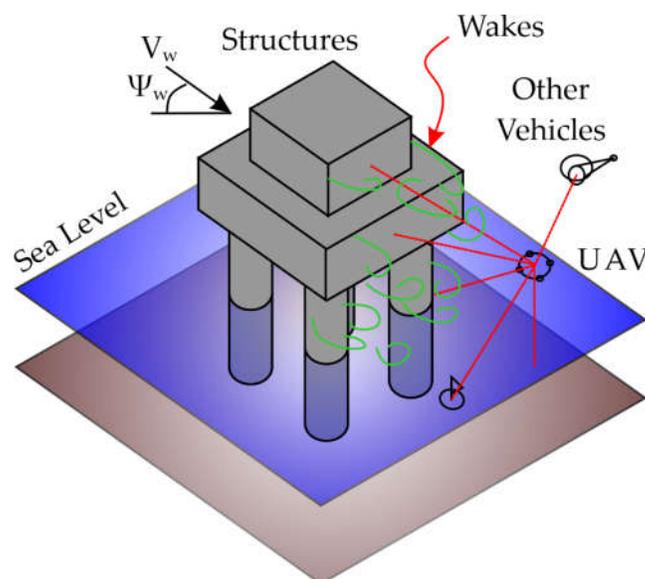


Figure 2. UAS operating environment for the selected mission scenario [17].

The mission scenario considered in this paper is some form of inspection task for the UAS on the leg of an oil platform. The UAS starts some distance away from the oil rig and is tasked with flying to a location near one of the legs. This task is sufficiently complex to allow the methodology to be rigorously tested and demonstrated for the purposes of this paper.

It is assumed that for this mission, just like for the manned case, a set of wind speeds and directions will exist in which the UAS will no longer be able to operate. The operator of the UAS, oil rig, and regulators would need to know the safe wind speed and direction operating envelope (OE) before any task could be authorized to proceed.

Of course, the advantage of using simulation is that many variables that could affect the safe operation of the UAS can be tested (given suitable simulation fidelity). Typically, these could include, but are not limited to:

- Initial mission position and goal;
- Geometry of the environment;
- UAS performance capability, e.g., turn rate, climb rate, bank angle, etc.;
- Actuator/sensor performance/degradation;
- Other environmental conditions, e.g., ambient light, sea state, etc. [17]

For the purposes of this paper, and for the sake of brevity, the study of this mission has been limited to a relatively small number of start and goal locations, a single, fixed environment geometry, but with varying UAS performance specifications and wind conditions (speeds and directions).

2.2. Simulation Environment

2.2.1. Simulation Architecture

The overall architecture of the simulation environment is shown in Figure 3. The sub-system groupings are shown in different colors. These are as follows:

- **Stitched Linear Vehicle Model (blue):** the air vehicle flight model used for the work described in this paper is a linear model derived from a more complex nonlinear one. To account for changes in vehicle dynamics throughout the flight envelope, these individual linear models are “stitched” together. This process is described in more detail later in the paper.
- **Stabilization and Tracking Controller (red):** this controller stabilizes the aircraft (a necessary function for the conventional helicopter configuration used) and provides a means for the vehicle model to follow the velocity and heading commands provided by the guidance system.
- **Guidance System (green):** this system calculates the required velocities and headings to follow the ground track received from the navigation system.
- **Navigation System (orange):** the navigation system creates a route for the aircraft to follow to achieve the mission, taking into account areas of risk (e.g., turbulent flow in the lee of the structure) and closest point of approach to the asset.
- **Mission Management (yellow):** in the form reported in this paper, this system is a fairly simple algorithm that decides when to switch to the next waypoint on the route and, once the final mission position has been achieved, when to bring the vehicle to a halt in a hover.
- **Dynamics (white):** this part of the simulation updates the state of the aircraft from the previous time step based upon the current control inputs.
- **Wind Model (cyan):** this element of the simulation updates the steady wind speed and direction to be used.

Each of these systems is described in more detail within the remainder of this subsection. The system was built to be modular, to aid future development and the inclusion of additional features, such as sense and avoid functions or more complex vehicle or environment dynamics. The definition of the notation and the axis systems used in Figure 3 can be found in Appendix A.

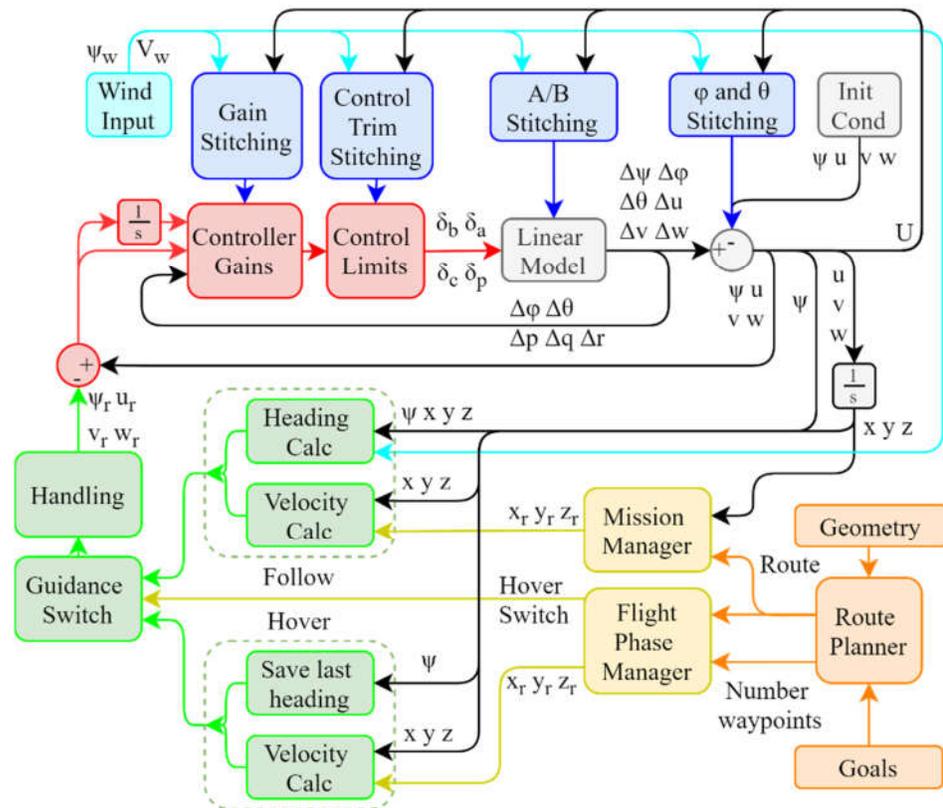


Figure 3. Simulation system architecture. Subsystems are grouped by color.

2.2.2. Vehicle Dynamics Linear Model

A wide range of unmanned aircraft configurations and airframes can potentially be used in offshore surveying tasks. For the initial development of the system reported in this paper, a conventional helicopter configuration aerial platform that was already available to the authors has been used.

The University of Liverpool maintains a large library of aircraft simulation models across several software packages, covering many configurations and scales. For this paper, the most appropriate aircraft model available was the Align T-Rex 700E. The T-Rex is a small aerobatic “3D” helicopter, typically used as a remotely piloted aircraft (RPA) for sport flying, see Figure 4. The T-Rex model was generated by appropriately scaling a version of an R-MAX model also available at the University of Liverpool. The R-MAX model has had limited validation against flight test data provided by the NRC [18].

The T-Rex has a 1.582 m diameter rotor with 2 blades, has a maximum takeoff weight of approximately 5.2 kg, and can be easily retrofitted with additional systems to enable autonomous operations. The T-Rex model used in this study was created using FLIGHTLAB, a state-of-the-art, component-based, selective fidelity modelling and analysis software package [19]. The nonlinear T-Rex model used to generate the linear models comprises a blade-element main rotor model with a Peters-He inflow model, a Bailey rotor for the tail rotor, and aerodynamic lookup tables to represent the forces from the fuselage. The aircraft also has a stability bar which is modelled as a rate feedback gain in the roll and pitch axes.



Figure 4. Picture of the Align T-Rex 700 E Helicopter (Picture by Mike Lehmann, 5 March 2007 (UTC), License—CC BY-SA 3.0 https://commons.wikimedia.org/wiki/File:Align_T-REX_2386.jpg, accessed on 19 February 2021).

FLIGHTLAB is specifically designed to enable real-time piloted simulation, but the models it produces can be computationally expensive to run. For the purposes of this paper, it was advantageous to simplify the flight model by linearizing it into a typical 9-state state-space aircraft model using FLIGHTLAB's built-in linearization analysis tools. The state-space model was then integrated into MATLAB/Simulink to enable rapid prototyping and testing in a multithreaded, faster than real-time environment. It is acknowledged that this linearization process necessarily eliminates some of the detail present in the full nonlinear FLIGHTLAB model, since all linear models are strictly valid for a particular operating point.

To partially mitigate this loss of modelling detail, a set of linear models were produced for a range of test points covering different flight speeds, wind speeds, and wind directions. The resulting state-space matrices were used as data points to interpolate between, to create an acceptable analogue to the nonlinear model even when operating conditions varied.

To obtain a linear model for a given flight condition, the nonlinear aircraft model was first trimmed on a northerly heading at a particular groundspeed, using the enhanced Newton trim method provided by FLIGHTLAB. The FLIGHTLAB linearization routine was then used to generate the relevant state and control (A and B) matrices for that condition. This procedure was followed for groundspeeds from -20 to 40 knots in 5 knot increments. A 5 knot wind from a 0° heading was then introduced at each ground speed and the trim and linearization procedure repeated. The wind speed was incremented in 5 knot steps until the trim procedure failed to return a stable trim point, and hence the edge of the wind envelope was found for that particular wind direction. The wind direction was then changed by 30° in a clockwise direction, and the wind speed again incremented to find the aircraft's trim limit for that environmental condition.

This procedure yielded a set of 1134 linear models to create a model space approximately bounded by the nonlinear model's trim envelope. The limits of that trim envelope for the T-Rex model for varying groundspeed are shown in Figure 5.

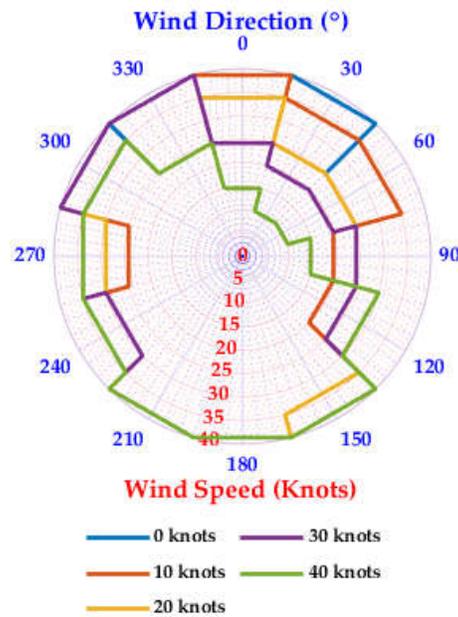


Figure 5. Trim envelope for the T-rex FLIGHTLAB model.

A comparison between the stitched linear models and the nonlinear models is made in Appendix B. It was found that for the on-axis responses, the linear models gave a good approximation to their nonlinear counterparts. For the off-axis responses, it was found that the agreement was reasonable while not as good as for the on-axis cases. This is a typical situation for helicopter modelling and simulation.

2.2.3. Vehicle Dynamics Stitching

To create the “full” vehicle dynamics model, each individual linear model was stitched to the next, based on their respective operating points. As stated previously, the stitching allows some of the nonlinear effects available in the FLIGHTLAB model to be approximated, without the extra computational effort of modelling the full nonlinear model, e.g., rotor blade aerodynamics.

