Intelligent Contingency Management for Urban Air Mobility

Irene M. Gregory[1], Michael J. Acheson[2], Barton J. Bacon[3], Thomas C. Britton[4],
NASA Langley Research Center, Hampton, VA, 23681, USA

Newton H. Campbell[5]
NASA Goddard Space Flight Center, Greenbelt, MD, 20771, USA

Jacob W. Cook[6], Jon B. Holbrook[7], Daniel D. Moerder[8], Patrick C. Murphy[9], Natasha A. Neogi[10], Benjamin M. Simmons[11], John D. McMinn[12] and Pieter G. Buning[13]
NASA Langley Research Center, Hampton, VA, 23681, USA

The third aviation revolution is seeking to enable transportation where users have access to immediate and flexible air travel; the users dictate trip origin, destination and timing. One of the major components of this vision is urban air mobility (UAM) for the masses. UAM means a safe and efficient system for vehicles to move passengers and cargo within a city. In order to reach UAM's full market potential the vehicle will have to be autonomous. One of the primary challenges of autonomous flight is dealing with off-nominal events, both common and unforeseen; thus, intelligent contingency management (ICM) is one of the enabling technologies. This paper proposes an ICM architecture with associated tools that would help enable the UAM vision.

I. Introduction

The third aviation revolution is seeking to enable transportation where users have access to immediate and flexible air travel; the users only need to dictate trip origin, destination, and timing. One of the major components of this vision is urban air mobility (UAM) for the masses. UAM means a safe and efficient system for vehicles to move passengers

[1] NASA Senior Technologist for Advanced Control Theory and Application, Research Directorate, AIAA Associate Fellow.
[2] NASA Aerospace Technologist, Dynamics Systems and Control Branch, Member AIAA.
[3] Research Engineer, Dynamic Systems and Control Branch, Member AIAA.
[4] Principal Technical Professional - Software Engineering, KBR, Corp.
[5] Senior Principal Solutions Architect, Science Applications International Corporation, NASA Enterprise IT.
[6] Research Engineer Trainee, Dynamics Systems and Control Branch, Member AIAA.
[7] Research Scientist, Crew Systems and Aviation Operations Branch.
[8] Research Engineer, Dynamic Systems and Control Branch.
[9] Research Engineer, Dynamic Systems and Control Branch, AIAA Associate Fellow.
[10] Research Scientist, Safety Critical Avionics Branch, AIAA Associate Fellow.
[11] Research Engineer, Flight Dynamics Branch, Member AIAA.
[12] Research Engineer, Dynamic Systems and Control Branch.
[13] Senior Research Scientist, Computational AeroSciences Branch, AIAA Associate Fellow.

and cargo within a city. UAM is not a new concept; currently there are helicopter services within large metropolitan areas that shuttle a small set of users between predefined set of destinations. The paradigm shift in this new incarnation of UAM is the democratization of the service (Figure 1) [1]. Thus, at its core, UAM refers to the aerial movement of people, cargo and information, from one point in an urban landscape to another. UAM has the potential to reduce emergency response time, aid in combatting congestion in dense, urban cores characterized by impasses (e.g., bridges and tunnels), and improve the comfort and speed of travel. The convergence of autonomous capability and adoption of electric vehicle technologies will likely set an industry standard and will enable these missions.

UAM also has the potential to enable a suite of advanced aerial mobility missions in the surrounding metropolitan areas, such as suburban and rural communities. As the technologies of UAM mature, the ability to facilitate intra-city transportation for short takeoff and landing applications, as well as operations such as package delivery for medical transport, will act to unlock opportunities for social and economic engagement in all areas of the nation.

The ability to access the airspace above urban areas for commercial opportunities via small unmanned aerial systems (UAS) under the FAA's Regulations (see Small Aircraft Rule (Part 107)) has created an industry that has been assessed to be worth $22 billion USD worldwide [2]. These assessed operations are mostly limited to surveying and surveillance applications, such as agricultural, infrastructure inspection, journalism, film making and law enforcement. Initial cargo-carrying operations, in rural areas, have just begun in the U.S., and are projected to become a dominant economic driver in urban and suburban areas. The UAS industry has also catalyzed interest in passenger-carrying UAM operations, which may be worth $15.2 billion USD by 2030 [3].

In this effort, the focus is on a concept of operations that will involve passenger carrying UAM vehicles operating in an urban environment. Operations are considered for an intermediate state, that is defined by hundreds of simultaneous operations; expanded networks including high-capacity UAM ports; many available UAS Traffic Management [UTM]-inspired Aircraft Traffic Management services; simplified requirements for pilot qualification; and low-visibility operations.[4] Note that this vision requires that several key assumptions be made, specifically regarding the increasing level of autonomous operation exhibited by the vehicle as well as for UAM airspace management.

The intermediate UAM mission is assumed to be comprised of a vehicle, that takes off from a pre-prepared landing site (e.g., a UAM port), and travels to another pre-prepared landing site. There is appropriate communications, navigation and surveillance infrastructure available to enable multiple simultaneous operations at a UAM port (and to enable hundreds simultaneous operations in the urban airspace). The vehicle has some level of human supervision, be it remote or in the cockpit. However, the assumption of a highly trained pilot may no longer be valid, and the role of the human is that of an operator. The vehicle may encounter disturbances (e.g., weather events) and disruptions (e.g., restricted airspace or weather) in the course of its journey; and it may have to deal with contingencies (e.g., faults and failures) in an increasingly autonomous fashion.



Figure 1. UAM notional mission

In its full instantiation the new market's operational model is anyone, anytime, anywhere. Convergence of improved battery technology, development of electric propulsion and increasing capability of autonomous systems has driven this paradigm shift. This paradigm shift has implications on the vehicles required for UAM. The full

development of this emerging market requires inexpensive autonomous vehicles with short development cycle and unconventional configurations to take full advantage of new technologies and to satisfy noise and weather tolerance constraints. Autonomous flight is required for full market instantiation and operation in complex densely populated environment imposes additional requirements on autonomous flight. Even early phase of implementation, prior to full autonomy and envisioning simplified vehicle operations by a pilot, still depend on increasing level of autonomous capabilities.

There are several challenges posed by UAM operations and these challenges can be broadly separated into airspace-oriented challenges, vehicle-oriented challenges, community integration challenges and cross-cutting challenges. Airspace challenges include, but are not limited to designing the airspace and operational procedures for UAM operations – including the design and operation of UAM ports and necessary supporting infrastructure, disruption due to air space restrictions, fleet management, and urban weather prediction. Vehicle challenges include safety, certification and noise qualities, as well as addressing issues such as increasing automation, manufacturing and supply chain issues, and maintenance. Community integration concerns are focused around public acceptance of the operations, their integration into a multi-modal transportation system and infrastructure (and the incorporation of smart cities technologies), as well as the local regulatory environment. Broad challenges such as safety, certification, autonomy and infrastructure required for UAM operations cut across all stakeholders in the UAM ecosystem and will require integrative solutions.

As these operations scale to the point where they will be profitable, it has been suggested that an increasingly autonomous operational paradigm will be required. [5, 6]. Moreover, in order to reach UAM's full market potential the vehicle will have to be autonomous. One of the primary challenges of autonomous flight, even at intermediate stage, is dealing with off-nominal events, both common and unforeseen; thus, intelligent contingency management (ICM) becomes one of the enabling technologies.

This paper presents one possible approach and high-level architecture for intelligent contingency management. The presentation is organized as follows. Section II discusses the proposed approach and high-level architecture, while Section III gives a brief overview of the development environment used for tool integration, testing and evaluation. Section IV discusses vehicle capability assessment; Section V presents mission execution aspects of the architecture. Section VI presents a brief discussion of human contribution to safety in the form of contributing resilient strategies that can be employed by any system agents, machine or human. Finally, Section VII provides a summary of key points.

II. Approach to Intelligent Contingency Management

An approach to intelligent contingency management and associated high-level architecture is proposed in Figure 2. The architecture and associated functionality would allow the vehicle to safely achieve its mission by flying from pt. A to pt. B under all vehicle-allowable weather conditions, in a high-density airspace complex urban environment, while reacting appropriately to off-nominal situations and contingencies without direct human control. Currently contingency management is a highly prescribed, rule-based approach. This effort is interested in exploring contingency management that expands the coverage of possible continencies, to include possible unanticipated situations, without exclusive reliance on highly prescribed, rule-based approach. Intelligent approach to contingency management may provide a more general coverage of vehicles and operating environment contingencies in a safe and cost-effective manner. The presented approach to ICM considers some emerging techniques in machine learning and explores their integration into safety-critical systems and potential levels of assurance. Some of the techniques under exploration are multi-agent reinforcement learning, variational autoencoders, and several of flavors of neural networks. We are also investigating several assurance techniques for both design and operational implementation.

The proposed high-level architecture in Figure 2 is essentially divided into two parts. First vehicle capability to perform its predefined mission has to be assessed. This involves understanding vehicle current state and predicting that state into the future. The vehicle current and predicted future state are fed into mission execution which controls all aspects of vehicle flight. Note that there is a feedback loop back from mission execution to vehicle capability assessment since the manner in which the mission is executed creates requirements on vehicle capability. Figure 3 provides the next level of detail of the proposed ICM architecture.

Vehicle capability is informed by sensor and fused data, combining internal and external sources, and is constrained by outside factors such as weather and traffic management constraints; the vehicle is an agent in a larger multi-agent system that operates by its own set of rules and constraints. Figure 3 provides next level of detail on how the vehicle capability to perform its predefined mission is assessed. The starting point is availability of vehicle models. High fidelity model is a necessary element of the integrated simulation-based evaluation tool. Additionally, due to unconventional nature of UAM-class vehicle configurations, high fidelity model is necessary to provide physical

insight into dynamic behavior of the vehicle. Any decisions made regarding vehicle capabilities must reflect the best understanding of the underlying physics. Additional vehicle models of variable fidelity are also a necessary part of the toolset. Dynamic models of appropriate fidelity are necessary for control design and trajectory planning functions. Moreover, an onboard model updated in real-time is a critical component of generating actual vehicle capabilities and informing overall control strategies.

One of the ways to measure vehicle capability is by examining flight and maneuver envelopes. The nominal version of the envelopes can be precomputed using high fidelity vehicle model. During flight actual flight and maneuver envelopes can be updated to reflect actual vehicle capabilities. Some of the methods that can be utilized to derive actual flight envelopes in flight are discussed later in the paper. Within the proposed architecture, vehicle envelopes are employed to ascertain trajectory following capabilities and safe reachable states, which predict vehicle future state as discussed in subsequent sections.

