

Dynamic Vehicle Assessment for Intelligent Contingency Management of Urban Air Mobility Vehicles

Newton H. Campbell Jr.*

NASA Goddard Space Flight Center, Greenbelt, MD, 20771

Michael J. Acheson[†] and Irene Gregory[‡]

NASA Langley Research Center, Hampton, Virginia, 23681

New algorithms will be required to ensure passenger and bystander safety during the expected era of autonomous urban air mobility (UAM) aircraft. This paper examines an approach for assessing the vehicle capability to fly itself and to complete a mission safely. The concepts combine elements of system identification, adaptive control, flight dynamics, envelope predictions, and handling qualities, as well as human pilot intuition. The approach is applied to a simulation of a generic distributed electric propulsion urban air mobility-type aircraft, which was developed under the NASA Transformational Tools and Technologies (TTT) project, Autonomous Systems / Intelligent Contingency Management subproject.

I. Nomenclature

C2	=	Command And Control
CONOPS	=	Concept of Operations
DNN	=	Deep Neural Network
GVS	=	Generalized Vehicle Simulator
HMC	=	Hamiltonian Monte Carlo
HNN	=	Hamiltonian Neural Network
HVAE	=	Hamiltonian Variational Autoencoder
MCMC	=	Markov Chain Monte Carlo
PDE	=	Partial Differential Equation
PDF	=	Probability Distribution Function
RL	=	Reinforcement Learning
RPH	=	Roll, Pitch, Heading
RPY	=	Roll, Pitch, Yaw
UAM	=	Urban Air Mobility
$\odot$	=	element-wise multiplication
$\cdot$	=	time derivative

II. Introduction

The emerging Urban Air Mobility market is being enabled by recent advances in distributed electric propulsion as well as increasing capability of autonomous systems. Typical vehicle configurations proposed for this application are unconventional configurations with multiple rotors and vertical take-off and landing (VTOL) capability. Moreover, many candidate configurations are hybrid VTOL: vehicles that take-off vertically and then transition to wing-borne flight.

There are a number of new technologies, without a reliability track record, in these new configurations under development. Moreover, vehicle dynamics are complicated by three phases of flight, with transition tending to have the most nonlinear dynamics and the smallest stability and controllability margins. In addition, the size of the vehicles and their projected operating environment, makes them more likely to encounter bird strikes or conduct operations

*Senior Principal Solutions Architect, Science Applications International Corporation, NASA Enterprise IT

[†]NASA Aerospace Technologist, Dynamics Systems and Control Branch, member AIAA.

[‡]NASA Senior Technologist for Advanced Control Theory and Application, Associate Fellow AIAA.

significantly impacted by local and micro weather. This combination of factors suggests that these vehicles should have a very robust contingency management capability both to ensure vehicle safe operations and to minimize disruptions to the larger air traffic system due to individual vehicle contingencies. Ref. [1] proposes an approach and an architecture to intelligent contingency management (ICM) that would allow these hybrid VTOL configurations to operate safely and efficiently under both anticipated and unanticipated contingencies. This paper considers a decision process for dynamic vehicle assessment aspect of ICM and discusses necessary tools and possible approaches. In this research, our underlying assumption is autonomous flight, with human interaction limited to specifying high level mission objectives. This allows for full consideration of necessary tools and identification of deficiencies in technologies, thus providing a possible technology research roadmap for increasing autonomous system capabilities beyond the current state.

The paper is structured as follows. Section III describes the problem being addressed and outlines some desired baseline autonomous tools required to assemble higher level autonomous decision making capability. Section IV provides an overview of the Vehicle Simulation Environment within which the Vehicle Flight Suitability Assessment is conducted. Section V discusses the technical approach (at a high level) to an autonomous flight envelope estimation tool. Finally, Section VI summarizes the approach and discusses future research directions of this group.

III. Problem Statement

Robust contingency management capability, especially without direct human intervention, is essential to autonomous flight. A major challenge is the development of tools required to support a resilient decision-making process that can deal with unanticipated and cascading contingencies. The current approach of prescribing a set of rules and procedures, associated with fault trees and common contingencies, is inadequate to address the anticipated operational environment for UAM. It is impractical to try to anticipate every eventuality and produce a set of procedures to follow. Practical contingency management systems must have a decision process that can handle incomplete information and conflicting procedures to produce safe sets of actions that the vehicle can take in response to an unanticipated contingency. This paper attempts to discuss specific challenges associated with a robust contingency management system decision process and identify the tools that would be required to support it.