To achieve this, the A and B matrices are organized into a 5-dimensional matrix, where 2 dimensions are for the row and columns of the A and B matrices and the other 3 dimensions are for the operating points. The operating points are forward flight speed in the horizontal plane, U , the wind direction, ψ_w , and the wind speed, V_w . This is shown in Table 1 alongside the ranges of operating points included.

Table 1. Operating points for model linearization.

Operating Condition	Range of Values (Min:Delta:Max)
Forward Velocity U	-20:5:40
Wind Speed V_w	0:5:40
Wind Direction ψ_w	0:30:330

The stitching is performed by interpolating each element of the matrices at query points. Further details about this process can be found in Appendix C. A cubic spline interpolation was used to ensure the gradients at the sample points were smooth.

One issue of note that arose with the stitching process was that of the need to deal with nonuniform grids. Since the A and B matrices are derived from the linearization of a

model that is trimmed at a set operating point, if the FLIGHTLAB model did not trim, then that operating point would be missing. Interpolation with nonuniform grids when close to the edge of the data set created some large gradients, overfitting, and large deviations from the expected value, due to missing data points for the interpolation. This was fixed by reconstructing the A and B matrices for the untrimmed operating points by fitting a surface to the trimmed points and extrapolating the untrimmed A and B matrices at the failed operating points. A check was then made to ensure the trim values for the controls were between their 0% and 100% limits. This prevented any anomalies at the boundary of the trim envelope whilst retaining realistic control trim values.

2.2.4. Stabilization and Tracking Controller Tuning

Conventional helicopter configurations are typically dynamically unstable in both hover and forward flight, [20] (except for yaw in the latter case). It has already been noted that the real aircraft on which the model is based has a mechanical means to try to artificially increase the stability of the airframe. However, in the absence of a pilot to provide further stabilization, a stability augmentation system had to be developed for the aircraft model. In addition, a tracking controller needed to be added to the model control system to allow it to “follow” the planned mission routes.

The required stabilization and tracking functions were developed using an LQR optimization method, augmented to allow the guidance variable to be tracked to be included (such that the gains for the tracking loop could also be tuned). The weightings required for the LQR tuning process were obtained manually via trial and error. Once the appropriate gains were achieved for each flight condition, they were gain-scheduled using the stitching process described in Section 2.2.3. Further details on the controller development can be found in Appendix D.

2.2.5. Guidance Command Algorithm

For the initial system proof-of-concept, the intent was to demonstrate that the system could follow a planned route in the presence of a steady wind component. The controller briefly described in Section 2.2.4 would therefore need to provide desired heading- and ground speed-following functions in the presence of this wind. Velocity and heading reference values are calculated using a path following guidance algorithm. The details of the algorithm developed can be found in Appendix E.

2.2.6. Vehicle Performance Limits

The maximum velocity of the UAS, V_{max} , was restricted by either the heading difference between actual and desired Figure 6a, or the distance to the waypoint Figure 6b; the more restrictive of the two was used. This prevented the UAS from moving until it had reached its desired heading and ensured that it reduced its speed as it approached a waypoint, to prevent overshoot.

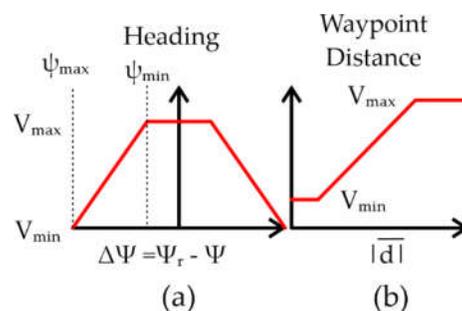


Figure 6. Velocity restrictions from heading difference and distance to waypoint. (a) heading restriction (b) distance restriction.

Restrictions were imposed upon the guidance commands to prevent accelerations and turn rates from becoming either too large and hence over controlling the system, and/or unrealistic for a vehicle of this size. This was imposed upon the values generated in two ways by the guidance system (see Section 2.2.5): first, saturation on the magnitude difference between the reference, u_r, v_r, w_r , and actual states, u, v, w (see Figure 7); second, a second-order transfer function to smooth the transitions from one state to another. This was done on both the velocity reference and the heading reference.

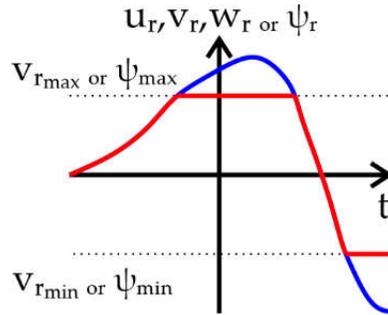


Figure 7. Restrictions to limit accelerations and turn rates.

2.2.7. Route Planning

For the guidance algorithm to function, a planned path for the vehicle model to follow had to be generated. To achieve this, the plan-view geometry of the oil rig was broken down into a grid using cellular decomposition. This resulted in a set of grid points around the legs of the platform and an adjacency matrix defining the connections between each of these grid points. The path planning for the UAS mission was then performed using an A* route finding algorithm [21], but where the heuristic was adapted to include a risk model to emulate the presence of an unsteady air wake. An example of such a risk model is shown in Figure 8. In this instance the risk was set to be analogous to the turbulence intensity of the velocity field and therefore the difficulty that the aircraft might encounter in maintaining controlled flight within it. The intent here is to eventually replace this heuristic risk model with one derived using computational fluid dynamics-derived wake and nonlinear simulations of the aircraft interacting with that wake. Further details of the cost function for the A* heuristic can be found in Appendix F.

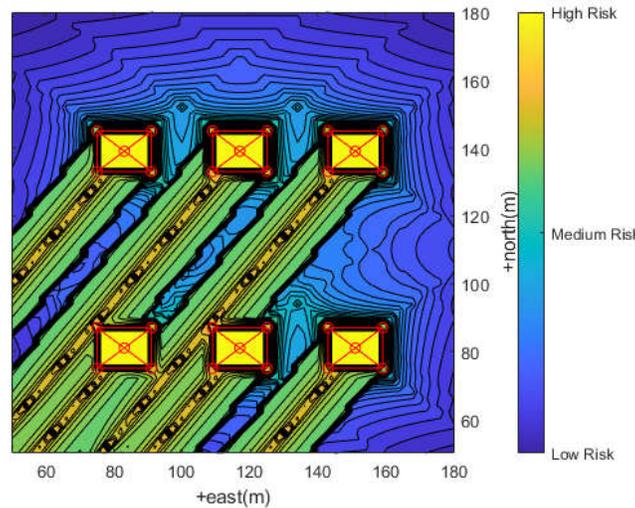


Figure 8. Contours of the risk map for NE wind. Risk is normalized between 0 and 1.

2.2.8. Mission Management

The mission is currently managed with a simple logic system. The intention is that, in future iterations of the simulation, these logic systems will be replaced by verifiable rational agents. The current waypoint is switched to the next waypoint when the UAS is either within a certain distance of the waypoint or has passed the waypoint and the next one is ahead of it. This prevents the UAS from overshooting a particular waypoint and then doubling back to try to reach it when the next planned waypoint is closer. The transition from waypoint-follow to hover is determined by the proximity of the UAS to the last waypoint and its velocity. This allows the UAS to slow to a near stop before trying to hold its current location.

2.3. Performance Metrics

Section 2.2 describes a fairly conventional simulation set-up from an engineering methods perspective. The main purpose of this work was to investigate how operational envelopes might be derived from it, both for operational design, clearance, and mission rehearsal purposes. This section describes how this was achieved in the current work.

2.3.1. Causes of Mission Failure

Inspired by the DIPES rating scale methodology, described in Section 1, it was envisaged that, as well as providing an estimate for the limit of operations of the UAS via an envelope similar to a SHOL, the nature of the failure that caused the limit for that flight condition should also be discoverable. The concept adopted is illustrated in Figure 9. This shows the results of a series of different hypothetical mission simulations showing which element of the system fails for different wind speeds and directions (Figure 9a). From these, a composite operational envelope can be constructed using the most restrictive failure mode per condition (Figure 9b). In order to understand the cause of the failure, however, a series of system-specific objective functions were required.

2.3.2. Objective Functions for Each Component

To be able to create the operational envelopes described in the previous section, three continuous components of the system were considered. For a real system, this would likely be many more, but three was considered sufficient, in this case, to prove the concept. The responsibility of each component in the mission determines the definition of the cost function. These are defined as follows:

- Actuator: to create the required control response output while leaving a margin of error as a contingency to allow reactions to unforeseen disturbances.
- Controller: to ensure that the current vehicle states follow the commanded states within a user-defined tolerance, defined as the difference between the actual and reference values, while maintaining controlled flight.
- Guidance: to cause the system to follow the desired path to within a desired separation distance.

A point to note here is that the above definitions are not meant to be the ultimate definitions of the subsystem, but rather an example of what the “user” may wish the subsystem to achieve. Different systems and applications may require different definitions of the subsystem’s role.

The idea of successful or failed missions for a given start/end point over all of the possible wind conditions is illustrated in Figure 9. In the manned aviation world, the center of the SHOL represents the target landing spot on the ship’s deck. For this work, the center of the plot represents the start point for the mission.

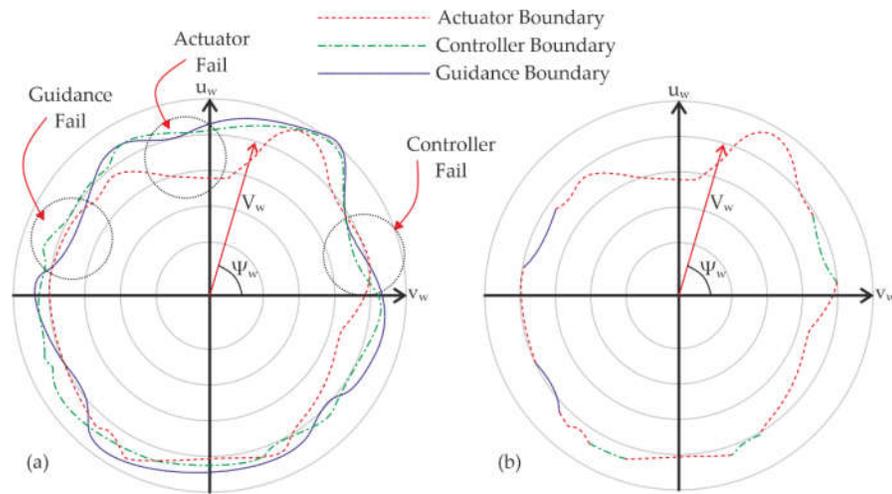


Figure 9. Illustration of how the subsystems can be combined and therefore allow the explanation of why a system failed to operate safely; (a) individual subsystems; and (b) envelope when limited by worst performing subsystem. [17].

The cost functions for each of the subsystems described above will be dealt with in turn. They are taken from a previously published paper by the authors [17]. The cost function for defining the actuator’s performance, F_A , is shown in Equation (1) and illustrated in Figure 10.

$$F(\bar{\theta}) = F_A = \max \left(\frac{1}{t_m} \int_0^{t_m} \sqrt{\frac{(\bar{x}_{actuator_i} - 0.5)^2}{0.5 - \bar{s}_{actuator_i}}} dt \right) \quad (1)$$

where n_A is the number of actuators, t_m is the segment simulation time, $x_{actuator_i}$ is the i th actuator output at time t , $\bar{s}_{actuator}$ the specified margin of error, and d_t the time step of the simulation.