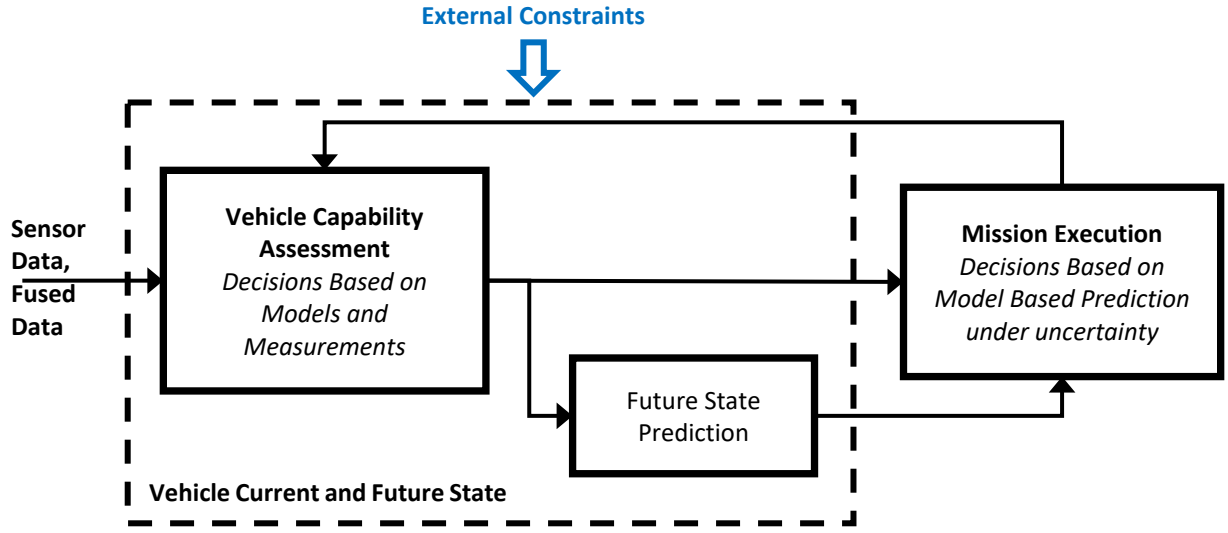


Figure 2. High level Intelligent Contingency Management (ICM) architecture

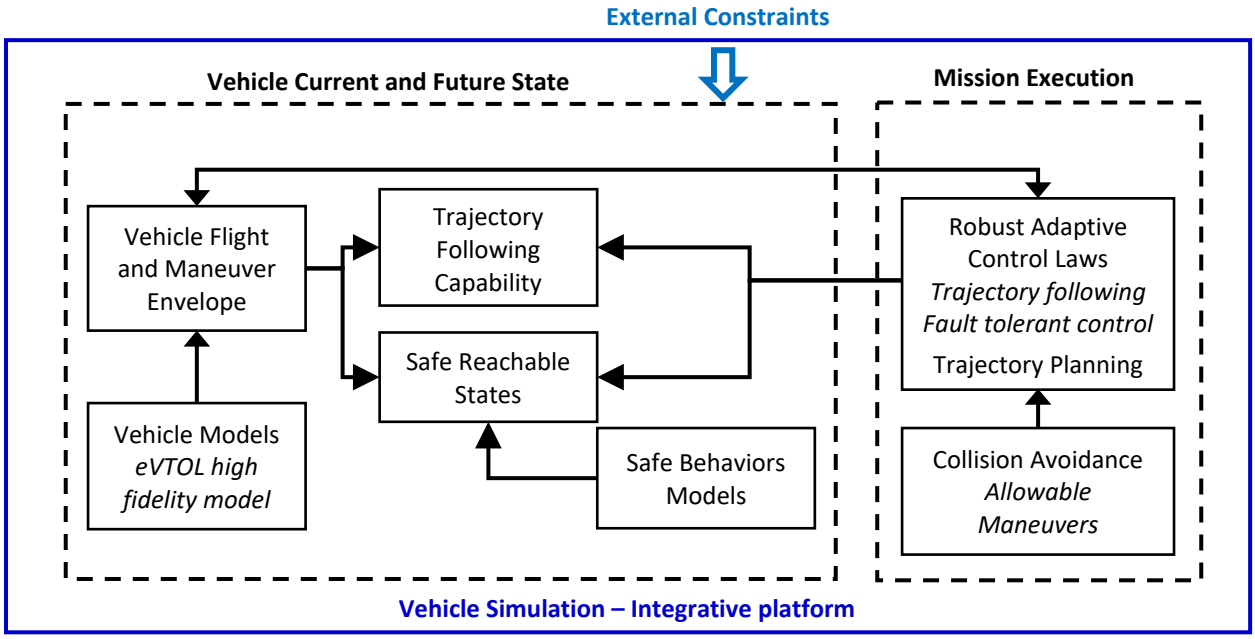


Figure 3. Lower level Intelligent Contingency Management (ICM) architecture

The last block in the vehicle state part of the architecture is the safe behaviors models. The approach discussed in this work is based on human contribution to safety focusing on resilient behaviors. More detailed discussion is presented in Section VI. Note that within this part of the ICM architecture decisions are made based on models as well as measurements with their associated uncertainty.

The mission execution part of the architecture, discussed in further detail in Section V, starts with robust adaptive control laws as the fundamental building block. The objective of these is to provide robust performance for trajectory following subject to atmospheric disturbances, vehicle model uncertainty and potential faults and failures. Since vehicle configurations are unconventional and propulsion technology is new, it is reasonable to assume that UAM-class vehicles would experience higher rates of faults and failures than is typical for a well-proven commercial transport aircraft. Hence, a robust fault-tolerant flight control is a necessity from safety and operational performance perspective. The ability of the robust adaptive control laws to handle faults/failures and atmospheric disturbances while providing robust performance directly informs and impacts vehicle capability assessment. One of the outcomes of the feedback decision loop is a requirement for onboard trajectory planning. Even if such capability may be available as part of the services architecture for traffic management system, it is a safety-critical function that must be available onboard in case of lost/degraded link event. Informing robust adaptive control and trajectory planning is another safety-critical function – collision avoidance. This paper explores a deep reinforcement learning approach with limited knowledge of intruding vehicle states to provide allowable maneuvers consistent with own ship capabilities.

The decisions made under mission execution block of the architecture are dependent on model-based prediction under uncertainty. Model based prediction is itself dependent on models of vehicle behavior. Hence, all decisions that are made are made under uncertainty. This makes multiple approaches to evaluate vehicle's ability to safely fly its specified mission under current and projected constraints a necessity for robustness and resilience. In addition, the objective of a well-designed intelligent contingency management system is to enhance vehicle operational safety as well as to minimize disruption to the larger system within which it operates.

III. Development Environment

This section briefly discusses the environment that the ICM architecture and its associated tools are developed and evaluated in. The development environment of choice is the MathWorks Matlab/Simulink® software which is widely used due to familiarity, ease of use, and evolving tools base. Moreover, ability of going from graphical system representation to automatically generated code that can be executed on a computational cluster for improved efficiency or integrated into other simulation environments makes Matlab/Simulink® a desirable development environment.

A. Simulation Environment – An Integrative Tool

To further investigation into UAM-type vehicles and create an integrated tool environment, a nonlinear simulation is being developed using the Matlab/Simulink® environment. The simulation makes use of capabilities within Simulink® to allow for vehicle-specific implementations, such as variant subsystems, subsystem references, and defined bus objects. These capabilities allow the implementation of a modularized framework that can support the simulation of low to medium fidelity vehicles. While the simulation is still under development, this section discusses existing or nearly completed capabilities. The modularized framework allows the modelling of multiple concept configurations of vertical takeoff and cruise vehicles for supporting the investigation of exploratory concepts for hierarchical control system and decision making, as well as the integration of intelligent contingency management tools currently under development.

Vehicle models include six degree-of-freedom dynamics, including aerodynamic, propulsion, actuator, and atmosphere models. The simulation architecture can accommodate variable fidelity models for each aspect of the vehicle. Further discussion of vehicle models is provided in Section IV. The simulation includes mechanisms capable of finding equilibrium manifold and performing model linearization. An integrated part of the overall tool is a linear simulation (based on the linearization results of the full nonlinear simulation) that supports control system design and analysis. Trajectory generation is also part of the toolbox. Currently trajectory data, derived from an Efficient Universal Trajectory Language (EUTL) specification, is utilized to provide reference inputs to the control system. Support for additional trajectory formats is intended.

In order to be able to conduct large-scale batch operations on computer clusters, models were developed specifically to perform code generation, making use of the existing vehicle subsystem references of the nonlinear simulation. Code generation can produce an executable, in addition to a static library capable of being integrated with third party simulation environments. Simulation parameters can be varied for each batch run. This capability is currently being used for neural network training through simulation.

B. Atmospheric Constraints

Another important part of the development environment is atmospheric disturbances since they have direct impact on vehicle's ability to precisely follow a given trajectory, both spatial and temporal specifications. Three types of atmospheric disturbances are being modeled in the near future. These are winds, modeled as variable magnitude and direction along the trajectory, discreet gusts, based on standard Military Specifications and applied on climb out and descent, and turbulence. Applying appropriate turbulence models is of particular importance to UAM-class vehicles due to their size and operating environment. Some of the challenges and proposed initial approach are outlined below.

The key point of turbulence modeling is to inject realism into the design and evaluation of aircraft, aircraft control systems, and piloting simulations. It is known based on aircraft dynamics and human response that the key frequency range for those topics is centered near 1Hz with an order of a decade span [7]. Knowing that, one way to approach the turbulence "problem" is to create something that has some 'appropriate' level of energy around 1 Hz. The Dryden and Van Karman models are the first approach where the 'appropriate level of energy' is informed by statistical measurements. The nuance of UAM is that certain assumptions for Dryden and Van Karman are not valid near the ground, although they still work and produce "turbulence like disturbances."

The number one necessary adjustment is to align the velocity components *(u,v,w)* of turbulence with the relative wind rather than with the body axes of the aircraft. This is because the turbulence near the ground is not isotropic. The intensity of vertical turbulence is different. Dryden and Van Karman take a step to acknowledge that with different sigmas and scale lengths below 1000 feet. The next items to address are surface roughness (city-scape lee flow) and localized heating and the impact this has on the patchiness of turbulence as one flies in and out of these cells/bubbles. The open question is how these effects are most effectively incorporated in a convenient model. To some extent the rotational/vortical flow downstream of terrain/buildings can be modeled as conventional body rate (*p, q, r)* gusts with appropriate energy levels (statistical parameters).

A more difficult question to answer is what are the deficiencies if the conventional turbulence approach is taken, which does produce turbulence. Who is to say the turbulence that is modeled isn't something that might be possible in the atmospheric boundary layer. One of the identified deficiencies of conventional turbulence modeling is that it is "always on." Pilots have commented that turbulence in reality is patchy. Some models started to incorporate time varying statistics, but these should be tied to the terrain/ time of day.