A. Autonomous Contingency Challenge

The challenge to enable an aircraft to autonomously handle contingencies in real time is a daunting task. Some of the major hurdles to overcome, in logical sequence, for automating contingency management include:

- Recognizing that a contingency event is occurring
- Correctly determining the nature/cause of the contingency based on incomplete and often conflicting information
- Identifying probable impacts or implications of the contingency
- Determining applicable procedures and identifying possible mitigating actions
- Determining deviations from emergency procedures (prior to execution) based on available information or unique compound nature of contingencies
- Performing corrective actions and or emergency procedures
- Reassessing contingency based on updated information and in response to corrective actions
- Updating corrective actions as the situation evolves
- Communicating and interacting with outside entities

Compounding the challenge are the spatial and temporal elements to contingency management. For example, spatially, an engine fire contingency is completely different if it occurs just prior to landing touchdown, immediately after takeoff or during cruise. Moreover, the time element of each contingency varies and may be a primary driver in corrective response and the time available for performing the contingency management actions listed above.

The magnitude of this task may be best illuminated by illustrating the real-time analysis and decisions that a human pilot progresses through when dealing with an unforeseen contingency. Figures 1a-1b depict a flowchart for human response of initial contingency assessment and contingency landing process (respectively). The flow in Fig. 1 is the process outline for decision making of a high-skill human pilot. In this research, we are looking at it as an initial framework for decision-making.

Currently, the autonomous capability to automate this framework does not exist. Therefore, we seek to establish a set of tools (toolkit) for subtasks that we assess in isolation and determine if an agent or agents can do that task. The goal is that the intelligent integration of these tools could begin to replicate key elements of the decision processes. In Section III.B of this research, we outline a set of desired automated tools that would support the larger automated decision process and in Section V, we provide an in-depth description of our approach to research and development for

Flight Suitability Process (Piloted):

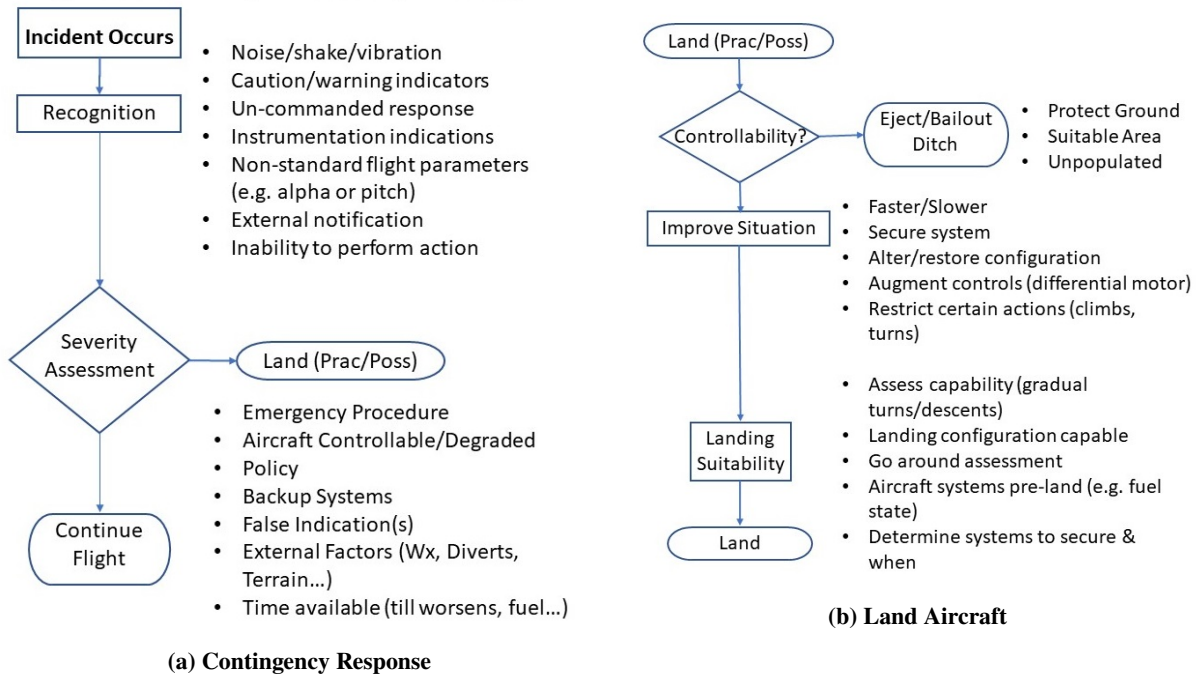


Fig. 1 Representative contingency flow chart

a tool that address a concrete case: flight envelope estimation.

B. Tools to Support Decision Process

This section provides an outline of lower level tools that provide the building blocks for higher level autonomous decision level processes. We introduce this as a starting point to build upon in the future.

Contingency: Onboard sensor indication of system issue (e.g., engine temperature, RPM failure, excessive vibration)

Tool: Confirmation of system issue using alternative sensors, flight parameters, etc.