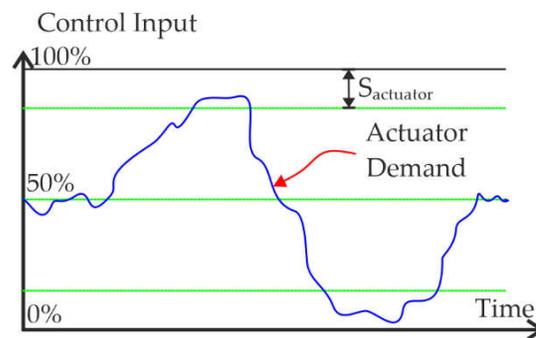


Figure 10. Definition of the cost function for the analysis of the actuator’s performance [17].

Here, the zero deflection point for the actuator is defined as 50%. This could also be considered to be the (unlikely) ideal case and is essentially a situation where no control input is required. The function is, in essence, a time average of the deviation from the midpoint normalized by the margin of error. The performance of all the actuators is averaged over time and the largest value from the four actuators is taken.

This function aims to create a single measure for all of the actuators over the time period of operation between 0 and 1. The cost function gives a gradual increase in the failure to meet specification. If an actuator reaches either 100% or 0%, this results in the failure to meet specification of the system being set to 1. This can be considered a critical failure since controlled flight beyond the actuators’ motion limits would not be possible.

The guidance controller’s performance, F_c , is defined in Equation (2) and shown in Figure 11.

$$F(\bar{\theta}) = F_c = \max \left(\frac{1}{t_m} \int_0^{t_m} \sqrt{\frac{(\bar{x}_{reference_i} - \bar{x}_i)^2}{\bar{S}_{controller_i}}} dt \right) \tag{2}$$

where n_r is the number of controlled states, $\bar{x}_{reference_i}$ is the command reference, \bar{x}_i the measured state of the system, and $\bar{S}_{controller_i}$ the specified maximum difference between the actual and reference values.

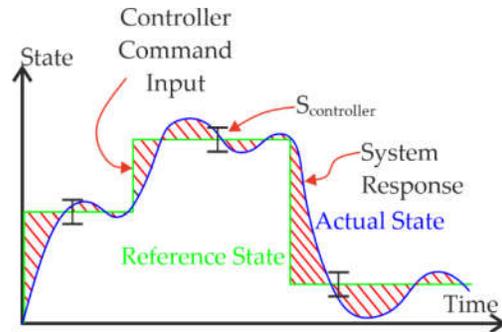


Figure 11. Definition of the cost function for the analysis of the controller’s performance [17].

It is essentially the same as the cost function used in linear quadratic regulator controllers. The difference between the reference and controlled state is normalized by a desired maximum distance. It is then averaged over both time and the largest failure to meet specification taken. A discontinuity exists when the system becomes unstable.

Lastly, the guidance performance, F_G , is defined by both Equation (3) and Figure 12.

$$F(\bar{\theta}) = F_G = \frac{1}{t_m} \int_0^{t_m} \frac{d_g}{\bar{S}_{guidance}} dt \tag{3}$$

where d_g is the perpendicular distance from the actual position to the desired path and $\bar{S}_{guidance}$ is the specified maximum deviation from the path.

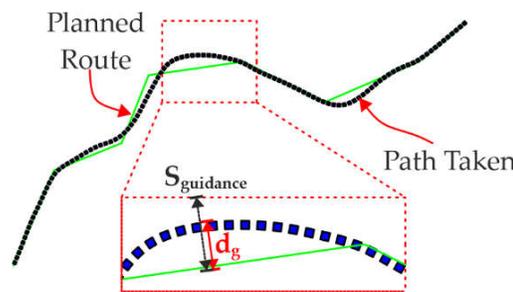


Figure 12. Definition of the cost function for the analysis of the guidance performance [17].

This guidance performance measure is the length of the vector perpendicular to the nearest point on the desired path from the system’s current location normalized by the specification, the desired maximum deviation from the path. A discontinuity does not explicitly exist for this function.

2.3.3. Segmenting the Mission

The methods of the previous subsection could be applied to the entire mission. However, because of the nature of the cost functions, it is easy to conceive of situations where false positive or negative results would be provided as a result of the poor performance

being averaged out. A simple solution to this was to break the mission up into shorter elements, in this case the segments between the waypoints. For a mission with only straight elements, as opposed to a mission using Dubins curves, which have straight and curved elements, this results in two types of segments, straight and turning, see Figure 13. This results in the straight segments overlapping the turn segments and turn segments overlapping each other. The performance metrics are then applied to each segment in a mission and the operating limit of the mission is given as the segment with the worst performance.

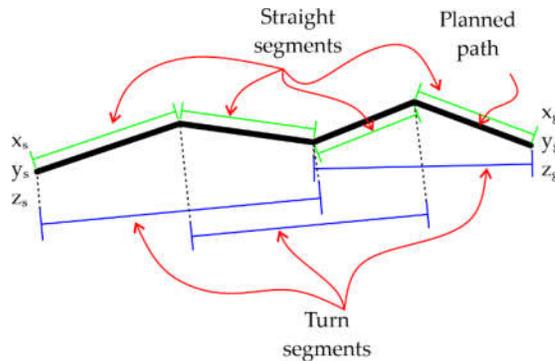


Figure 13. Segmenting of mission to allow the analysis of distinct sections.

2.4. Experimental Test Matrix

With the simulation environment set up as described in Section 2.2 and the means to measure the operation limits of the air system in place, as described in Section 2.3, the concept was tested using the mission described in Section 2.1. The possible instances of the missions tested are shown in Figure 14. Possible start locations of the inspection task are shown in green and the final inspection hover points in red. The black boxes represent the legs of the oil rig. The UAS is tasked, for a given start location, with planning a route and flying safely to a designated inspection point (one of the red dots). This must be accomplished for a range of (steady) wind conditions (speeds and directions). The geometry is based on an open source example of an oil rig geometry [22], where the measurements in Figure 14 are specific to this geometry.

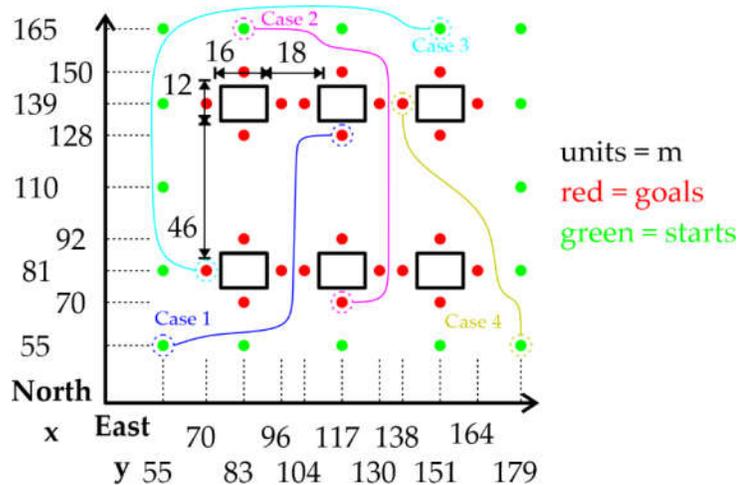


Figure 14. Plan view of the environment with the start and goal locations highlighted in green and red dots.

The conditions tested are shown in Table 2, where the range, calculation step increments, and number of variations are given. A total of 2880 combinations of wind speed and wind direction were tested. Simulations comprising 4 different combinations of start and goal locations were attempted. These are shown in Table 3. The 2880 variations of the wind speed and wind direction were tested for each start and goal combination. This resulted in a total of 11,520 simulations of the UAS flying around an oil rig under a range of environmental conditions.

Table 2. Variables tested for each task.

Variable	Range	Step	Total Variations	Total Combinations
Wind Speed (knots)	$5 \geq V_w \leq 40$	$\Delta 1$	40	2880
Wind Direction (deg)	$0 \geq \psi_w \leq 355$	$\Delta 5$	72	

Table 3. Table of the four cases shown in the paper.

Case Number	Start	Goal	Num Sims
1	[55 55]	[128 117]	2880
2	[165 83]	[70 117]	2880
3	[165 151]	[81 70]	2880
4	[55 179]	[139 138]	2880
		Total	11,520

The performance specifications applied were varied to determine the sensitivity of the different specifications and to make sure that the results were consistent with the idea that more strict specifications should result in a more restrictive operating envelope. The variations in the specifications are shown in Table 4.

Table 4. Sensitivity study performance specification variations.

Performance Specification	Range	Step	Total Variations
Actuator (%)	$25 \geq \bar{S}_{Actuator} \geq 5$	$\Delta 5$	5
Controller Heading ($^\circ$)	$0.75 \leq \bar{S}(1)_{Controller} \leq 1.75$	$\Delta 0.25$	5
Controller Velocity (m/s)	$0.25 \leq \bar{S}(2:4)_{Controller} \leq 0.45$	$\Delta 0.05$	5
Guidance (m)	$1 \leq \bar{S}_{Guidance} \leq 5$	$\Delta 1$	5

3. Results

This section presents the key results of the simulation exercise described above. It is, of course, not possible to sensibly show all of the results for all of the test points. As such, where necessary, for a selection of the cases of Table 3 above, results achieved with the performance specifications defined in Table 5 will be used.

Table 5. Selected performance specifications.

$\bar{S}_{Actuator} = 15\%$
$\bar{S}(1)_{Controller} = 1.25^\circ$
$\bar{S}(2:4)_{Controller} = 0.35 \text{ m/s}$
$\bar{S}_{Guidance} = 3 \text{ m}$

3.1. Simulation Performance

The specification of the PC used for the simulations is given in Table 6. The simulations were run using MATLAB's parallel computing toolbox using 32 workers. Simulations, including the start-up and data saving steps, took on average ~4 s each to complete. Therefore, for the results shown in this paper a total of ~12 h 48 min simulation time was

used. For the analysis and postprocessing of the results, it took 11 min to apply the cost functions and generate the operating envelopes for a given specification.

It should be noted that the total simulation time is dependent on the batch size, the number of workers, and the range of conditions tested. Smaller batch sizes have a higher proportion of computational overheads, a smaller number of workers results in a lower CPU time being used, and for some conditions, the missions are naturally longer based upon the planned route geometry.

Table 6. Simulation PC specifications.

Processor	AMD Ryzen 9 3950X
Cores	16
Speed	3.5 GHz
Logical Processors	32
RAM	Corsair 32GB Vengeance LPX DDR4 3200MH
Memory	64 GB

3.2. Route Planning Results

Figure 15 illustrates the basic output of the route planner, i.e., all of the planned routes for Case 3. The green lines show where the system has achieved mission success, whilst the blue lines show the missions where results were discarded from multiple simulations; these colors correspond to results shown in Section 3.3. The black arrow indicates the straight-line direction of travel from the mission-start to the mission-end location, ignoring any obstacles in the path, and corresponds to the arrow that appears on subsequent plots. This figure is shown to illustrate the number of missions tested and then used to form an Operating Envelope for a given case. For the purposes of discussion, four test points from Case 3 have been selected to illustrate the method's results. The wind conditions for each of these test points are shown in Table 7.

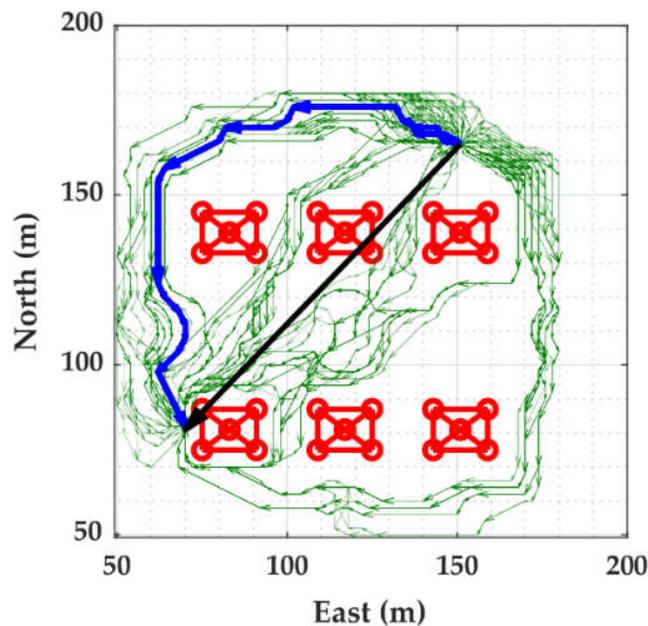


Figure 15. Bird's-eye-view of planned routes for Case 3.