Heavy weather in general is part of the ATC constraints, these dynamic areas would require trajectory replanning and, thus, modification of the original mission. The simulation environment is being developed in such a way as to accommodate new factors that the ICM vehicle system would have to deal with.

IV. Vehicle Capability Assessment

As been described in architecture discussion (Section II), vehicle capability assessment is integral to making decision regarding nominal mission execution or determining necessary modifications if vehicle diminished capability requires. This section outlines development of vehicles models, dynamic flight envelope computation, and future capability prediction using reachable set approach.

The new class of vehicles, often hybrid aircraft-rotorcraft designs, emerging to meet UAM missions offers many design challenges as well as opportunities. For these new vehicles the designs present significant aerodynamic complexity and nonlinearity. Developing models using either CFD or wind tunnels becomes a non-trivial task as the new designs can have 3-5 times the number of factors to consider compared to a traditional airplane or rotorcraft. In addition, the factors can present significant complex interactions. The first part of this section presents the approaches taken to meet the UAM vehicle modeling challenges.

A. Vehicle Model

1. UAM Modeling Challenge

NASA along with partners in government, industry, and academia are working toward developing a new era in air transportation, referred to as Urban Air Mobility (UAM). Several technologies are being developed to make this possible and aerodynamic modeling is a key area being advanced. This area is foundational allowing aircraft designers, flight simulation engineers, flight control designers, and a variety of other technical teams to evaluate and improve their design concepts and technologies. One challenge for modeling is that UAM configurations are typically hybrid vehicles combining conventional airplane and rotorcraft concepts. The combined concepts offer tremendous design opportunity but often at the cost of much higher complexity both in design and in aerodynamic behaviors.

An additional challenge results from the need for aerodynamic models at differing levels of fidelity depending on the phase of development. Early design phases may require lower-fidelity analytical codes and models to ensure basic stability, control, and mission performance at an acceptable cost. Later phases require progressively more sophisticated codes and more detailed aerodynamic models to ensure enough fidelity in predicting aerodynamic behaviors before significant further investments are made. More detailed aerodynamic models with higher fidelity come with much higher costs in terms of time and resources.

Another challenge to modeling occurs when investments are made in high fidelity computational codes, wind tunnel testing, or flight testing. In these later design phases, it is much more difficult to feedback changes into the earlier design phases without substantial costs in re-design. Flight testing can be less costly when sub-scale prototypes are used, however, satisfying similitude requirements is more difficult for rotorcraft and blended airplane-rotorcraft vehicles.

2. Approach to Modeling UAM Vehicles

To ameliorate the challenges caused by complexity, costs, and difficult feedbacks in the aircraft design process, NASA is supporting, under the Transformational Tools and Technologies (TTT) Project, efforts to significantly shorten development and certification time for novel aircraft configurations. One aspect of this goal is supported by development of methods that enhance and speed up the aerodynamic modeling process and in turn allow higher fidelity approaches to provide feedback and gain earlier participation in the design process. Several research efforts have been made and continue to be made in ground-based testing [8-13], flight testing [14-17], and in computational methods [18-24] that seek increased efficiencies and high-fidelity modeling. However, a more recent focus ins Ref. [24-29], has been to more closely address the modeling challenges described herein.

In Ref. [24] preliminary work was described for development of a testing process called Rapid Aero Modeling or RAM. RAM is a guided testing process that can produce mathematical models in a form useful for flight dynamics simulation, design, and analysis. It is based on foundational concepts in two well-established bodies of theory: design of experiment (DOE) and aircraft system identification (SID). The process works toward obtaining aerodynamic models that satisfy user-defined levels of fidelity in terms of model prediction errors. However maximum fidelity is always limited by the accuracy and measurement capabilities of the test facility and physical model under test.

The general RAM concept is presented in Figure 4 as a simplified flow chart. On the left is the user-desired level of model fidelity and a sequence of pre-designed experiments, in the form of sequential DOE blocks of test points, appropriate for the vehicle under test. As the test facility processes each block to produce measurements for model identification, the resulting stepwise regression and Analysis of Variance (ANOVA) tables characterize the relative significance of model terms and other sources of variation. Effectively, the RAM process provides a control loop around the vehicle under test that guides the test through a series of data blocks that are used to estimate models and evaluate their performance. The series of data blocks takes advantage of the sequential nature of designed experiments and only requests more data if it is needed for either estimating a higher-order model or for more detailed investigation in a region where model performance is inadequate. The test is completed when model validation tests are passed, indicating the requested level of fidelity is achieved, or the test may conclude when the limits of the test apparatus measurement capabilities are reached. The choice of identification methods, prediction error metrics, or how the test regions are split is not restricted by the general RAM process. More recently RAM was applied to both wind tunnel and computational experiments [26, 27]. In Ref. [26], RAM-T is demonstrated through application to wind-tunnel experiments for both a conventional aircraft and an eVTOL configuration. In Ref. [27], RAM-C is demonstrated in application to computational experiments for an airfoil and the Lift+Cruise configuration using OVERFLOW. OVERFLOW is a computational fluid dynamics (CFD) flow solver for the time-dependent, compressible, Reynolds-averaged Navier-Stokes equations using multiple overset structured grids [30-32]. The overset grid capability is particularly suited for building up complex configurations by adding components, even allowing the modeling of rotating rotor blades if desired. The use of structured grids provides faster solution turnaround than unstructured grid solvers, at the expense of a more labor-intensive grid generation process. Thus, the solver provides a range of capabilities that allows modeling of UAM components such as rotors, propellers, wings, etc., while allowing users to make trades between computational resources and levels of fidelity.

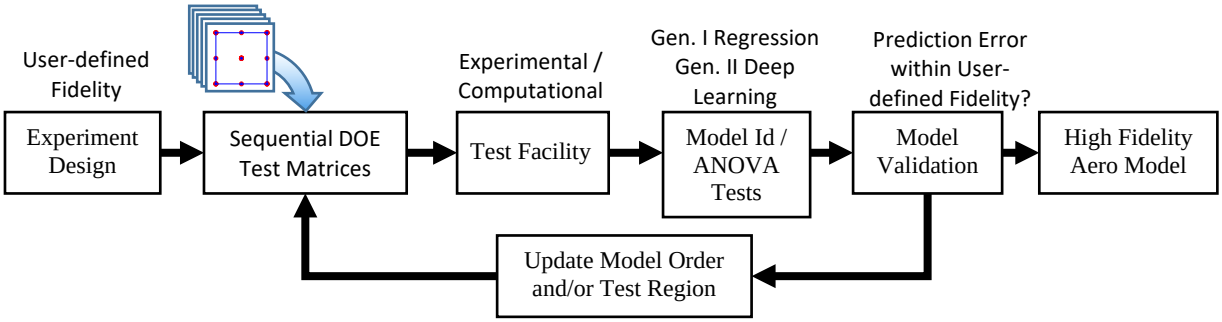


Figure 4. RAM process

In the current study the UAM vehicle of interest is the NASA Lift+Cruise (L+C) configuration, shown in Figure 5. The L+C is a NASA concept multirotor eVTOL vehicle developed under the NASA Revolutionary Vertical Lift Technology (RVLT) Project. This design is still in a relatively early design phase where many features have been defined but the aircraft primarily exists in the form of CAD models appropriate for computational studies. The approach to developing aerodynamic models for the L+C configuration has been to apply the RAM process in combination with OVERFLOW [27]. The target vehicle aerodynamic models are developed treating the configuration in three separate configurations shown in Figure 6. This approach allows simulation models to reflect five different possible flight regions: (1) Glider mode without any propulsion, (2) Cruise mode or airplane mode with only the pusher propeller, and (3-5) Hover, Vertical Flight, and Transition modes with lifting rotors and pusher propeller operational. These models enable vertical flight, hover, and transition region between rotorcraft and airplane flight capabilities.

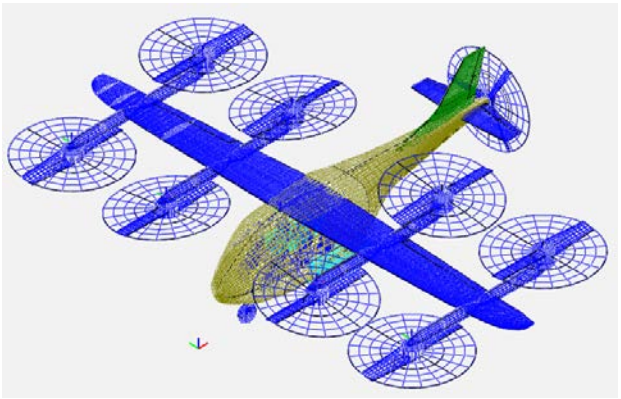
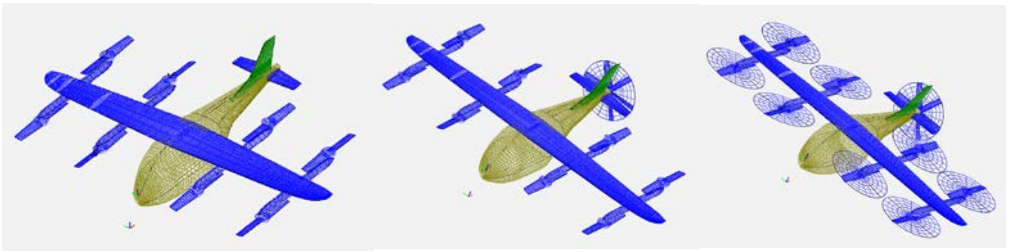


Figure 5. Lift + Cruise configuration



Glider mode | Cruise mode | Vertical, Hover, and Transition modes

Figure 6. Three Lift + Cruise configurations used to model five flight modes

3. Model update onboard – general discussion

Another key component of vehicle modeling for contingency management is on-board, real-time, persistent aerodynamic monitoring and model adjustment robust to changing aerodynamics. This consists of recognizing a change in the nominal vehicle aerodynamics as well as determining the necessary aerodynamic model increments to the baseline model. The high-fidelity vehicle aerodynamic model developed offline is utilized to compare the model predictions and the sensed vehicle response. If the model prediction and actual vehicle response significantly differ, then the flight computer is alerted to the possible presence of an aerodynamic deviation from the baseline model. This could, for example, be vehicle propulsor degradation or a control surface failure. After sensing a possible failure state, the vehicle takes actions to identify aerodynamic model updates through application of both conventional and artificial neural network-based system identification techniques, while also maintaining sufficient passenger ride quality. The identified aerodynamic model increments are then used to inform adaptive flight control logic, provide a vehicle capability assessment, and estimate the new vehicle flight envelope.