Process outline: Utilize additional information to confirm contingency. For example, loss of RPM signal could be confirmed with power draw or rotor input while monitoring for aerodynamic response. Develop and utilize confirmation list for each sensor. This list would constitute independent sensors/actions to confirm abnormal sensor indication.

Contingency: Preemptive review if upcoming basic flight maneuver is possible (e.g., turn, climb or descent)

Tool: Utilize onboard models to estimate if desired maneuver can be executed

Process Outline: Incorporate process to categorize upcoming desired trajectory segments into basic maneuver categories: climb/descent, acceleration/deceleration, and turns. Utilize short time horizon look ahead (e.g., 6-10 seconds) if basic flight maneuver can be performed utilizing onboard models. Modeling sources include: linearized dynamic and aerodynamic models, Neural network baseline and effector failure models, dynamically estimated flight envelopes.

Contingency: Desired basic flight maneuver is not performed successfully

Tool: Flight maneuver execution assessment tool

Process Outline: Establish basic flight maneuver categories such as: standard rate turn, half-standard rate turn, nominal climb/descent, pure acceleration/deceleration etc.) Establish performance metrics to determine if executed maneuver was successfully performed with some remaining control authority margin for disturbances.

Contingency: Effector and/or aircraft component failure that effects flying qualities

Tool: Determine landing configuration and flight parameters to make emergency landing

Process Outline: Perform incremental progression towards landing configuration to determine parameters that support necessary maneuvers for safe landing profile. Necessary maneuvers include: descent, turns, descent arrestment, go around, etc. Combine other tools (e.g., predictive models and flight maneuver performance assessment tools) in incremental stages (e.g., speeds) to determine if hover is possible or some forward minimum speed is required for safe landing.

Contingency: Flight parameter achieved but requires out-of-family effector inputs

Tool: Warning/indicator flag assessment for out-of-family inputs required to achieve typical flight maneuver

Process Outline: Utilize onboard linearized models or neural network modeling to provide warning indication that typical flight maneuver is requiring atypical flight effector configuration. For example, prior to commencing approach to landing, a high angle-of-attack is required to maintain level flight. This tool is intended to initiate further assessment/response tools as required.

Contingency: Indications of system degradation/failure that warrant switching to backup or alternate system

Tool: Perform assessment if it is safe to switch to backup system and then execute switch to alternate system. (Presumes other tool provides indicator of need to switch)

Process Outline: Prior to securing primary system, utilize onboard models (linear, neural net etc.) to determine if aircraft is controllable and assess level of degradation expected in performance. Develop assessment metrics and validation processes to determine that switchover was performed successfully (e.g., current measurements on primary and secondary batteries confirm switchover occurred)

Contingency: Time required for upcoming event exceeds available time

Tool: (Suite of) Time until failure timers

Process: Establish a suite of timers that process time until failure. Some examples of timers include:

- Time required for current trajectory exceeds electrical power available
- Time until unusual attitude encountered less than specified minimum
- Time until system failure (e.g., overheat of motor) is less than flight time required

This toolset is intended to characterize the temporal nature of contingencies/emergencies.

IV. Vehicle Simulation Environment

This section briefly discusses the environment for development and evaluation of the ICM architecture and associated decision making tools [1]. The development environment of choice is Matlab/Simulink 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.

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 modeling 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. Due to this capacity to model multiple vehicles, in this paper, we refer to the vehicle simulation software application as the *Generalized Vehicle Simulator*, or *GVS*.

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. The simulation includes mechanisms capable of finding equilibrium manifold and performing model linearization. Full envelope flight control and trajectory generation are part of the tool set. Selectability of appropriate fidelity vehicle models allows for efficient decision making tool development and evaluations.

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

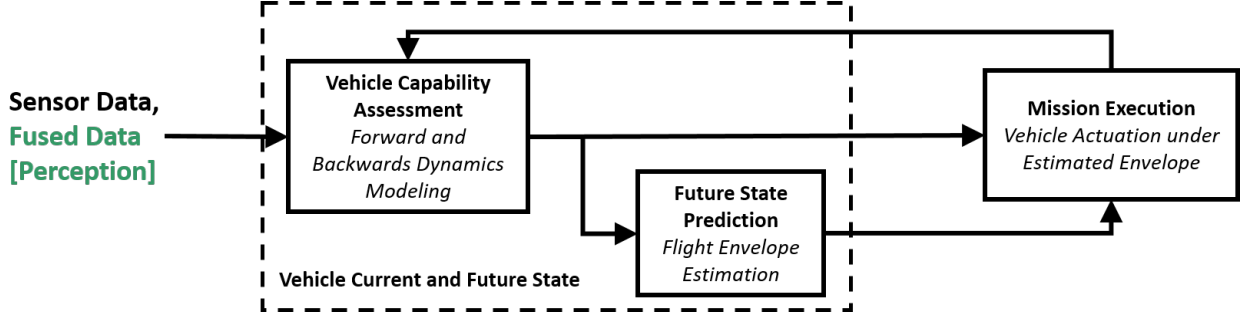


Fig. 2 Estimation of the flight envelope under off-nominal states is enabled by model-building and assessment at every step of the ICM process.

environments. Our intent is to use this executable and library for large-scale parallel simulation on high-performance computing environments at NASA Langley Research Center. Simulation parameters can be varied for each batch run. This capability is currently being used for neural network training through simulation.