Table 7. Conditions for the selected responses shown.

Test Point	Wind Speed	Wind Direction
A	7	30
B1	6	10
B2	7	10
C	8	330

The plan view and routes taken for the three selected test points are shown in Figure 16. To aid ease of reference, the colors of the lines are consistent with the same conditions through the plots in Section 3.4.

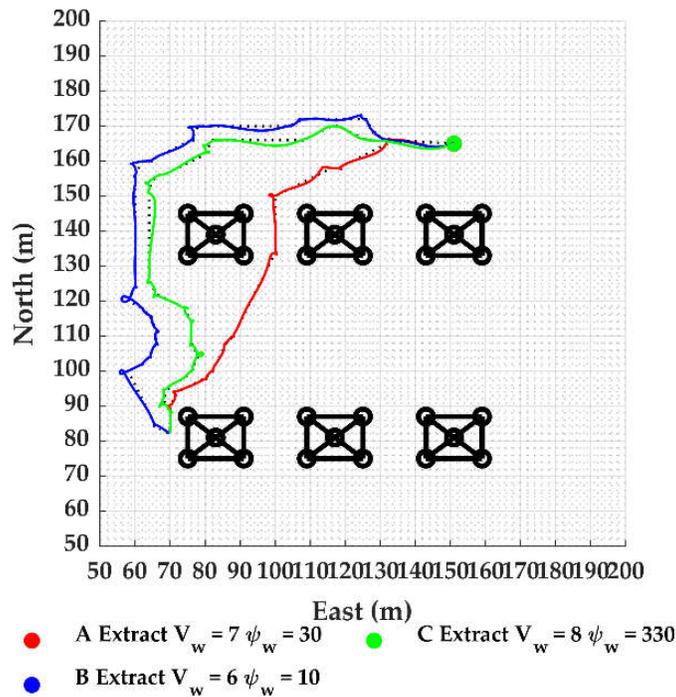


Figure 16. Bird's-eye-view of the extracted results.

3.3. Simulated Mission Success or Failure

Figure 17 shows a typical output for one of the test cases listed in Table 3. The dots represent all of the test points attempted:

- The green dots represent the contiguous sets of wind conditions for which the mission was completed successfully, i.e., all performance specifications were met.
- The gray dots represent the test points where the mission “failed”; a failed simulation means that the UAS encountered a situation, or was attempting a maneuver, that resulted in a response of the system that exceeded one or more specifications.
- The blue dots are interesting in that they represent test points that resulted in mission “success”, but that were discarded because there was a gap between those points and the contiguous set of successful mission simulations. It has been assumed that the envelope for a given wind direction must have an unbroken, contiguous set of wind speeds from the zero wind speed condition to the edge of the envelope. This naturally removes the possibility of exclaves of successful simulations or enclaves of failed simulations within the envelope. It also removes possible gaps in the envelope if viewed along a line of increasing wind speed; for example, at a wind direction of 240° in

Figure 17. It would not be a safe operation if it had to rely on the wind speed not dropping below a particular value for the mission to succeed.

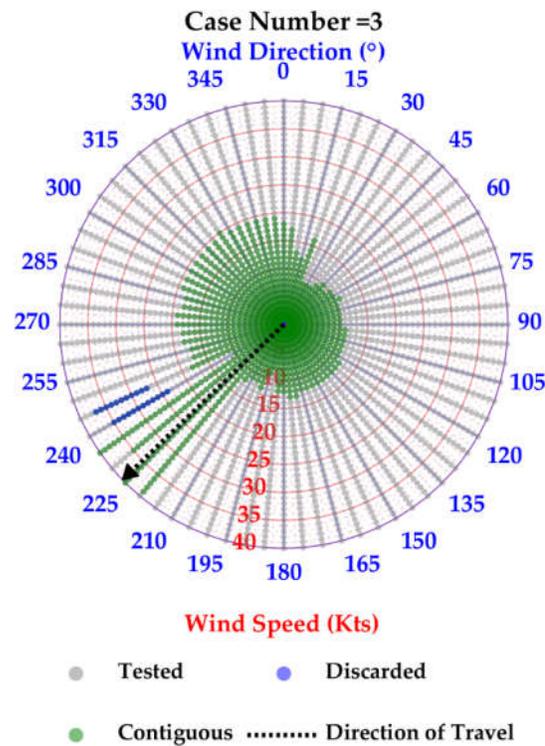


Figure 17. Full test matrix with failed and successful missions.

The straight-line direction of travel between the start and the end point is illustrated by the black dotted line and arrow. So, for example, in Figure 17, the “average” direction of travel between the start and the end point is south westerly. Note that, as is common practice, the wind direction indicates where the wind is coming from. As such, the wind direction is always pointing towards the center of the circular diagram.

3.4. Test Point Time History Data

This section presents time histories from each of the test points A, B1, B2, and C to contextualize the predicted operating envelopes shown later in Section 3. They are presented to demonstrate how a failure in a subsystem identified in the envelope manifests itself within the full operating envelope of the system. These test points were selected from the results in Case 3 because they represent failures of different systems.

Test point A was highlighted as the operating limit where the failure subsystem was the actuators. The demands made of the four controls of the helicopter are shown in Figure 18. The solid black line denotes the maximum and minimum actuator demand available and the dotted black line denotes the specified performance limit. The vertical gray dashes show the different segments that the cost functions have been applied to. It can be seen that all of the control actuators are within the specified limits except the lateral control actuator. For the lateral control actuator, the performance specification limit is exceeded between 60–70 s into the mission. Despite the exceedance appearing to be small, it should be noted that this is the defined edge between failure and success. The actuator has failed to satisfy the specification whereas, for this test point, all other specifications for all other systems have been met. This information is useful to the user as they may decide to relax the specification to increase the available envelope or simply live with the restriction.

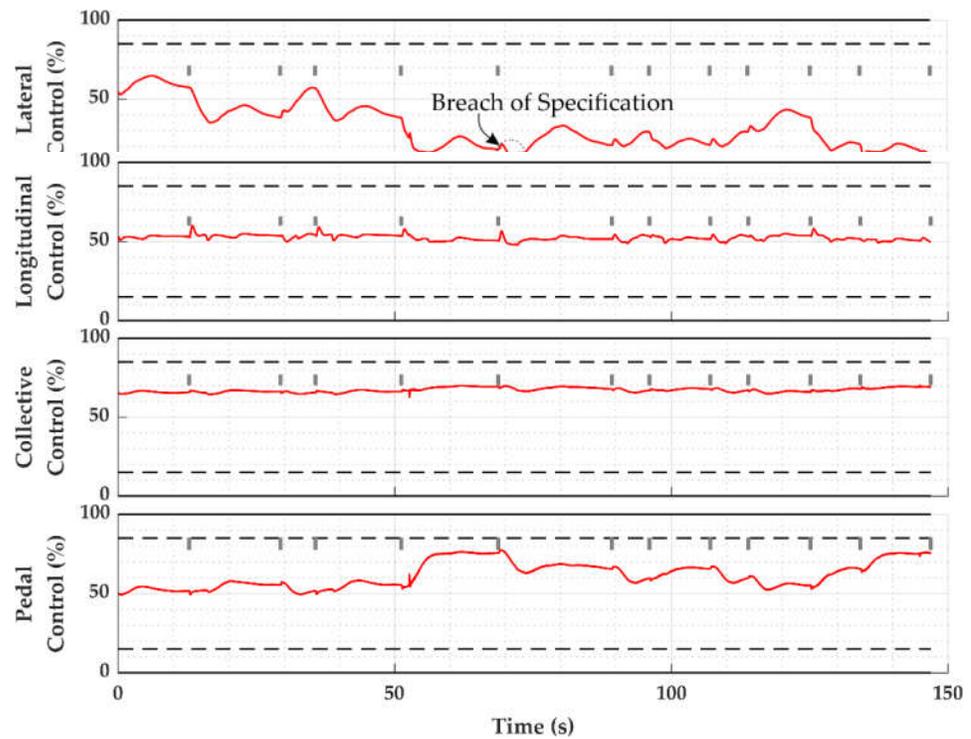


Figure 18. Control demands for the actuators for test point A, $V_w = 7$ $\psi_w = 30$.

A guidance failure was flagged for test point B1 in the simulation and the relevant parameters are shown in the top half of Figures 19 and 20. The deviation from the path clearly exceeds the specified allowed deviation at numerous points during the mission. Conversely, the proximity to a nearby object specification only comes close to being exceeded near the end of the mission where the UAS is reaching an inspection point close to the leg of the oil rig.

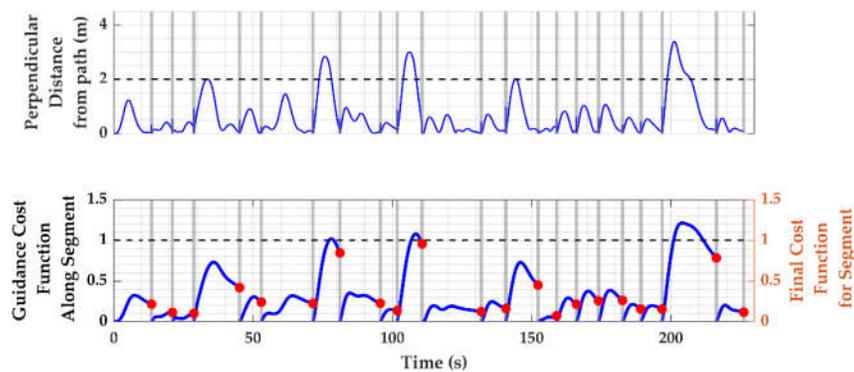


Figure 19. Deviation from path for test point B1, $V_w = 6$ $\psi_w = 10$.

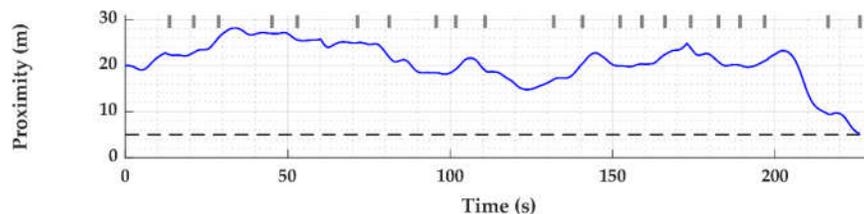


Figure 20. Proximity to object for test point B1 $V_w = 6$ $\psi_w = 10$.

Test point B1 is on the edge of the envelope, which is the conditions under which the system is about to fail to meet specifications. As a comparison, the same plot is shown in Figure 21, but where the wind speed has been increased by 1 knot. The cost function as it is calculated along a mission segment (blue line) and the final evaluation of the cost function for a mission segment (red marker) are shown in the bottom half of Figure 19 and Figure 21.