4. Model flight control design

The aerodynamic model used for control development and failure modeling is derived from a semi-empirical strip theory modeling tool that builds the aircraft by components [33]. This buildup of the aircraft allows for a rapid creation of a dynamic model for non-conventional configurations and provides fast calculations of the six-degree-of-freedom aero-propulsive forces and moments. The modeling approach, based on first principles of aerodynamics and momentum theory, provides a competent aerodynamic model of a complex aircraft when access to high-fidelity wind tunnel or flight data is unavailable.

B. Vehicle Safety

1. Flight envelopes

The runtime safety of the vehicle will not solely be determined by direct analysis of whether or not actuation will allow the vehicle to remain within a set of flight envelopes. Rather, two probabilistic methodologies for assurance of safe vehicle operations that apply to both nominal and non-nominal conditions are introduced:

- Probabilistic assessment of movement of vehicle operational performance into or out of a set of flight envelopes;
- Probabilistic flight envelope estimation when the vehicle shifts from nominal to non-nominal states, or from non-nominal to non-nominal states.

The probabilistic inferences for both of these methodologies are two elements of a larger belief space for the ICM architecture. Tools derived from them feed into a cognitive architecture that inform subsystems across the larger ICM architecture.

First, a set of detectors are developed that provide, with quantified levels of uncertainty, detections that a loss-of-control event is occurring or that the vehicle is approaching the limits of an envelope (whether it is returning to or leaving the bounds of an envelope) [34]. These detectors rely on a variational neural network model, known as a conditional variational autoencoder, to form probabilistic models of the latent variables that correspond to flight dynamics. These probabilistic models are formulated as conditional distributions, with a sub-population defined by the set of envelopes in which the vehicle is currently operating (which can be trivially calculated if the envelope is well-defined). While theoretically, a single conditional variational autoencoder can be created that would capture all the features of flight and environmental dynamics necessary to form, it would be infeasible to train and use such a neural network in any flight environment. Instead, the technical approach for describing the belief space is to (1) identify statistical relationships (rules) between the dimensions of observable vehicle measurements, (2) form conditional variational autoencoders to model the latent features of those relationships, and (3) form a belief space based on the conditional joint probability density functions of these latent features. This approach is illustrated in Figure 7.

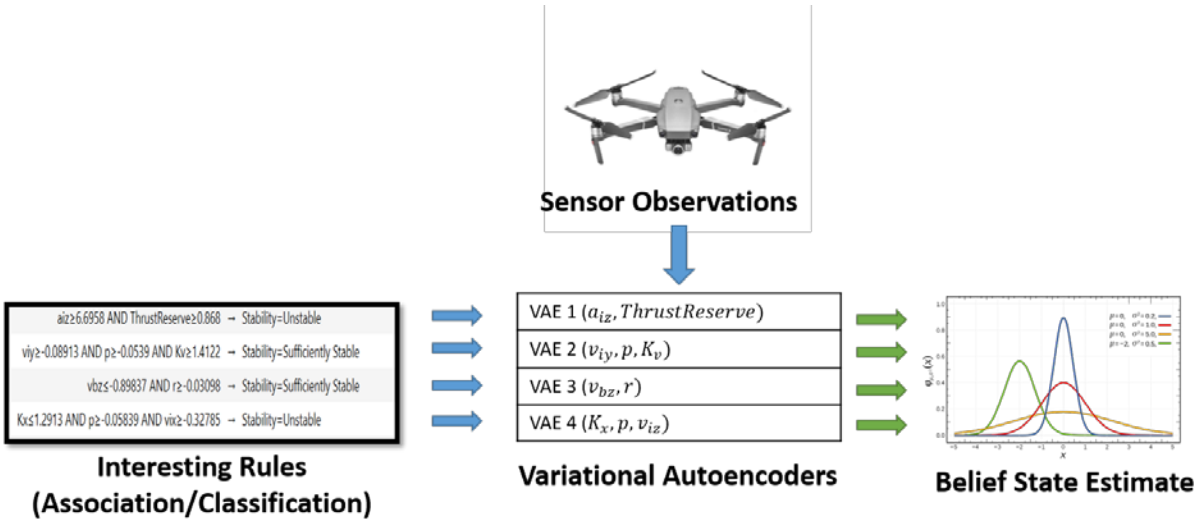


Figure 7. Belief State Estimation with respect to Vehicle Flight Envelopes

This approach for detection has been experimentally validated using empirically defined flight envelopes, defined for nominal flight at design-time and subsequently tested/validated at run-time. To enable contingency management, the vehicle needs a definition of new flight envelopes that are a consequence of off-nominal conditions, such as when operating post-damage, under extreme weather conditions, or when carrying disruptive cargo. For the ICM concept of operations (CONOPS), these envelopes must be estimated in real-time. The approach leverages physics-informed neural network models and sampling to speed up a modern framework for flight envelope estimation. Specifically, neural network models were trained, with loss functions derived from Lagrangian and Hamiltonian mechanics, to reduce sampling times for determining trim sets and control effector inputs, as well as increasing the accuracy with which reachable sets can be estimated [35]. Illustrated in Figure 8, a set of new sampling tools and reachable set estimation methods using the integrated simulator described in Section III will be experimentally validated to demonstrate their utility in real-time operation.

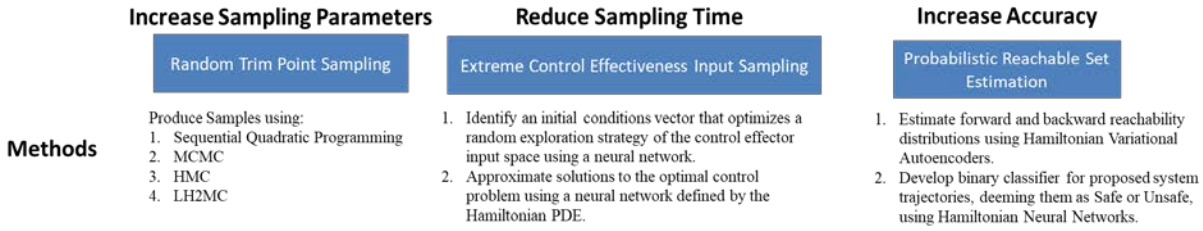


Figure 8. Proposed Methods for enabling real-time Flight Envelope Estimation

2. Reachable sets

The set of reachable states for the vehicle system is defined as all of the states that the vehicle is capable of attaining, given a set of initial states and assumptions under the defined set of allowable control inputs. The subset of those states that satisfy all of the specified safety properties is defined as the safe reachable set of the system. From an execution-centric point of view, system verification can be performed using the forward reachable sets (i.e., given a starting condition, are all reachable sets of the system safe) or the backwards reachable sets (i.e., the set of safe states defined by the specified properties is backwards propagated to see if there is an intersection with the set of allowable initial states of the system). Either Lagrangian or Eulerian methods can be used to propagate the vehicle dynamics in order to perform the necessary computations for either forward or backward reachability. The model used for the vehicle dynamics will determine which method is most efficacious (e.g., number of state variables, non-linearity etc.). Exploration of assumptions, uncertainty modelling, effects of linearization, and over-approximation techniques on the vehicle model will enable the use of safe reachable sets to generate safe trajectories in both an offline and potentially online fashion.

3. Assured containment

Assured containment refers to the concept by which a vehicle is guaranteed not to exit a prespecified geographical area. Assured containment is not simply geofencing, as it requires that a significant safety argument be present as to why the vehicle will be range limited. For instance, an assured containment system should have no single point of failure, that is, it should not be solely dependent on a single source (e.g., GPS) for localization. Additionally, it should have an independent means of mitigating an impending boundary violation, and thus should not depend on the autopilot for flight termination (or other mitigation actions). Furthermore, the algorithms used to ensure that the vehicle is remaining within the boundary should be provably correct. Assured containment will enable the vehicle to bound its geographical state to a safe set.

V. Mission Execution

The mission execution part of the architecture includes robust adaptive control laws as the fundamental building block. The objective of these is to provide robust performance for trajectory following subject to atmospheric disturbances, vehicle model uncertainty and potential faults and failures. The ability of the robust adaptive control laws to handle faults/failures and atmospheric disturbances while providing robust performance directly informs and impacts vehicle capability assessment. One of the outcomes of the feedback decision loop is a requirement for onboard trajectory planning. Informing robust adaptive control and trajectory planning is another safety-critical function – collision avoidance.

A. Vehicle Control

Urban Air Mobility (UAM) operations are based on vertical takeoff and landing (VTOL) aircraft that transition from a rotor or thrust borne flight to a more efficient wing or lift borne flight. UAM vehicles tend to fall into one of the three general classes of VTOL configurations: tiltwing, where the wing and propulsion rotate as a unit from a vertical to horizontal position, a tiltrotor where the wing is fixed and the propulsion units rotate between a vertical and horizontal position, or Lift+Cruise where both the wings and propulsion units are fixed and there are separate propulsors that provide lift in hover and thrust in forward flight. Common across all platforms is the use of propulsion units and aerodynamic surfaces as control actuators. Their combination in different flight regimes provides redundancy in control force and moment generation however their individual effectiveness and energy requirements vary as the flight envelope is traversed.

Transitioning VTOL aircraft must operate in three general flight modes: hover, forward flight, and transition. The first two are well understood flight regimes but operationally present two very distinct modes of flight. Transition, a less well understood regime, must seamlessly stitch the two flight modes together taking into account the changing aerodynamics of the aircraft as well as the dynamic nature of control force and moment production from available actuators.

The combination of the transitioning aerodynamics and control actuation make VTOL aircraft, in general, very complex. Compounding these complexities is the plethora of configurations being proposed for the UAM market, each with their own unique aerodynamic properties and available control effectors. From the outset, it is easy to see that an automatic control system is needed to improve vehicle flyability and provide foundational capabilities to build on as we transition to autonomous operations.

Robust Uniform Control architecture is utilized as a baseline control for VTOL aircraft. Robust Uniform Control is a configuration independent framework which unifies the control design across all flight regimes, and, thus, provides a uniform set of control commands throughout the entire flight envelope [36]. The control architecture builds upon the foundational principals of Robust Servomechanism Linear Quadratic Regulator (RSLQR) [37], but formulates the problem using general accelerations as the commanded input which can then be mapped to the physical control actuators, i.e. rotors and surfaces [38]. However, for transitioning aircraft, the relative degree of the acceleration to the control inputs change as the aircraft transitions between flight regimes. This is handled by introducing the roll and pitch angles as pseudo control inputs [39], thus retaining a relative degree of one across all flight regimes. The formulation enables a consistent control strategy over the entire operating envelope of a VTOL aircraft.

The baseline algorithm unifies the control design over all flight regimes providing an intuitive set of control commands to the operator or guidance system that are consistent across the operating envelope. The general structure of the controller – how redundant control effectors and changing actuation methods are handled as the aircraft transitions between flight modes – makes it particularly suitable for UAM aircraft. The baseline controller is augmented with a control allocator [40] and an adaptive algorithm, Affine General Inverse (AGI) and L1 adaptive control, respectively.

The novel linear control allocation AGI algorithm [41] is utilized to optimally allocate the effectors (e.g. rotors and aerodynamic surfaces) to achieve the desired controller moments across both the lateral and longitudinal controllers for all phases of flight. The allocation algorithm is an iterative generalized inverse method that accommodates affine moments, preferred control, effector position and rate limits to achieve weighted optimal control, $\vec{u}_{opt}$, by minimizing:

$$\vec{u}_{opt} = argmin_{\vec{u}} (\vec{u} - \vec{u}_0)^T W (\vec{u} - \vec{u}_0) \quad \text{such that}$$
$$\vec{m} = \vec{m}_0 + B\,\vec{u} \quad \text{subject to}$$

$$\vec{u}_{upr} \geq \vec{u} \geq \vec{u}_{lwr} \quad \text{(effector position limits)}$$
$$\dot{\vec{r}}_{upr} \geq \frac{d}{dt}\vec{u} \geq \dot{\vec{r}}_{lwr} \quad \text{(effector rate limits)}$$

with the desired controller moments $\vec{m}$, affine moment intercept $\vec{m}_0$, the linear control effectiveness matrix B, the preferred control $\vec{u}_0$, and the positive definite (or semi-definite) weighting matrix W. Moreover, the effector position limits are allowed to be one-sided (e.g., $\vec{u}_{upr} \geq \vec{u} \geq \vec{u}_{lwr} \geq 0$ or $0 \leq \vec{u}_{upr} \leq \vec{u} \leq \vec{u}_{lwr}$) or dual sided limits (straddle zero).

Incorporation of this allocation algorithm has shown to be highly successful in satisfying the effector position and rate limits, while obtaining desired moments with a weighted control magnitude equivalent to or smaller than other allocation algorithms (including the industry standard quadratic programming method). Typically, controllers are ignorant on whether requested moments are physically achievable. For controller moments which are achievable, the current algorithm yields allocations which generate the moment within both effector position and rate limits. For controller moments that are not achievable, the algorithm provides the optimal allocation for the maximal achievable moment along the desired moment direction.

For autonomous vehicles, optimal effector allocation point solutions for the desired controller moment within effector position and rate limitations are essential to dealing with effector saturation issues that typically hinder a wide variety of control architectures. While the AGI algorithm yields point solutions within limits, the AGI solution also provides optimal allocation solutions in a neighborhood of the desired moment. This is accomplished as the AGI algorithm solution is of the form:

$$\vec{u}_{opt} = \vec{c}_0 + P(\vec{m} - \vec{m}_0)$$

with the desired controller moments $\vec{m}$, affine moment intercept $\vec{m}_0$, the linear, the generalized inverse P, and a fixed affine offset vector $\vec{c}_0$ in the null space (kernel) of the control effectiveness matrix B. This affine solution generated by using an offset vector and generalized inverse matrix enables direct computation of solutions around $\vec{m}$ (e.g., changes in moment magnitude and direction) and it is then trivial to ensure the solutions to neighboring moments are within effector position and rate limitations.

Autonomous decision processes stand to benefit greatly from the additional neighborhood solution information. This additional information includes a local analytic rate of change of control allocation with change in moment and a local volume of attainable moments within effector limits reachable from the current state. This information could be used to yield a local temporal and spatial controllability assessment or enable intelligent modification of desired dynamics by change of desired moment direction/magnitude. Also, a new control paradigm is possible whereby the local attainable moments at the current time step are used to intelligently specify future control based on current flight vehicle capability. For example, a future control regulator command (difference between the desired signal and the current signal), could be intelligently chosen to minimize future command error based on the current local attainable moments set.

The L1 adaptive control augmentation is used due to its flight proven capability to provide robust performance in the face of faults, failure and large model uncertainties while providing a predictable response [42]. This particular adaptive control formulation is considered certifiable under current DO-178C standards [43]. L1 adaptive control has been flown on a number of various UAV configurations, including NASA AirSTAR T2 aircraft [44, 45], as well as the National Test Pilot School variable stability Learjet [46]. Since L1adaptive control augmentation spans the entire flight envelope and will be central in adaptive new control strategies that might be required for vehicle's safe mission execution in the event of significant fault or failure, current implementation is focused on linear parameter varying and switching systems formulations [47, 48]. The advantaged of direct adaptive control is that it will attempt to maintain a set of desired dynamics and L1 adaptive control, in particular, exhibits graceful performance degradation if the nominal dynamics are not possible to recover [45]. This ability to keep the vehicle in stable flight provides time for other systems to determine new vehicle dynamic capabilities and an opportunity to modify control strategies that

are commensurate with the new capabilities. Dynamic control strategy updates as a function of vehicle capability is being actively pursued.

B. Trajectory Planning

In the state of the art, airspace has been sufficiently sparsely populated that missions are planned well in advance and are based more on conservative rules of separation than on vehicle dynamical considerations. Once planned, the mission's trajectory must be followed, in the sense that if dispersed from the planned trajectory, it is the pilot's responsibility to rejoin it.

In UAM, more crowded airspace and rapidly changing airspace use demands force each vehicle's planned trajectory to contribute materially to the joint performance goal of maximizing the margin for safely inserting more traffic into the airspace. Furthermore, the notion of trajectory following becomes moot. Aside from the fact that aircraft are always dispersed from their command trajectories, due to disturbances and uncertainties, ongoing changes in airspace utilization requirements force frequent, perhaps continual, to the command trajectories of airspace participants. What's wanted is a system that generates command trajectories from the vehicles' current states.

The trajectory planning element of this architecture translates mission requirements expressed in terms of probabilistic moments into trajectories for system models in which informational, environmental, and dynamical uncertainties are expressed in terms of randomly distributed parameters. The solution process optimizes and imposes constraints on the expectation performance of a collection of random samples from the parameters' distribution. As such, it generates a set of "neighboring" optimal trajectories from which the desired mission trajectory is drawn as a function of the current estimate of the system parameters' values.

The estimates of moments appearing in the problem accrue incrementally, as new sample trajectories are added to the sample set. Because of this, probabilistically optimized and constrained mission trajectories can be computed at mission time for current estimates of system parameters. The optimal trajectories can also be used to populate a neural network which can be rapidly queried when necessary.

The problem formulation for the trajectory planning component emphasizes simplicity. With the exception of the randomly varying parameters, the system representation is deterministic, employing Monte Carlo sampling to build up higher-level moments. Each sample trajectory is explicitly tied to a trajectory that satisfies mission requirements for the current mission moment estimate. Additionally, the trajectory ensembles are computed for temporally discretized models that explicitly respond to sensed errors at the conclusion of each time step, so that the trajectories are corrected by feedback.

C. Collison Avoidance

The ability to navigate through a congested urban air environment in an autonomous collision-free manner is an essential requirement of any autonomous urban air vehicle system. Challenges arise due to the close proximity of vehicles, the intent or goal of one vehicle is unobservable to other vehicles, and the decentralized sequential decision process that must occur most likely without the exchange of information. It is fortunate the multi-agent collision avoidance problem of the autonomous urban air vehicle is shared and central to many robotic applications such as multi-agent coordination, autonomous navigation through human/robot crowds and computer crowd simulation and that the tools developed for these in a 2D environment potentially could be expanded to the urban air 3D environment. Moreover, these tools address similar challenges to collision free autonomous flight. Specifically predicting the other agent's motion as well as anticipating interaction patterns and doing so in a computationally tractable manner for real time implementation. One of the most promising use Deep Reinforcement Learning (DRL) to improve baseline policies for collision free navigation in crowds [49]. It is proposed to use such an approach as a starting point to create a viable autonomous collision avoidance system for an urban air vehicle.

The approach of [49] is of particular interest with many desirable features. In it, each agent is aware of only a subset of the other agent's states. The solution is completely decentralized as it would be for the urban air collision avoidance problem. To accelerate learning, DRL does not start from no knowledge but starts with a network trained from a known solution via supervisory learning. This baseline policy is expressed in terms of a value network that encodes the estimated time to a goal, a point further along the undisturbed planned trajectory, given an agent's joint configuration (positions and velocities) with its nearby agents. A bootstrapped estimate of the value of current state, consisting of the sum of a reward and a discounted value of on the next state's value determines the better action to take. Both the reward and the value of the next state, of course are dependent on the action taken. DRL uses an epsilon-greedy version of the policy to explore other options and improve it. To make the approach computationally tractable for real time implementation, DRL offloads the online computation of anticipating interactions to an off-line learning procedure.

Both the known solution and the DRL phase formulate the collision avoidance problem as a Markov partially observable sequential decision making problem where the decision to the change agent's velocity vector, i.e. the action, is determined strictly from the current joint state. Each agent has a set of states that are observable to others (position and velocities) and states know only to the agent (goal, desired velocity). The joint state consists of the agent's states as well as the observable portion of the nearby agent. The known solution is the Optimal Reciprocal Collision Avoidance (ORCA) approach [50] which selects the next velocity closest to the preferred velocity that avoids collisions based on the current joint state. The DRL phase will start with a value network initialized by the training data generated from ORCA consisting of joint state/value pairs, where the value is determined from.

$$V\left(s_{t_0}^{jn}\right) = \sum_{t=t_0}^{t_g} \gamma^{(t-t_0)\cdot v_{pref}} R\left(s_t^{jn}, \pi\left(s_t^{jn}\right)\right)$$

where s_t^{jn} denotes the joint state at *t* and $\pi\left(s_t^{jn}\right)$ is the policy determining action, t_i is the time of any decision point along the trajectory before the goal is reached at t_g . Here, R is the reward assessed over each time interval providing a negative penalty for getting too close or colliding with the other agent, unity if the goal is reached, and zero otherwise. DRL modifies the policy underlying the value function to maximize it at each joint state, leading to a policy that minimizes the time to reach the goal while avoiding collisions. It should be pointed out that this is a departure from more standard approaches such as ICAROUS [51], which predicts potential well-clear violations (aircraft pairs jointly occupying a parametric volume) over a look ahead time, offers guidance in what not to do as in ranges of track, ground speed, vertical speed, and altitude maneuvers that will result in well-clear violations, and if vehicles are not well-clear, optimally chooses from a set of admissible horizontal and vertical maneuvers.