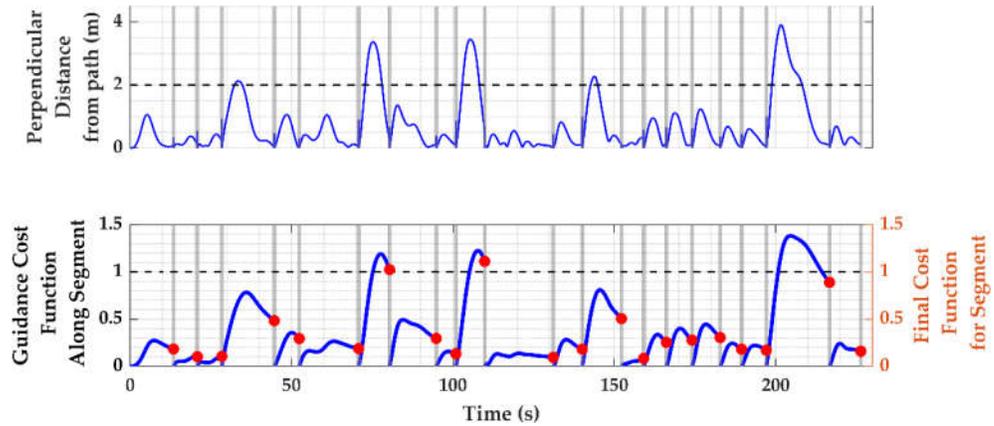


Figure 21. Deviation from path for test point B2, $V_w = 7$ $\psi_w = 10$.

The controller was flagged as the failing system in test point C. The controller tracking errors are shown in Figure 22. It is clear that for the heading error, lateral error, and forward velocity error, the specified performance has been exceeded on numerous occasions.

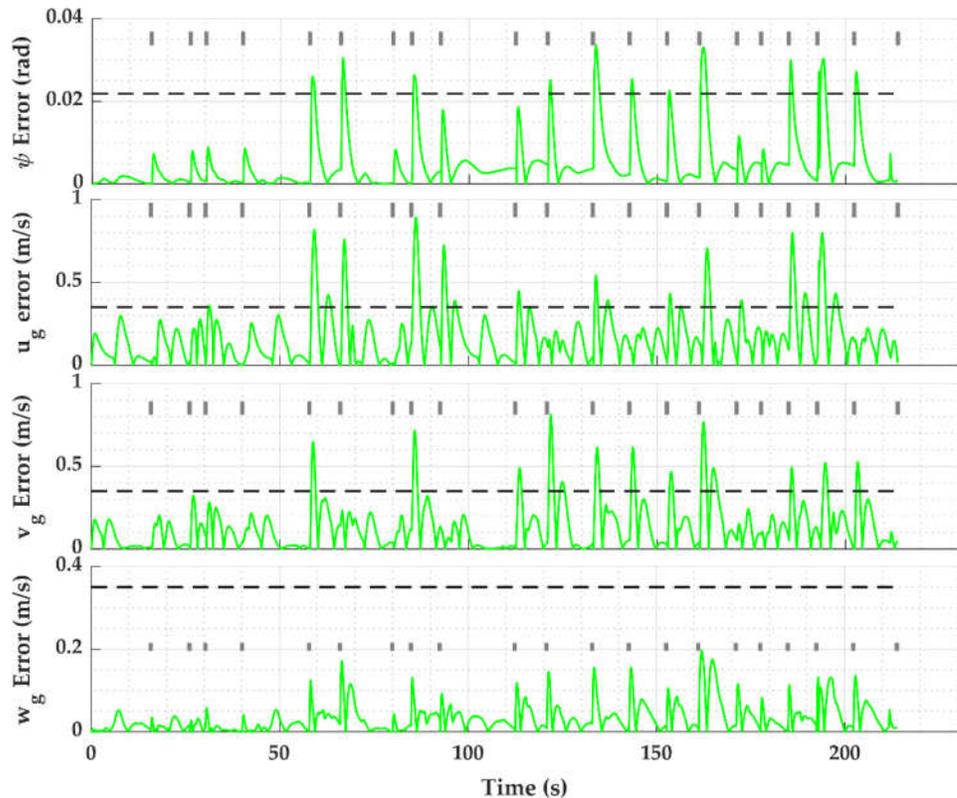


Figure 22. Tracking controller errors for test point C, $V_w = 8$ $\psi_w = 330$.

An important thing to note is that small breaches in performance are allowable in the method developed, should this be desirable for the user. This is to allow flexibility in the analysis and to not create a hard limit where one is not necessary. The exception to this is the proximity and stabilization performance metrics. For the former, a hard limit must be employed as breaching the limit is equivalent to a collision between the aircraft and the asset. For the latter, if a limit is reached, the aircraft would become unstable, potentially resulting in a loss of the airframe.

3.5. Operating Envelope Creation

The previous section showed the results for three separate individual missions, each with their own wind speed and direction. This section shows the operating envelope that is generated by combining the results of all of the wind conditions tested, including those already shown.

Individual subsystem envelopes are superimposed on one another in Figure 23, for a case using the specifications of Table 5. It can be seen that, for a given set of specifications, different subsystems provide the most restrictive operating envelope for the range of wind conditions tested.

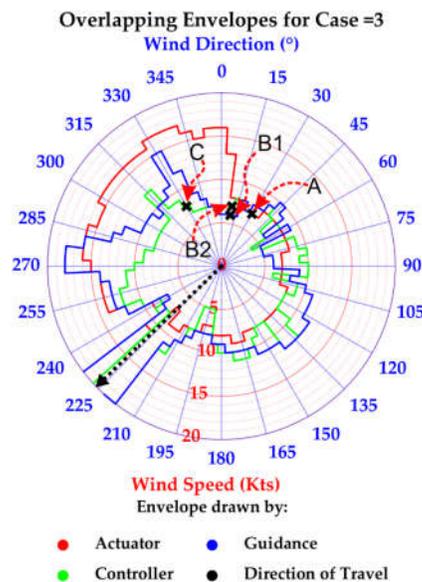


Figure 23. A focused plot of the envelopes for each subsystem for a given performance specification. Selected conditions shown as Xs.

Figure 24 then shows the “final” envelope that results from plotting the most restrictive subsystem limitation, irrespective of which one caused it. The black crosses on these two figures indicate where on the envelope test points A, B1, B2, and C from Section 3.3 correspond to. Figure 24 is the net result of the process described in this paper, where all of the information in the previously presented response plots for every wind condition has been distilled down to a single, hopefully understandable plot.

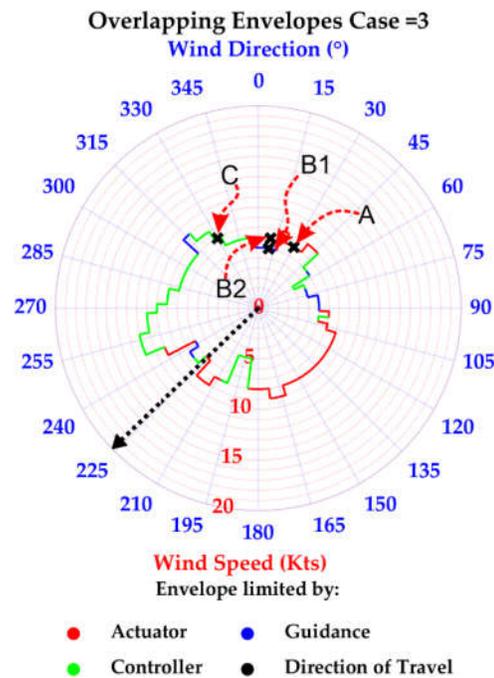


Figure 24. The envelopes for Case 3 if only the lowest performing subsystem is drawn. Selected conditions shown as Xs.

A key element of the proposed process is for the easy identification of the subsystem that is the cause of a mission failure for a particular condition. This might be useful, for example, at the design stage of such a system, by allowing the designer to explore solutions to maximize the operational envelope of the vehicle.

A feature of note is the “peninsula effect” that occurs in some of the plots for a particular wind condition. This is a simple result of the direction of travel, largely aligning with the wind direction for a large proportion of the flown missions. The UAS is more capable flying into the wind than it is across the wind, for a given ground track. Since the planner takes into account all of the possible locations of an air wake via the dummy risk model, and therefore tries to plan around it, each dot in Figure 17 can result in a slightly different route being planned. The range of routes planned for each case is shown in Figure 15. Therefore, the “peninsulas” occur when the “average” planned route happens to align more closely with the wind direction. A slight change in the wind direction may mean that an entirely different route plan is selected, with different combinations of into- and crosswind segments. This may have a segment that causes the UAS to fail to meet one or more of the performance specifications.

3.6. Performance Specification Sensitivity Study

Figure 25 shows sample results from the investigation into the results obtained for a variation in the values of the subsystem specifications used. The specifications used for the middle value are given in Table 5, where the increases or decreases in its value are given in the legend of the plots. The sensitivity of the envelope to the specification can be clearly seen. The strictest specification results are shown in red. This varies through green, blue, and pink, with the least restrictive specification results being shown in cyan. As would be expected, in general, a more restrictive specification leads to a more restrictive operating envelope limit.

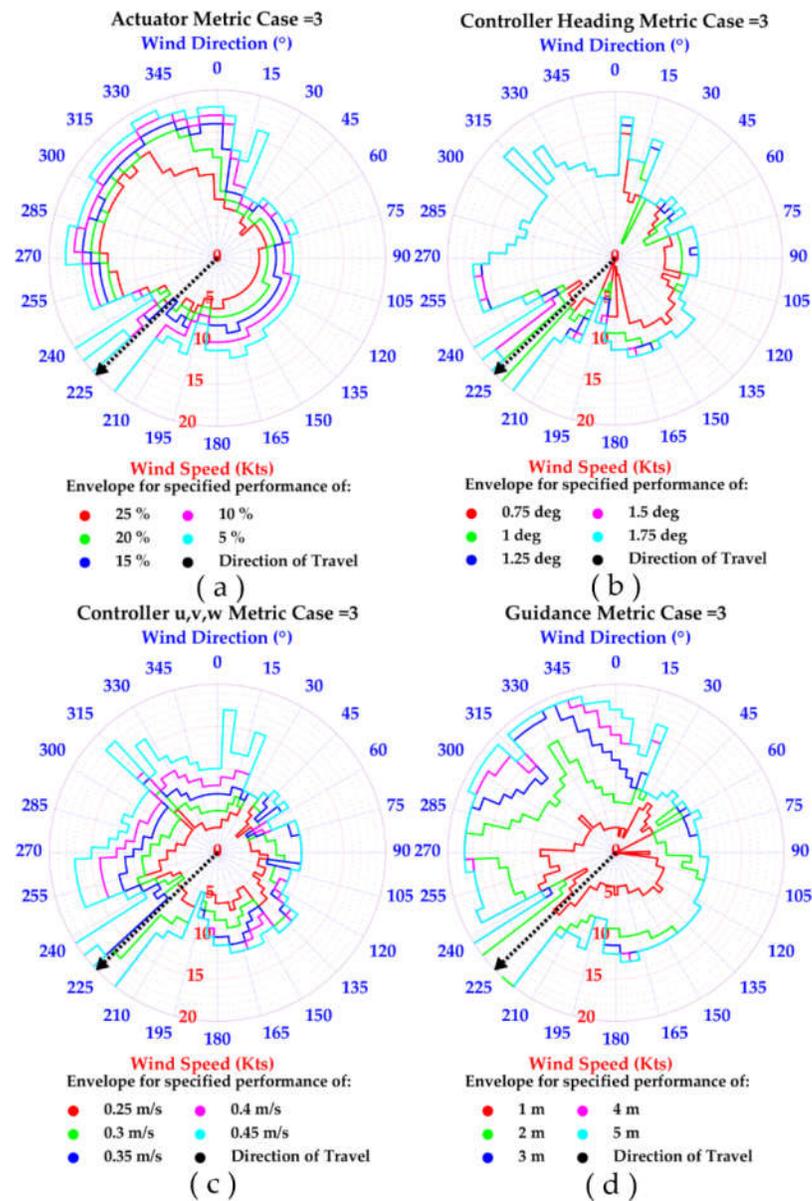


Figure 25. Variations in operating envelopes resulting from a range of performance specifications for the (a) actuator (b) controller heading (c) controller velocity and (d) guidance

4. Discussion

4.1. Helicopter Operating Limit Envelope for an Offshore Asset Inspection Mission

The activity presented in this paper set out to investigate whether or not manned SHOL simulation techniques could be applied to UAS autonomous operations to demonstrate an inspection mission’s operating limits and the reason that the limit is achieved. Figure 24 illustrates that this is the case. For the selected mission, Figure 24 presents an “at-a-glance” method either for a UAS designer to understand which subsystem(s) might need to be improved to increase operational capability or for a mission planner (either human or silicon) to make a “go” “no-go” decision either before or during a particular flight (assuming wind conditions are known).