The policy underlying the optimal value function (which DRL seeks) also is determined by the optimal action over a large set of actions *A*, fanning in 3D about from the desired velocity directed towards the goal, as

$$a_k = \underset{a_k \in A}{\mathbf{arg\,max}} \left(R\left(s_k^{jn}, a_k\right) + \gamma^{\Delta t \cdot v_{pref}} V\left(s_{k+1}^{jn}\right) \right)$$

where the current position and action are part of s_k^{jn} and the next action a_k is part of s_{k+1}^{jn}. Also part of s_{k+1}^{jn} is the agent's next position which can be obtained through propagation with a_k and the unknown action of the other agent $\tilde{a}_k$ along with its next position. A tilde is used to reference the other agent. The left hand side results from a recursive relation between $V\left(s_k^{jn}\right)$ and $V\left(s_{k+1}^{jn}\right)$, often used in bootstrapping [52] to create better value estimates at the current state with estimates at future states. In the learning phase, even though V is not optimal, this expression determines action in the learning scenarios.

To fully capture the value function across a 3D state-space $s_k^{jn} \in \mathbb{R}^{20}$, which includes agent sizes, the agents preferred speed and direction to goal, goal position, in addition to all observable states, a fully connected deep neural network with ReLU nonlinearities is used as a function approximator. The value network is denoted by $V\left(s_k^{jn} \mid \theta_V\right)$ where θ_V is the set of weights and biases of the neural network. Here, the joint state dimension is increased by only six over the 2D case.

The reinforcement learning algorithm employs a Double Deep V-Learning (DDV-L) approach which utilize two function approximators $V\left(s_k^{jn} \mid \theta_V\right)$ and $V'\left(s_k^{jn} \mid \theta_{V'}\right)$ to estimate and improve the value function. Here, V serves as the value critic selecting the actions in the learning scenarios, and V' , a network of the same structure, serves as a value target critic, assigning state/value pairs $\left(s_k^{jn}, y_k\right)$, $\left(\tilde{s}_k^{jn}, \tilde{y}_k\right)$ to each state visited as

$$y_k = R\left(s_{k:k+1}^{jn}\right) + \gamma^{\Delta t \cdot v_{pref}} V'\left(s_{k+1}^{jn} \mid \theta_{V'}\right).$$

In DDV-L, a number of mini-batches of encounters are run. Each consisting of sets of random initial conditions, the resulting trajectories defined by V and state-value pairs processed with V' to be added to the network's experience

set. After each mini-batch, a subset of state-value pairs is randomly drawn from the experience set and used to update θ_V. To promote learning, the greedy action defined above is used $1-\varepsilon$ of the time, and the rest of the time, a_k is randomly drawn from A. Parameters in V' are updated periodically to those in V which was shown in [53] to improve stability during the optimization. In addition, network performance evaluated on a set of test cases not included in the learning set to determine convergence. The 3D extension of the DRL phase amounts only to a larger joint state and larger set of actions to choose from using V.

A 3D extension was also required of ORCA in order to create the training data. As stated, ORCA does not directly minimize the time to goal while avoiding collision, it finds the next velocity closest to the preferred velocity that avoids collisions. To clarify, note the relative velocity of the agent with respect to nearby agent is $v-\tilde{v}$. Over the next interval $t\in(t_k,t_{k+1})$, the relative position obeys $(v_k-\tilde{v}_k)t=(p(t)-\tilde{p}(t))+(\tilde{p}_k-p_k)$ if the velocities of both agents remain unchanged. A collision will occur if $\|p(t)-\tilde{p}(t)\|<r+\tilde{r}$, within the interval, or equivalently if the relative velocity resides in the velocity obstacle shown in Figure 9 in 2D and its 3D extension. For each agent pair, if collision is eminent, ORCA determines the course correction for each agent by first determining the smallest relative velocity change to move the relative velocity to the boundary. Let u denote this change and n its unit direction. The velocity change is split equally between agents defining an admissible half plane for the 2D problem and a half space in the 3D problem, where each the next velocity can be chosen, specifically $v_{k+1}\in\{v\,|\,(v-(v_k+u/2))\cdot n\geq 0\}$ and $\tilde{v}_{k+1}\in\{v\,|\,-(v-(v_k-u/2))\cdot n\geq 0\}$. As the number of agents increase, the admissible region for next velocity of each agent is the union of half planes or half spaces, the boundary defines a convex hull. The collision-free next velocity closest to the desired velocity corresponds to a point on the boundary closest to the desired velocity and is solved using the methods described in [54].

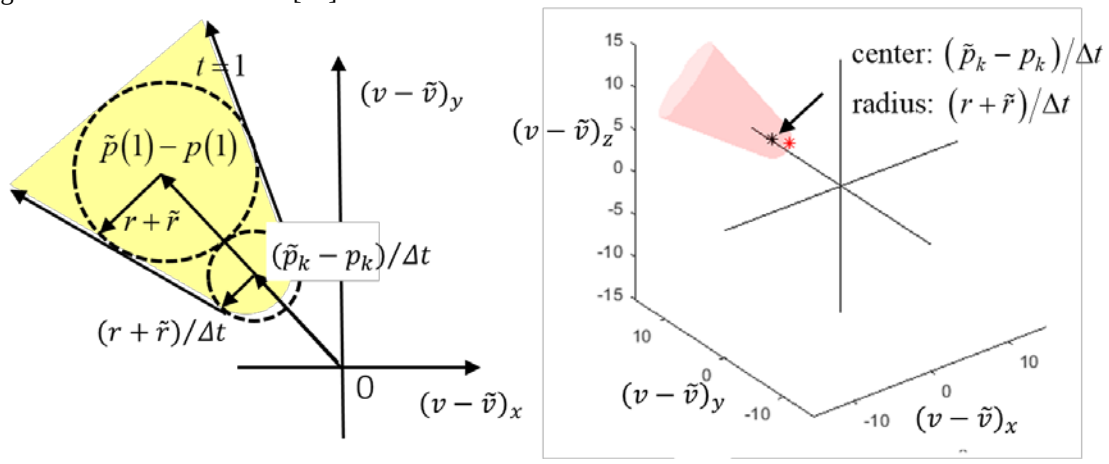


Figure 9. 2D/3D Velocity Obstacles

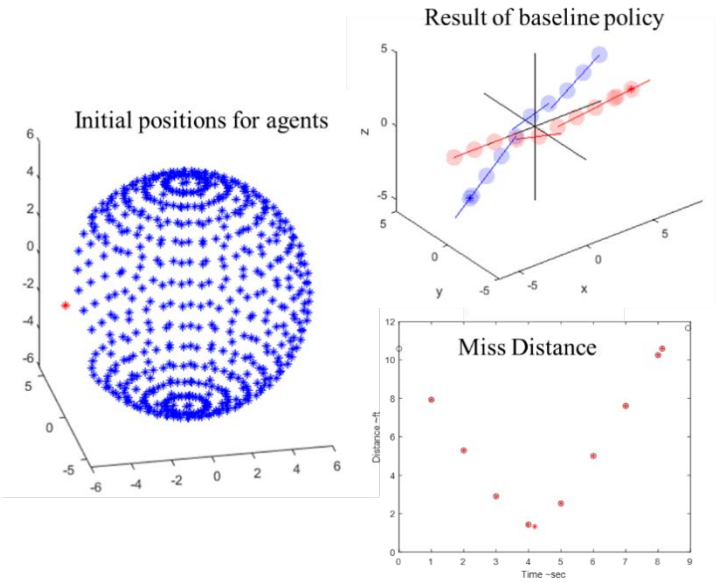


Figure 10. Generation of Training Data

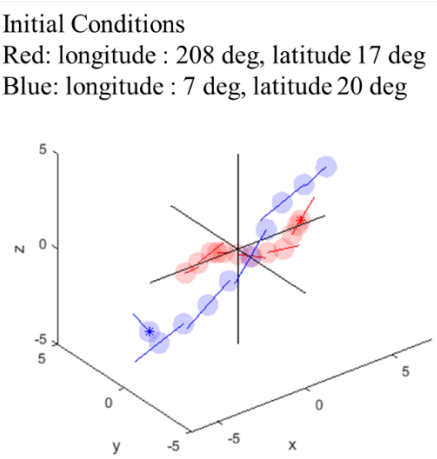
In setting up the initial training data, shown in Figure 10, one agent assumes the same initial position and diametrically opposite goal while the other agent assumes 565 different initial position states all with diametrically opposite goals. Both initially travel at the same speed and with velocity directions unchanged would collide at the origin. A sample encounter is also shown in Figure 10 with ORCA providing the collision free guidance with the agents positions shown every second and the sequential decision making occurring every two seconds. Each agent's velocity directions are shown and theirs goals are designated by asterisks. To assign a value to each state visited, the original value function expression is used to obtain the joint state/value pairs required to train the network. For 565 encounters, the experience of two agents produced 11408 joint state/value pairs to be used as training data. For the initial training process, ORCA generated 157 additional encounters at other initial conditions to serve as a validation set for the initial training process with 3145 joint/states. As a starting point, the neural network's structure is the same as [49] with three hidden layers (150, 100, 100), with 20 inputs/1 output.

It should be noted that rotating any set of initial state positions about the origin essentially produces the same encounter. To eliminate this redundancy, the input to the network, i.e. joint state, is first transformed to an agent-centric coordinate system with the origin at the agents position and the x-axis directed towards the goal. This is always done in training and interrogating the network, so the behavior captured by the network is invariant to coordinate transformation. With these agent-centric joint states/value pairs the initial value network is trained by back propagation to minimize a quadratic regression error

$$\mathbf{arg\,min}_{\theta_V} \sum_{j=1}^{N_{eps}} \sum_{k=1}^{N_j} \left(y_k - V\left(s_k^{jn}; \theta_V \right) \right)$$

where for notational convenience s_k^{jn} is now the transformed joint state.

A sample encounter in Figure 11 shows the performance of the ORCA trained network using the greedy policy defined above. In this initial result, the agents did miss one another. The τ_{CPA} is the time over the one second interval for the closest approach. The result is far from time optimal in that blue agent after avoiding the collision did not take a more direct path its goal. Of course, the results should not be expected to match the behavior of the ORCA algorithm since the objectives differ. However, DRL is expected to improve the baseline and move the network to a more time optimal behavior as was demonstrated in the 2D environment in [49]. Currently obtaining meaningful results during the DRL phase of this approach is still very much a work in progress.



τ_{CPA} (s)	Miss Distance (m)
1.0000	7.2944
1.0000	4.4013
1.0000	2.0098
1.0000	0.2669
0.2214	0.1356
0	1.2792
0	3.6547
0	6.3787
0	9.1380

Figure 11 Initial Neural Net Driven Trajectory

VI. Human Contribution to Safety: Strategies for Resilient Performance

A. Safe Behaviors & Identification of Safe Strategies

Traditional approaches to safety management focus on collection of data describing unwanted states (i.e., accidents and incidents) and analysis of undesired behaviors (i.e., faults and errors) that precede those states. Thus, in the traditional view of safety, safety is both defined and measured by its absence, namely the lack of safety. In extremely high confidence systems like commercial air transport, however, opportunities to measure the absence of safety are relatively rare. Ironically, a critical barrier to measuring safety and the impact of mitigation strategies in commercial aviation is the lack of opportunities for measurement.

While traditional approaches to safety that focus only on minimizing undesired outcomes have proven utility, they represent, at best, an incomplete view of safety in complex sociotechnical domains such as aviation. For example, pilots and controllers successfully manage contingencies during routine, everyday operations that contribute to the safety of the national airspace system. However, events that result in successful outcomes are not systematically collected or analyzed. Characterization and measurement of routine safety-producing behaviors would create far more opportunities for measurement of safety, potentially increasing the temporal sensitivity, utility and forensics capability of safety assurance methods that can leverage these metrics.

Data from structured interviews and voluntary event reports were analyzed to study how pilots and air traffic controllers adjusted their work to sustain safe and efficient operations when encountering expected and unexpected disturbances during Area Navigation (RNAV) arrivals into Charlotte Douglas International Airport (KCLT). Data were examined to identify statements describing four basic capabilities of resilient performance: anticipating, monitoring for, responding to, and learning from disruptions. An initial taxonomy of 19 resilient performance strategies was described.

B. Formalization of Safe Strategies

Safety-producing behaviors that have been expressed as strategies in a natural language may be translated into a more formal environment for analysis, verification, and potential implementation in increasingly autonomous agents. A shifting locus of control from human to automated agents requires that resilient behaviors formerly exhibited by the human agent be evinced by the human-machine team. Novel role allocations between the human and machine require a careful analysis of how the interactions between the agents, along with the actions they themselves take, may give rise to the necessary resilient behavior. The ability to appropriately model agent information sharing, agreement on a common knowledge base, monitoring, and communication are vital to determining whether the compositional behavior of the human-machine team yields the desired safety producing behaviors.

The approach to verifying whether the identified strategies for resilient behavior involves translating natural language strategies into the cognitive architecture framework, Soar [55]. Currently, this translation process is manual in nature. Soar encompasses multiple memory constructs (e.g., semantic, episodic, etc.) and learning mechanisms (e.g., reinforcement, chunking etc.). Soar production rules are expressed in first order logic, which makes them amenable to verification. Finally, Soar is a programmable architecture with an embedded theory. This enables the

execution of Soar models on embedded system platforms and allows the study of the design problem through rapid prototyping and simulation. Additionally, this allows for human-in-the-loop simulation with the use of the XPlane simulation platform, enabling preliminary validation activities.

After expressing the strategies for resilient behavior in Soar, they are then automatically translated into the formal verification framework Uppaal [56]. Temporal logic properties are easily expressed in Uppaal, and the tool can generate and visualize counterexamples. Uppaal also allows the execution of properties to be verified as temporal logic queries to exhaustively check the satisfaction of relevant safety properties. The provably-correct automated translation procedure between Soar and Uppaal outlines the means by which a set of relational rules that represent the communal knowledge of the system (e.g., knowledge base) are captured and analyzed.

Because system safety is an emergent property, it is important to consider the robustness of the verification process, specifically the effect of environmental assumptions being violated, or unforeseen inputs being encountered. This relies on examining execution traces which generate the new elements of the knowledge base, and then evaluating them with respect to the strategies being captured. The approach to formal verification focuses on formally checking whether the knowledge base is consistent and creates classes and subclasses that allows for the generalization of a particular strategic instance.

For further detail on the approach to establishing resilient behavior in aviation systems with increasingly autonomous systems can be found in [57].

VII. Summary

An approach to intelligent contingency management and associated high-level architecture for UAM-class vehicles is proposed. The architecture and associated functionality would allow the vehicle to safely achieve its mission by flying from pt. A to pt. B under all vehicle-allowable weather conditions, in a high-density airspace complex urban environment, while reacting appropriately to off-nominal situations and contingencies without direct human control. This effort is exploring contingency management that expands the coverage of possible contingencies, to include possible unanticipated situations, without exclusive reliance on highly prescribed, rule-based approach. Intelligent approach to contingency management may provide a more general coverage of vehicles and operating environment contingencies in a safe and cost-effective manner. This section summarizes major points about the proposed approach.

The proposed high-level architecture is essentially divided into two parts. First vehicle capability to perform its predefined mission has to be assessed. This involves understanding vehicle current state and predicting that state into the future. The vehicle current and predicted future state are fed into mission execution which controls all aspects of vehicle flight. There is a feedback loop back from mission execution to vehicle capability assessment since the manner in which the mission is executed creates requirements on vehicle capability.

The new class of UAM vehicles offer significant opportunities in design but come at a cost of much higher degrees of complexity with many more interacting factors that result in difficult nonlinear aerodynamic responses and factor interactions. In this case, building appropriate models for flight dynamics simulations is a central component in supporting design and analysis of these new vehicles. Advanced computational tools will be a critical element, especially in early design stages, to capture the complex aerodynamics. Application of conventional experimental methods applied to this new problem are likely to miss significant aerodynamic interactions and nonlinearities. To address this issue, NASA is working to develop an advanced modeling approach called RAM that provides an automated, run efficient, and statistically rigorous process for testing and modeling complex vehicles.

Part of vehicle capability assessment is vehicle safety. The runtime safety of the vehicle will not solely be determined by direct analysis of whether or not actuation will allow the vehicle to remain within a set of flight envelopes. Rather, two probabilistic methodologies for assurance of safe vehicle operations that apply to both nominal and non-nominal conditions are introduced:

- Probabilistic assessment of movement of vehicle operational performance into or out of a set of flight envelopes;
- Probabilistic flight envelope estimation when the vehicle shifts from nominal to non-nominal states, or from non-nominal to non-nominal states.

The probabilistic inferences for both of these methodologies are two elements of a larger belief space for the ICM architecture. Tools derived from them feed into a cognitive architecture that inform subsystems across the larger ICM architecture.

The dynamic complexity of the UAM configurations, the newness of propulsion technology and operation in highly complex, densely populated environment requires high performance control laws with robust fault tolerance that would maximize the range of failsafe operations. The approach considered in this paper is a baseline trajectory

tracking controller with an architecture that unifies the command inputs and design process in all flight regimes. It is augmented with the L1 adaptive and control allocation algorithms to increase its robustness to disturbances and failures. The uniform command structure simplifies vehicle operation through the use of an intuitive command set that can be used for both autonomous trajectory tracking operations or by a low skilled operator. The control architecture has been used to demonstrate full flight envelope capabilities of a Lift+Cruise class aircraft, including dynamic transition from hover to forward flight and vice versa as well as trajectory tracking of UAM representative trajectories.

In the event that pre-defined trajectory is no longer the preferred way to execute the mission, trajectory replanning must be done onboard. There are multiple ways under consideration of performing trajectory replanning with one approach discussed in the paper. The architecture's trajectory planning element produces the idealized trajectory that forms the basis for the operation of the other elements of the system. Because this element operates with incomplete or corrupted information, a planned trajectory is produced prior to mission outset, but is repeatedly updated as the mission is traversed. Each update comprises a path from the best estimate of current conditions that satisfies estimated constraints and boundary conditions. The solution process for these trajectories is intended to optimize mean system behavior while constraining higher moments.

A collision avoidance strategy motivated by recent work in autonomous navigation for moving through human/robot crowds is under consideration for potential application in an autonomous urban air environment. The 2D formulation has been extended to 3D for an approach that consists of an initial supervisory training exercise from a known solution followed by Deep Learning phase designed to improve the baseline strategy to make it more time optimal while avoiding collisions. A value network was trained on data from the known solution and shown to avoid collision although. Work proceeds on the DLR phase to duplicate the results demonstrated in the 2D environment of pedestrian robotic navigation.

The identification, modelling, and analysis of root causes of accidents and incidents dominate conventional safety management approaches. However, the effect of humans' safety-producing behavior on the overall resilience of the system is often neglected. Additionally, there is a shifting locus of control between humans and automation in emerging operations. Without an understanding of the human contribution to safety, it is difficult to assess the effects of these novel role allocations on overall system safety. Work aimed at identifying safety-producing behaviors is ongoing, and the abstraction of these behaviors into resilient performance strategies for the purposes of formal modelling and verification is a primary objective.

Tool development within the structure of the proposed architecture for ICM is ongoing. Increasingly complex scenarios are being used to evaluate approaches, briefly outlined in this paper, and to ascertain advantages and limitations as the research progresses towards autonomous contingency management.

VIII. References

[1] Gregory, I., Rizzi, S., Wincheski, R., Neogi, N., Siochi, E., "Self-Aware Vehicles for Urban Air Mobility," in AIAA Guidance, Navigation and Control Conference @SciTech Forum, Kissimmee, FL, 2018.

[2] Deloitte ConOps contract https://www2.deloitte.com/us/en/pages/energy-and-resources/articles/the-future-of-mobility-in-aerospace-and-defense.html

[3] https://www.marketsandmarkets.com/Market-Reports/urban-air-mobility-market-251142860.html

[4] NASA Grand Challenge Industry Day slides - https://www.nasa.gov/uamgc

[5] McKinsey Study - https://www.nasa.gov/sites/default/files/atoms/files/uam-market-study-executive-summary-v2.pdf

[6] Booz Allen Hamilton Study https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20190001472.pdf

[7] McMinn, J., "Extension of a Kolmogorov Atmospheric Turbulence Model for Time-Based Simulation Implementation," *AIAA Guidance, Navigation and Control Conference*, August 1997, AIAA Paper 97-3532.

[8] DeLoach, R., "Applications of Modern Experiment Design to Wind Tunnel Testing at NASA Langley Research Center," *36th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 1998, AIAA Paper 98-0713.

[9] Landman, Drew, Simpson, Jim, Vicroy, Dan D., and Parker, Peter, "Response Surface Methods for Efficient Complex Aircraft Configuration Aerodynamic Characterization," *Journal of Aircraft*, Vol. 44, No. 4, July-August 2007. doi: 10.2514/1.24810

[10] Murphy, Patrick C. and Landman, Drew, "Experiment Design for Complex VTOL Aircraft with Distributed Propulsion and Tilt Wing," *AIAA Atmospheric Flight Mechanics Conference, AIAA SciTech Forum*, AIAA Paper 2015-0017, Jan 2015.