Upon inspection of Figure 25, it is clear that a large range of envelopes for each subsystem can be drawn, depending on the performance specifications used. For the actuator

and guidance, reasonably large changes in the specification of the subsystem cause reasonable large changes in the envelope. However, for the controller, the range is far smaller for the heading and velocity tracking. This implies a high sensitivity to the specification. Knowledge about the operating limits' sensitivity to the various subsystem specifications is, of course, useful information. An operating limit sensitivity to a particular performance specification implies that effort must be expended on setting the specification correctly and ensuring that the simulation fidelity for that subsystem is sufficiently high to generate confidence in the results achieved.

It has been previously noted that the simulation system has been built in a modular fashion to allow for more complex components to be substituted for the simpler ones currently in use. For the system presented in this paper, the decision-making capability was very primitive. The intention is to replace these with, for example, rational agents. The rational agents can be formally verified using techniques such as those described in [23,24]. The techniques presented in this paper thus find a place between the formal verification of autonomous system decisions, for example, that the correct/safe decisions are made given what the vehicle perceives about its world, and physical testing of real systems in a practical environment. It is argued that the techniques described here will provide a much more realistic assessment of operating boundaries than we see with aggressive abstractions used in formal methods, while providing more confidence (across a range of parameter changes) than we see during individual physical tests.

The work presented in this paper is a proof-of-concept and, as such, subject to a number of limitations, which will be discussed in what follows.

4.2. Limitations

Whilst the paper presents the desired operating limit for a particular mission alongside the limiting subsystem, it is based upon simulations that use stitched linear models operating in steady wind conditions. The veracity of the results will be dependent on how representative both the vehicle and environment models are of their real counterparts. For example, the environment around such an offshore asset is unlikely to be "steady". Unsteady perturbations need to be included in the simulation environment to assess their effect. Beyond this, to increase confidence in their validity, the simulation results would need to be validated using real-world testing.

The operating envelopes presented in this paper represent a "brute force" approach to the problem, i.e., all test points in the test matrix are "flown" and their success or failure measured. The large number of "mission failure" test points shown in Figure 17 implies significant wasted computation time. Similarly, the benign cases well within the operating envelope represent cases that need not necessarily have to be tested. The areas of interest are the conditions around the envelope itself, the edge cases. Therefore, a strategy for searching for only the edge cases would reduce the time taken to find the envelope.

The results presented in this paper are task-, vehicle- and offshore asset-specific. The method, however, is task independent and therefore applicable to other vehicles, scenarios, and offshore asset geometries. This, in part, mirrors the manned aviation world where the SHOL is specific to the ship and the helicopter. There is also a specific maneuver that is performed to land the helicopter on the ship, for example. The difference between the offshore mission presented here and the helicopter-ship operation is that the complexity is increased for the offshore case given that there are many more possible start and end locations. For practical (and computational time) purposes, there may need to be an operational procedure that limits the possible missions to a single start location. There may also only ever be specific locations that need to be inspected, again, limiting the number of possible mission end locations. All of this would limit the number of simulations that would need to be conducted and hence the time required to obtain an operating envelope.

4.3. General Application

The system presented in this paper is specific to a particular mission but the architecture is generic. For a different mission, the problem geometry would need to be changed such that the planner could plan routes around different structures and obstacles. If the sensor platform were to change, then the vehicle model would need to be changed, alongside the model of its guidance and control system. The allowable performance specifications might also need to be changed to suit the vehicle/ mission/ operator tolerance to risk and so on. The intention of the paper is to try to illustrate the potential of modelling and simulation in helping to create safe unmanned system operations through a specific example. It is left to the reader to take this methodology and apply it to their specific needs.

5. Conclusions and Future Work

5.1. Concluding Remarks

A modelling and simulation approach has been presented to address the problem of assuring safe operation of an autonomous UAS performing an inspection mission around an offshore energy asset. The following conclusions can be drawn:

- Using both real and simulated manned helicopter–ship operations as an inspiration, operating limit envelopes for the specified vehicle/offshore asset combination can be generated.
- The operating limit envelopes produced provide a condensed, intelligible information regarding the capability of the UAS in the presence of the offshore asset and its corresponding wake. They provide useful information for both system designers and inspection service operators (and onboard flight management systems) alike.
- The use of operating limit techniques for the offshore inspection mission are more complex than the manned helicopter-to-ship case due to the larger number of mission profiles that could potentially be flown.

The work described in this paper provides a useful start point for assessing the operational safety of a UAS performing an inspection mission around an offshore asset. However, the method needs to be developed further, as described in the following section.

5.2. Future Work

5.2.1. Nonlinear Dynamics of UAS

Quasi nonlinear helicopter vehicle dynamics are used in the current simulations, where the nonlinear effects are partially included through the model stitching process. However, the effect of the linearization on the size and shape of the resultant operating envelope is not clear and needs to be established. This can be achieved by replacing the stitched linear model with the full nonlinear model from which it was derived.

5.2.2. Unsteady Wind

The current simulation uses a steady wind component in combination with a risk model to represent the unsteady wake in the lee of the structure. The manned aviation world has experimented with the use of inserting an unsteady wake model into the simulation environment. Repeating this for the unmanned case will be a useful next step for this work.

5.2.3. Search Algorithm to Find Operating Limit

As previously noted, the gray dots of Figure 17 represent wasted computational effort. This is especially true if a single “mission failure” simulation is computationally expensive or there are many possible combinations of failure conditions. This implies that a search algorithm to only look for and around the edge cases is needed. The cost functions presented in this paper allow a parameter space to be formed. This parameter space can then be searched for the edge cases and thus allow the operating envelope to be found

more quickly. The algorithm would not only need to be able to search through multiple dimensions, but also to search for multiple maxima/minima. The operating envelope is not only defined by a single point in the parameter space, but by multiple points. It is posited that a nature-inspired algorithm would be applicable to this type of problem due to their flexibility and robustness when applied to multivariate problems.

5.2.4. Optimization of Autonomous System Parameters

Since the area of the operating envelope can be considered to be the maximum performance of the system for a given set of performance specifications, an interesting question lies in the idea of using the envelope as a means to automatically tune the system's performance. If the simulation and envelope creation algorithm can be wrapped within an optimization algorithm, it could be asked to maximize the size of the envelope within the best performance capabilities of each of the subsystems under investigation.

5.2.5. Simulation Validation through Real-World Testing

The results presented in this paper are generated using modelling and simulation with models of the aerial sensor platform and its environment. The resulting SHOL is therefore subject to the uncertainty contained within all of the models used and the assumptions made in creating them. Sections 5.2.1 and 5.2.2 outline how the fidelity of the modelling can be increased by introducing further modelling complexity. However, it is recognized that, in the end, the SHOLs produced will need to be validated by real-world testing. This testing forms part of the future plans of the work and will involve laboratory-scale experimentation in the first instance.

Author Contributions: Conceptualization, M.J., V.P., M.W., and C.D.; methodology, M.J.; software, V.P. and C.D.; validation, V.P. and C.D.; formal analysis, V.P.; investigation, V.P.; resources, V.P. and C.D.; data curation, V.P. and C.D.; writing—original draft preparation, V.P.; writing—review and editing, C.D. and M.J.; visualization, V.P.; supervision, M.J.; project administration, M.J.; funding acquisition, M.F. and M.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Engineering and Physical Sciences Research Council, grant number EP/R026173/1.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The models, scripts, and example data set used in this paper can be found on the Liverpool Data Catalogue. <https://doi.org/10.17638/datacat.liverpool.ac.uk/1407>, uploaded on 23 July 2021.

Acknowledgments: The authors would like to acknowledge the support of the ORCA RAI hub and the Royal Academy of Engineering.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A

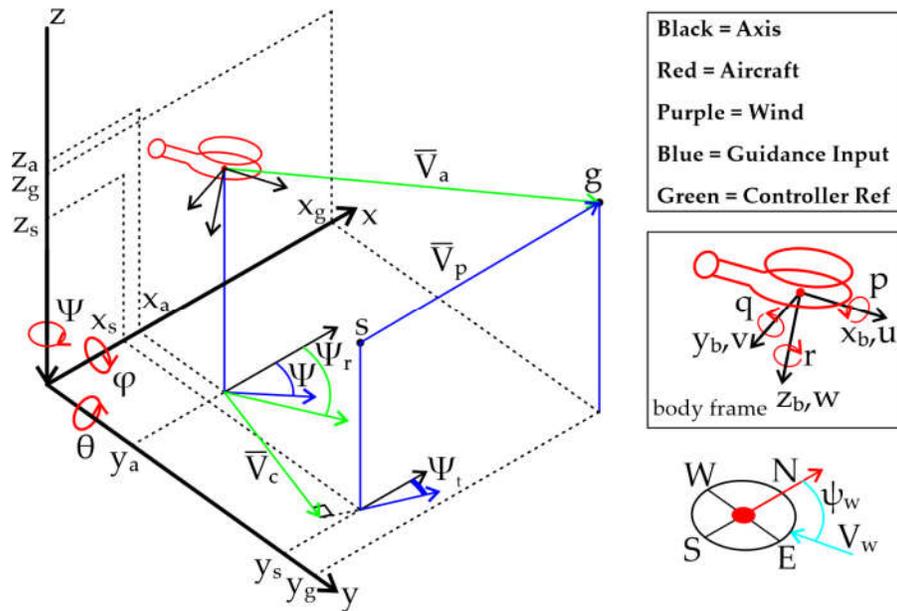


Figure A1. Axes system and variables for the simulator.

Table A1. Nomenclature.