[11] Landman, D., Simpson, J., Mariani, R., Ortiz, F., and Britcher, C., "Hybrid Design for Aircraft Wind-Tunnel Testing Using Response Surface Methodologies," *Journal of Aircraft*, Vol. 44, No. 4, July-August 2007. doi: 10.2514/1.25914

[12] Rothhaar, P., Murphy, P. C., Bacon, B. J., Grauer, J., "NASA Langley Distributed Propulsion VTOL Tilt Wing Aircraft Testing, Modeling, Simulation, Control, and Flight Test Development," *AIAA Atmospheric Flight Mechanics Conference, AIAA Aviation 2014*, AIAA Paper No. 2014-2999.

[13] Busan, Ronald, Rothhaar, Paul, Croom, Mark, Murphy, Patrick C., Grafton, Sue, O'Neal, Anthony, "Enabling Advanced Wind-Tunnel Research Methods Using the NASA Langley 12-Foot Low Speed Tunnel," *AIAA Aviation and Aeronautics Forum and Exposition 2014*, AIAA Paper 2014-3000.
[14] Simpson, James R. and Wisnowski, James W., "Streamlining Flight Test with the Design and Analysis of Experiments," *Journal of Aircraft*, Vol. 38, No. 6, November-December 2001.
[15] Brandon, J. M. and Morelli, E. A., "Real-Time Global Nonlinear Aerodynamic Modeling from Flight Data," *Journal of Aircraft*, Vol. 53, No. 5, September-October 2016, pp. 1261-1297. doi: 10.2514/1.C033133
[16] Brandon, Jay M., Morelli, Eugene A., "Nonlinear Aerodynamic Modeling From Flight Data Using Advanced Piloted Maneuvers and Fuzzy Logic," NASA/TM–2012-217778.
[17] Morelli, E. A., "Efficient Global Aerodynamic Modeling from Flight Data," *50th AIAA Aerospace Sciences Meeting*, AIAA Paper 2012-1050, January 2012.
[18] Williams, Brianne, Y., Landman, Drew, Flory, Isaac L., and Murphy, Patrick C., "The Effect of Systematic Error in Forced Oscillation Testing," *Aerospace Sciences Meeting*, AIAA Paper 2012-0768.
[19] Cummings, R.M. and Schütte, A., "Integrated Computational/Experimental Approach to Unmanned Combat Air Vehicle Stability and Control Estimation", *Journal of Aircraft*, Vol. 49, No. 6 (2012), pp. 1542-1557. doi: 10.2514/1.C03143.
[20] Ghoreyshi Mehdi, Cummings, Russell M., "Unsteady Aerodynamics Modeling for Aircraft Maneuvers: a New Approach Using Time-Dependent surrogate Modeling," *30th AIAA Applied Aerodynamics Conference*, AIAA Paper 2012-3327, June, 2012.
[21] Murphy, P.C., Klein, V., Frink, Neal T., and Vicroy, Dan D., "System Identification Applied to Dynamic CFD Simulation and Wind-Tunnel Data," *AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper 2011-6522, August 2011.
[22] Murphy, Patrick C., Klein, Vladislav, Frink, Neal T.: Nonlinear Unsteady Aerodynamic Modeling Using Wind Tunnel and Computational Data. *Journal of Aircraft*. Vol. 54: 659-683, No. 2, March-April 2017. doi: 10.2514/1.C033881.
[23] Frink, Neal T., Murphy, Patrick C., Atkins, Harold L., Viken, Sally A., Petrilli, Justin L.: Computational Aerodynamic Modeling Tools for Aircraft Loss-of-Control. *Journal of Guidance, Control, and Dynamics*. Vol. 40: 789-803, No. 4, April 2017. doi: 10.2514/1.G001736.
[24] Murphy, P.C., Aubuchon, V. V., Busan, Ronald C., Hatke, David B., Weinstein, Rose, "Preliminary Steps in Developing Rapid Aero Modeling Technology," *AIAA SciTech Forum*, AIAA Paper 2020-0764, January 2020.
[25] Murphy, P. C., Simmons, B. M., Hatke, D. B., Busan, R. C., "Rapid Aero Modeling for Urban Air Mobility Aircraft in Wind Tunnel Tests," *AIAA SciTech Forum*, January 2021.
[26] Murphy, Patrick C., Buning, Pieter G., Simmons, Benjamin M., "Rapid Aero Modeling for Urban Air Mobility Aircraft in Computational Tests," *AIAA SciTech Forum*, January 2021.
[27] Simmons, B. M., and Murphy, P. C., "Wind Tunnel-Based Aerodynamic Model Identification for a Tilt-Wing, Distributed Electric Propulsion Aircraft," *AIAA SciTech Forum*, Virtual Event, 2021.
[28] Simmons, B. M., "System Identification for Propellers at High Incidence Angles," *AIAA SciTech Forum*, Virtual Event, 2021.
[29] Busan, R. C., Murphy, P. C., Hatke, D. B., and Simmons, B. M., "Wind Tunnel Testing Techniques for a Tandem Tilt-Wing, Distributed Electric Propulsion VTOL Aircraft," *AIAA SciTech Forum*, Virtual Event, 2021.
[30] Derlaga, Joseph M., and Buning, Pieter G., "Recent Progress in OVERFLOW Convergence Improvements," 2020 *AIAA SciTech Forum*, AIAA Paper 2020-1045, Jan. 2020.
[31] Buning, Pieter G., and Pulliam, Thomas H., "Near-Body Grid Adaption for Overset Grids," AIAA Paper 2016-3326, June 2016.
[32] Nichols, Robert H., Tramel, Robert W. and Buning, Pieter G., "Solver and Turbulence Model Upgrades to OVERFLOW 2 for Unsteady and High-Speed Applications," AIAA Paper 2006-2824, June 2006.
[33] Cook, J., and Hauser, J., "A Strip Theory Approach to Dynamic Modeling of eVTOL Aircraft," AIAA SciTech Forum, AIAA, 2021.
[34] Campbell Jr., N., Grauer, J., Gregory, I., "Loss of control detection for commercial transports using conditional variational autoencoders" *AIAA Scitech 2021 Forum: Intelligent Systems*, Virtual Event, 2021.
[35] Campbell Jr., N., Acheson, M., Gregory, I. "Dynamic Vehicle Assessment for Intelligent Contingency Management of Urban Air Mobility Vehicles" *AIAA Scitech 2021 Forum: Special Session*, Virtual Event, 2021.
[36] Raab, S., Zhang, J., Bhardwaj, P., and Holzapfel, F., "Proposal of a Unified Control Strategy for Vertical Take-off and Landing Transition Aircraft Configurations," AIAA AVIATION Forum, AIAA, 2018. https://doi.org/10.2514/6.2018-3478.
[37] Lavretsky, E., and Wise, K. A., Robust and Adaptive Control with Aerospace Applications, 1st ed., Springer-Verlag, London, 2013.
[38] Harkegard, O., and Glad, S. T., "Resolving actuator redundancy-optimal control vs. control allocation," Automatica, Vol. 41, 2005, pp. 137–144.
[39] Akkinapalli, V. S., and Holzapfel, F., "Incremental Dynamic Inversion based Velocity Tracking Controller for a Multicopter System," AIAA Guidance, Navigation, and Control Conference, AIAA, 2018. https://doi.org/10.2514/6.2018-1345.
[40] Acheson, M. J., Cook, J., and Gregory, I., "Examination of Unified Control Incorporating Generalized Control Allocation," AIAA SciTech Forum, AIAA, 2021.
[41] Acheson, M. J., "Affine Generalized Inverse for Optimal Control Allocation," AIAA Aviation and Aeronautics Forum and Exposition (AIAA AVIATION 2018), AIAA Paper 2018-3902, Atlanta, GA, 2018. doi:10.2514/6.2018-3902.
[42] Hovakimyan, N. and Cao, C. *L1 Adaptive Control Theory*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2010.

[43] Bhattacharyya, S., Cofer, D., Musliner, D., Mueller, J., Engstrom, E., "Certification Considerations for Adaptive Systems," NASA/CR–2015-218702.
[44] Gregory, I.M., Xargay, E., Cao, C., and Hovakimyan, N., "Flight test of an L1 adaptive controller on the NASA AirSTAR Flight test vehicle." *In AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, 2010.
[45] Gregory, I. M., Xargay, E., Cao, C. and Hovakimyan, N., "Flight Test of L1 Adaptive Control Law: Offset Landings and Large Flight Envelope Modeling Work." AIAA Guidance, Navigation and Control Conference, Portland, OR, August 8-11, 2011.
[46] Ackerman, K., Puig-Navarro, J., Hovakimyan, N., Cotting, M.C., Duke, D.J., Carrera, M.J., McCaskey, N.C., Esposito, D., Peterson, J.M., and Tellefsen, J.R., "Recovery of desired flying characteristics with an L1 adaptive control law: Flight test results on Calspan's VSS learjet." *In AIAA SciTech Forum*, San Diego, CA, 2019.
[47] Snyder, S. "L1 adaptive control within Learn-To-Fly." 2019. http://hdl.handle.net/2142/106252.
[48] Snyder, S., Zhao, P., Hovakimyan, N., "L1 control for switching reference systems: Application to flight control," *11th IFAC Symposium on Nonlinear Control Systems NOLCOS 2019*: Vienna, Austria, 4–6 September 2019.
[49] Chen, Y.F, Liu, M., Everett, .M, and How J., "Decentralized Non-communicating Multiagent Collision Avoidance with Deep Reinforcement Learning," Proceedings IEEE International Conference on Robotics and Automation, pp 285-292, June 2017.
[50] Van den Berg, J., Lin, M., and Manocha, D., "Reciprocal n-body Collision Avoidance," Robotics Research, ser Springer Tracts in Advanced Robotics, Springer Berlin, Heidelberg, no. 70 pp 3-19, 2011.
[51] Consiglio, M., Munoz, C., Hagen, G., and Balachandran, S., "ICAROUS: Integrated Configurable Algorithms for Reliable Operations of Unmanned System," 2016 IEEE/AIAA 35th Digital Avionics System Conference (DASC), pp. 1-5, Sept. 2016
[52] Sutton, R.S. and Barto, A. G., *Introduction to Reinforcement Learning*, 2nd ed. Cambridge, MA, MIT Press 2018.
[53] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D., "Human-Level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, pp. 529-533, Feb 2015.
[54] de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. *Computational Geometry: Algorithms and Applications*, Springer-Velag, 2008.
[55] Laird, J., The SOAR Cognitive Architecture. MIT Press, 2012.
[56] Behrmann, A. David and K. Larsen, "A tutorial on UPPAAL" in Formal Methods for the Design of Real-Time Systems (Revised Lectures), Germany, Berlin: Springer-Verlag, vol. 3185, pp. 200-237, 2004.
[57] Neogi, N., and Holbrook, J, "Creating Formal Characterizations of Routine Contingency Management in Commercial Aviation", AIAA Scitech Forum and Exposition, Virtual Forum, January 11-21, 2021.