Symbol	Definition	Units	Symbol	Definition	Units
Simulation					
x, y, z	Position components in world frame cartesian axes	m	p, q, r	Aircraft angular body rates	$\frac{rad}{s}$
x_a, y_a, z_a	Aircraft actual position components	m	U	Aircraft groundspeed in horizontal plane	$\frac{m}{s}$
x_b, y_b, z_b	Aircraft body axes	m	V	Aircraft true airspeed	$\frac{m}{s}$
x_s, y_s, z_s	Aircraft start position components	m	V_w, ψ_w	Wind speed and wind direction	$\frac{m}{s}, rad$
x_g, y_g, z_g	Aircraft goal position components	m	u_w, v_w	Wind speed components in the inertial frame	$\frac{m}{s}$
u, v, w	Aircraft body indicated velocity components in $x, y,$ and z directions, respectively	$\frac{m}{s}$	$\delta_a, \delta_b, \delta_c, \delta$	Aircraft lateral, longitudinal, collective, and pedal control inputs, respectively	$\%$
u_g, v_g, w_g	Aircraft body ground velocity components in $x, y,$ and z directions, respectively	$\frac{m}{s}$	$\bar{U}, \bar{V}_w, \bar{\psi}_w$	Vectors of aircraft groundspeed, wind speed, and wind direction that models were produced at	$knots, knots, rad$
ψ, θ, ϕ	Aircraft orientation in heading, pitch and roll Euler angles, respectively	rad	U_q, V_{wq}, ψ_w	Query points for groundspeed, wind speed, and wind directions	$knots, knots, rad$
Controller					
A, B	Linear model state and control matrices		Q, R	LQR method weighting matrices	

A_{aug}, B_{aug}	Augmented state and control matrices	\bar{W}	A weighting with reference to a particular variable as a subscript	
L	A filler matrix used to produce the augmented output matrix	k_{base}	Database of LQR tuned control gains	
I	An identity matrix			
Guidance				
f, f_a, f_b	Generic function identifiers	\hat{a}	Unit vector between aircraft start and goal states	
$f_{a\psi}, f_{a\psi}, f_{b\psi}, f_{b\psi}$	Partial derivatives of equations f_a and f_b with respect to V and ψ	\bar{e}	Cross tracking error	$\frac{m}{s}$
i, j	Step or iteration count	u_r, v_r, w_r, ψ_r	Reference values for aircraft body velocities and heading	$\frac{m}{s}, rad$
$\bar{a}, \bar{b}, \bar{c}, \bar{d}$	Vectors between the various aircraft position states	m		
Navigation				
α, β	Cost function weightings	R	Risk metric	
g, h	The Euclidean distance between grid points, and the straight line distance between a grid point and goal	m	r	Aggregate risk between two points
Performance Metrics				
$F()$	An objective function	\bar{S}	Error specification for a subsystem cost function	
$G()$	A penalty function	t_m	Maximum simulation time	s
$\bar{\theta}$	A set of parameters	\bar{x}	A system state	
d_p, d_l	Distance from nearest successful point and perpendicular distance from line between nearest success points	m	n, n_a, n_r	Number of subsystems, actuators, and controller loops, respectively
d_g	Perpendicular distance from the path	m	P_{actual}	Actual proximity to an obstacle m
$P()$	A Pareto function			
F_A, F_C, F_G, F_N	Objective functions for the actuator, controller, guidance and navigation, respectively			

Appendix B

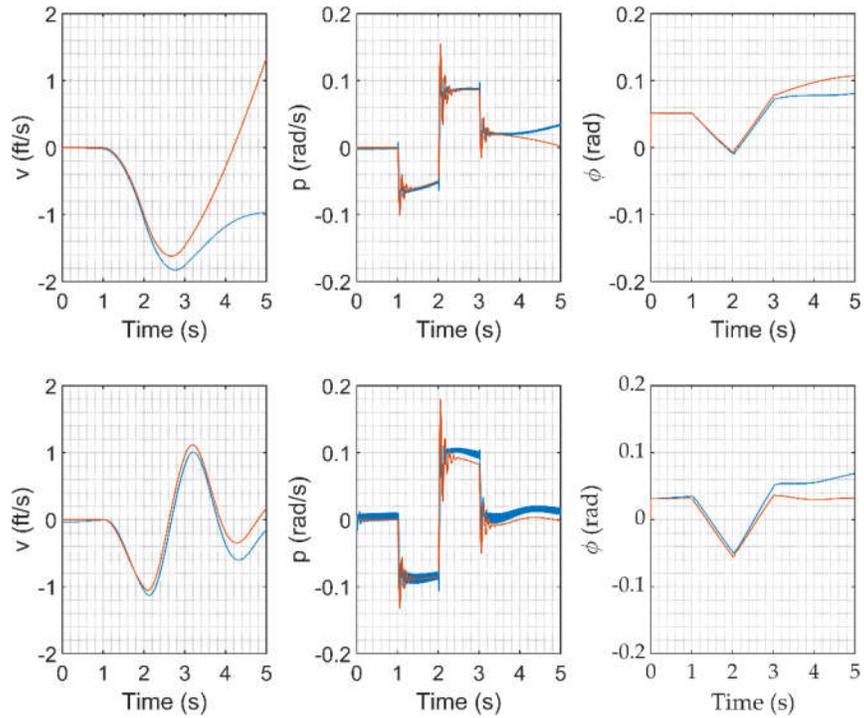


Figure A2. Nonlinear (blue) vs. linear (red) response for a 2% doublet input on the lateral control. Top row is for hover and bottom row for forward flight at 20 Kts.

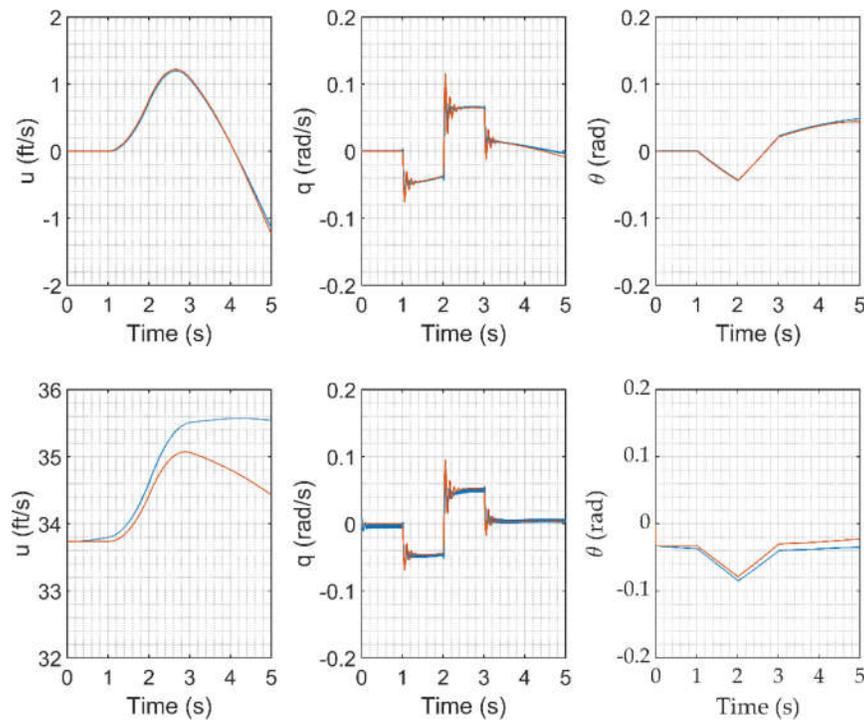


Figure A3. Nonlinear (blue) vs. linear (red) response for a 2% doublet input on the longitudinal control. Top row is for hover and bottom row for forward flight at 20 Kts.

Appendix C

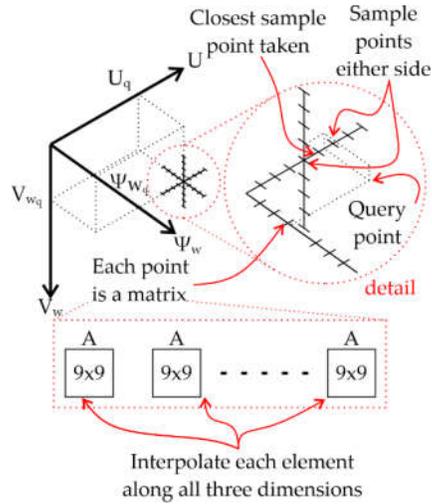


Figure A4. Stitching method to interpolate between the different linear models based on the changing operating points.

Equations (A1)–(A4) provide the basis for this stitching process:

$$\mathbf{A}_{ij} = f(U_q, V_{wq}, \psi_{wq}, \bar{U}, \bar{V}_w, \bar{\psi}_w, \mathbf{A}_{base_{ij}}) \text{ for } i = 1:9 \text{ and } j = 1:9 \quad (\text{A1})$$

$$\mathbf{B}_{ij} = f(U_q, V_{wq}, \psi_{wq}, \bar{U}, \bar{V}_w, \bar{\psi}_w, \mathbf{B}_{base_{ij}}) \text{ for } i = 1:9 \text{ and } j = 1:4 \quad (\text{A2})$$

$$\mathbf{k}_{ij} = f(U_q, V_{wq}, \psi_{wq}, \bar{U}, \bar{V}_w, \bar{\psi}_w, \mathbf{k}_{base_{ij}}) \text{ for } i = 1:4 \text{ and } j = 1:13 \quad (\text{A3})$$

$$\overline{trim}_i = f(U_q, V_{wq}, \psi_{wq}, \bar{U}, \bar{V}_w, \bar{\psi}_w, \overline{trim}_{base_i}) \text{ for } i = 1:6 \quad (\text{A4})$$

Appendix D

The matrices used in the LQR tuning process are shown in Equations (A5)–(A7):

$$\mathbf{L} = [0_{4 \times 2} \quad \mathbf{I}_{4 \times 4} \quad 0_{4 \times 3}]_{4 \times 9} \quad (\text{A5})$$

$$\mathbf{A}_{aug} = \begin{bmatrix} \mathbf{A}_{9 \times 9} & 0_{9 \times 4} \\ \mathbf{L}_{4 \times 9} & 0_{4 \times 4} \end{bmatrix}_{13 \times 13} \quad (\text{A6})$$

$$\mathbf{B}_{aug} = \begin{bmatrix} \mathbf{B}_{9 \times 4} \\ 0_{4 \times 4} \end{bmatrix}_{13 \times 4} \quad (\text{A7})$$

The weightings achieved for the LQR tuning, Q , are given in Equations (A8)–(A12):

$$\bar{W}_{\phi, \theta, \psi} = [10, 10, 10000] \quad (\text{A8})$$

$$\bar{W}_{u, v, q} = [100, 100, 100] \quad (\text{A9})$$

$$\bar{W}_{p, q, r} = [10, 10, 10] \quad (\text{A10})$$

$$\bar{W}_{u_r, v_r, w_r, \psi_r} = [100, 100, 100, 100] \quad (\text{A11})$$

$$\bar{W}_{\delta_b, \delta_a, \delta_c, \delta_p} = [0.1, 0.1, 0.1, 0.1] \quad (\text{A12})$$

The tuning was then done on all the models at the operating points following Equations (A13)–(A15):

$$\mathbf{Q} = [\bar{W}_{\phi, \theta, \psi} \quad \bar{W}_{u, v, q} \quad \bar{W}_{p, q, r} \quad \bar{W}_{u_r, v_r, w_r, \psi_r}]_{13 \times 13} \quad (\text{A13})$$

$$\mathbf{R} = [\bar{W}_{\delta_b, \delta_a, \delta_c, \delta_p}]_{4 \times 4} \quad (\text{A14})$$

$$\mathbf{k}_{base_{U, V_w, \psi_w}} = LQR(\mathbf{A}_{aug}, \mathbf{B}_{aug}, \mathbf{Q}, \mathbf{R}) \quad (\text{A15})$$

Appendix E

The heading of the UAS is calculated by taking into account the wind, desired ground speed, and the direction of travel toward the waypoint, see Figure A5. This allows the UAS to crab into the wind and still travel to the waypoint. By asserting that the ground speed of the UAS toward the waypoint, u_g , should be V_{max} and that the side slip ground speed, v_g , should be zero, the following equations can be formed:

$$f_a(V, \psi_r) = u_g = V \cos(\psi_t - \psi_r) \pm V_w \cos(\psi_w - \psi_t) - V_{max} = 0 \tag{A16}$$

$$f_b(V, \psi_r) = v_g = V \sin(\psi_t - \psi_r) \pm V_w \sin(\psi_w - \psi_t) = 0 \tag{A17}$$

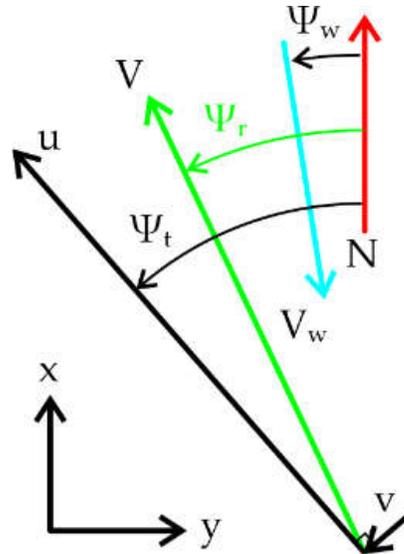


Figure A5. Calculation of UAS heading while tacking into the wind.

This creates two simultaneous nonlinear equations, which are then solved using a multidimensional Newton–Raphson method. The Jacobian matrix is formed, as per Equations (A19)–(A23), using backward difference to estimate the gradient. These equations are then iterated until a tolerance is met:

$$f_{aV} = \frac{f_a(V_{i-1}, \psi_{r_i}) - f_a(V_i, \psi_{r_i})}{V_{t,i-1} - V_{t,i}}, \tag{A18}$$

$$f_{a\psi} = \frac{f_a(V_i, \psi_{r_{i-1}}) - f_a(V_i, \psi_{r_i})}{\psi_{r_{i-1}} - \psi_{r_i}} \tag{A19}$$

$$f_{bV} = \frac{f_b(V_{i-1}, \psi_{r_i}) - f_b(V_i, \psi_{r_i})}{V_{i-1} - V_i}, \tag{A20}$$

$$f_{b\psi} = \frac{f_b(V_i, \psi_{r_{i-1}}) - f_b(V_i, \psi_{r_i})}{\psi_{r_{i-1}} - \psi_{r_i}} \tag{A21}$$

$$\begin{bmatrix} V_{i+1} \\ \psi_{r_{i+1}} \end{bmatrix} = \begin{bmatrix} V_i \\ \psi_{r_i} \end{bmatrix} - h_s \begin{bmatrix} f_{aV} & f_{a\psi} \\ f_{bV} & f_{b\psi} \end{bmatrix}^{-1} \begin{bmatrix} f_a(V_i, \psi_{r_i}) \\ f_b(V_i, \psi_{r_i}) \end{bmatrix} \tag{A22}$$

It can be seen that, using these equations with an increasing crosswind, the result will be for a demand to turn increasingly into the wind, as would be expected. In the extreme case, this would lead to the desired heading being perpendicular to the required direction of travel. Whilst this might be possible with a helicopter, it is undesirable behavior. To prevent it, a limit on the difference between the heading of the aircraft while tacking into the wind and the desired direction of travel is applied.

The ground velocity of the aircraft is calculated using simple waypoint following with cross-tracking error [25], see Figure A6. This is done by first calculating the vector toward the next waypoint from the previous using the following equations:

$$\bar{a} = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} - \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} \quad (\text{A23})$$

$$\hat{a} = \frac{\bar{a}}{|\bar{a}|} \quad (\text{A24})$$

Then the vector to the goal from the UAS current location using the following:

$$\bar{b} = \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} - \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} \quad (\text{A25})$$

The perpendicular vector to the path can then be calculated via:

$$\bar{d} = \bar{b} - (\hat{a} \cdot \bar{b}) \hat{a} \quad (\text{A26})$$

The goal vector then combines the goal vector with the cross-tracking vector to get the resulting vector:

$$\bar{e} = \bar{b} + \bar{d} = \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} \quad (\text{A27})$$

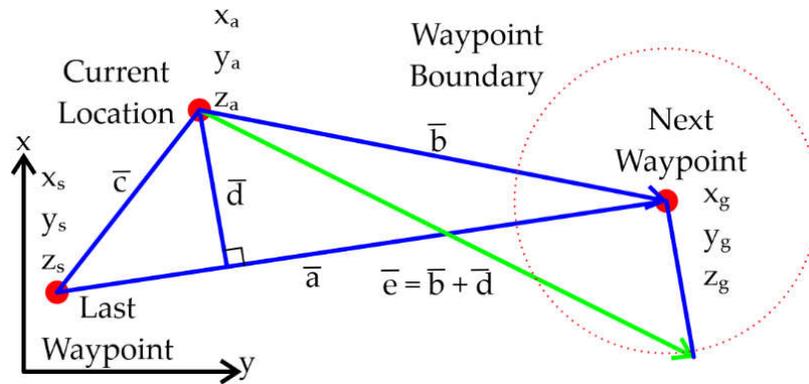


Figure A6. Calculation of velocity using waypoint following and cross-tracking error.

These vector components are then rotated from the inertial axis to the body axis to form the reference values for the controller to follow.

Appendix F

The cost function for the A* heuristic is as follows:

$$Cost_i = ((Cost_{i-1} + g_i)(1 - \alpha) + h_i \alpha) \frac{\beta_D}{\beta_t} + ((Cost_{r_{i-1}} + R_i \alpha)(1 - \alpha) + A_{r_i} \alpha) \frac{\beta_R}{\beta_t} \quad (\text{A28})$$

$$\beta_t = \beta_D + \beta_R \quad (\text{A29})$$

$$R_i = R_{object_i} + R_{wake_i} \quad (\text{A30})$$

where β_R and β_D are weightings between the risk and distance; α is the weight between cumulative cost and heuristic cost; R_i is the combined risk from the object proximity risk, R_{object} , and wake risk, R_{wake} , modelled at the grid point; g_i is the Euclidean distance between the grid points; h_i is the straight line distance from the grid point to the goal; and A_{r_i} the aggregate risk of moving from one grid point to another calculated as the integral between one grid point and another.

References

1. Soubry, A.; Leadsom, A. Oil and Gas Workforce Plan. HM Government. July 2016. Available online: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/535039/bis-16-266-oil-and-gas-workforce-plan.pdf (accessed on 7 January 2021)
2. Cable, V.; Davey, E.; Moore, M.; Ballard, G. UK Oil and Gas: Business and Government Actio. HM Government. March. 2013. Available online: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/175480/bis-13-748-uk-oil-and-gas-industrial-strategy.pdf (accessed on 7 January 2021)
3. Wooldridge, M. *An Introduction to Multiagent Systems*; 2nd ed. University of Liverpool, UK: John Wiley & Sons, 2009.
4. CAP 722 Unmanned Aircraft System Operations in UK Airspace: Guidance; UK Civil Aviation Authority: Gatwick, UK, 2015.
5. Huang, H. *Autonomy Levels for Unmanned Systems (ALFUS) Framework*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2007.
6. Scanlan, J.; Flynn, D.; Lane, D.; Richardson, R.; Richardson, T.; Sobester, A. *Extreme Environments Robotics: Robotics for Emergency Response, Disaster Relief and Resilience*; UK-RAS Network: Loughborough, UK, 2018.
7. Carico, G.D.; Fang, R.; Finch, R.S.; Geyer, W.P., Jr.; Krijins, H.W.; Long, K. Helicopter/Ship Qualification Testing: Part 1 Ductch/British Clearance Process. *NATO RTO* **2003**, *22*, 13–23.
8. Forrest, J.S.; Owen, I.; Padfield, G.D.; Hodge, S.J. Ship-Helicopter Operating Limits Prediction Using Piloted Flight Simulation and Time-Accurate Airwakes. *J. Aircr.* **2012**, *49*, 1020–1031.
9. Finlay, B.A. Ship Helicopter Operating Limit Testing—Past, Present and Future’, In Proceedings of the *RAeS Rotorcraft Group Conference on ‘Helicopter Operations in the Maritime Environment’*, London, UK, 7–8 November 2001.
10. S. K. Advani and C. H. Wilkinson, ‘Dynamic Interface Modelling and Simulation: A Unique Challenge. In Proceedings of *Royal Aeronautical Society Conference on Helicopter Flight Simulation*, Royal Aeronautical Soc. (RAeS), London, UK, Nov. 2001.
11. Healey, J.V. *Simulating the Helicopter-Ship Interface as an Alternative to Current Methods of Determining the Safe Operating Envelopes*; Naval Postgraduate School: Monterey, CA, USA, 1986; pp. 48–58.
12. Roscoe, M.F.; Wilkinson, C.H. DIMSS-JSHIP’s M&S Process for Ship Helicopter Testing and Training. In Proceedings of AIAA Modeling and Simulation Technologies Conference and Exhibit, Monterey, CA, USA, 5–8 June 2002.
13. Owen, I.; White, M.; Padfield, G.D.; Hodge, S.J. A virtual engineering approach to the ship-helicopter dynamic interface—A decade of modelling and simulation research at the University of Liverpool. *Aeronaut. J.* **2017**, *121*, 1833–1857.
14. Hodge, S.J.; Forrest, J.; Padfield, G.D.; Owen, I. Simulating the Environment at the Helicopter-Ship Dynamic Interface: Research, Development and Application. *Aeronaut. J.* **2012**, *116*, 1155–1184.
15. Kelly, M.; Watson, N.A.; Hodge, S.J.; White, M.D.; Owen, I. The Role of Modelling and Simulation in the Preparations for Flight Trials Aboard the Queen Elizabeth Class Aircraft Carriers. In Proceedings of the 14th International Naval Engineering Conference, The Institute of Marine Engineering, Science and Technology (IMarEST), October 2014. Available online: <https://doi.org/10.24868/issn.2515-818X.2018.037> (accessed on 7 January 2021).
16. Memon, W.A.; Owen, I.; White, M.D. SIMSHOL: A Predictive Simulation Approach to Inform Helicopter–Ship Clearance Trials. *AIAA* **2020**, doi:10.2514/1.C035677
17. Page, P.; Webster, M.; Fisher, M.; Jump, M. Towards a Methodology to Test UAVs in Hazardous Environments. In Proceedings of the ICAS 2019, The Fifteenth International Conference on Autonomic and Autonomous Systems, Athens, Greece, 2–6 June 2019; pp. 38–45.
18. Fell, T.R.; White, M.D.; Owen, I. Sensitivity study of a small maritime rotary UAS operating in a turbulent airwake. In Proceedings of the 71st AHS Forum, Virginia Beach, VA, USA, 27–29 May 2015.
19. Val, R.D.; He, C. Validation of the FLIGHTLAB Virutal Engineering Toolset. *Aeronaut. J.* **2018**, *122*, 519–555.
20. Padfield, G.D. *Helicopter Flight Dynamics*, 1st ed. Blackwell Science Ltd: London, UK, 1996.
21. Furgeson, D.; Likhachev, M.; Stentz, A. A guide to heuristic-based Path Planning. In Proceedings of ICAPS’05 15th International Conference on Automated Planning & Scheduling, 5–10 June 2005, Monterey, CA, USA.
22. Moss CS50 MKII Oil Platform. Available online: <https://grabcad.com/library/moss-cs50-mkii-oil-platform> (accessed on 7 January 2021).
23. Webster, M.; Western, D.; Araiza-Illan, D.; Dixon, C.; Eder, K.; Fisher, M.; Pipe, A. A Corroborative Approach to Verification and Validation of Human-Robot Teams. *Int. J. Robot. Res.* **2020**, *39*, doi:10.1177/0278364919883338.
24. Fisher, M.; Cardoso, R.C.; Collins, E.; Dadswell, C.; Dennis, L.A.; Dixon, C.; Farrell, M.; Ferrando, A.; Huang, H.; Jump, M.; et al. An Overview of Verification and Validation Challenges for Inspection Robots. *Robotics*, **2021**, *10*, doi:10.3390/robotics10020067.
25. Shima, T.; Rasmussen, S. *UAV Cooperative Decision and Control: Challenges and Practical Approaches*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2009.