V. Technical Approach — Flight Envelope Estimation

In the context of the required tools outlined in the above section, we present a detailed description of one of the key building blocks — flight envelope estimation. We study and develop tools for flight envelope estimation for an aircraft operating in an off-nominal state. Our principal assumption is that the aircraft is operating only at the control of one or more artificially intelligent command-and-control (C2) agents. These agents have access to a pre-computed set of flight envelopes for nominal operations. The tool addresses two categories of in-flight contingency: the aircraft has either shifted from a nominal state to an off-nominal state (e.g. vehicle damage, weather, faulty operation, etc.), or the aircraft has shifted from an off-nominal state to another off-nominal state (e.g. additional damage, recovery, change in weather, etc.). The problem of estimating the flight envelope fundamentally is solved by recomputing, with quantified accuracy, the constraints of system operations through approximation. Like our other dynamic vehicle capability assessment tools, estimation needs to occur in real-time. The concept of operations (CONOPS), illustrated in Figure 2, is for a change in state to be recognized by other agents on the vehicle, followed by a real-time estimate of the new flight envelope. During vehicle capability assessment, we capture, sample, and interpolate vehicle sensor data and form real-time models of the dynamics of the vehicle. These models are leveraged both to aid in the interpretation of dramatic shifts that necessitate a contingency, as well as to inform the flight envelope estimation tool of when the vehicle has been driven to a new qualitative state. A flight envelope estimation occurs through a parallel analysis of probable future states and prior states. When a set of probabilistic flight envelopes can be estimated with sufficient confidence, mission execution occurs only for operations that fall within the bounds of these estimated envelopes.

Beyond the problem of estimation, several studies have thoroughly characterized the flight envelope in ways that we will explore for our concept of operations (CONOPS). For example, studies in carefree maneuvering assess the response of the vehicle to control inputs using piecewise linear approximation in order to ride the flight envelope boundary [2]. This approach captures significant nonlinearities in vehicle response and provides a nonlinear correction for limit parameters. However, an evaluation between linear and non-linear models can result in significant optimization problems. Furthermore, a generalized approach would require a higher-fidelity nonlinear model to account for relevant couplings in its responses. The authors of Ref. [2] do hypothesize that the advantages presented by neural network models may be a good fit for such nonlinear correction.

Neural networks have previously been proposed as potential non-linear models for many aspects of the flight envelope estimation problem [3–6]. When implementing these methodologies in real-time, however, such approaches fall prey to the standard pitfalls of deep learning: insufficient data, lack of computational power, inexplicable inference, uncertain/arbitrary convergence criteria, and unpredictable training times. Recent studies have taken steps to reduce the necessary computational power and speed of estimation using neural networks and execute flight estimation in real-time [7, 8]. While successful in fast estimation, each of these methods fall prey to at least three of the five aforementioned issues.

Aside from neural networks, recent frameworks have sought to apply the level set method to determine the safe flight envelope, formulating the problem in terms of the reachable sets [9, 10]. These methods derive very good solutions, but

require significant computation and are only sufficient for offline estimation. To address the problem in online scenarios, we start by formulating our approach to flight envelope estimation after one of these frameworks. We reconstruct the flowchart for flight envelope estimation from Ref. [10] in the context of several new techniques that we propose to characterize for this domain in Figure 3.

Briefly, let x represent a state vector, $\delta(t)$ represent admissible input vectors at time t , and $\dot{x} = f(x, \delta(t))$ and $\dot{x} = -f(x, \delta(t))$ represent the forward and backwards dynamics of the aircraft, respectively. The estimated probability distribution of the dynamics are computed by Monte Carlo (MC) analysis using our ICM Generalized Vehicle Simulator (GVS), described in Section IV. A trim point is selected from a computed sampling space, starting with a random point in the space. A heuristic search using control input samples is then performed to explore the flight envelope in a particular direction, resulting in a small candidate set of optimal controls. After a flight trajectory simulation with these trim points and control inputs, the state at time horizon T_f is stored as a single sample. The model of these dynamics is updated as the aircraft is flying a particular trajectory toward a time horizon T_f . A sufficient number of samples (some empirically established threshold) are generated and interpolated to form a continuous trim set. The probability density functions (PDFs) for the forward and backwards reachable set (denoted $f_{x_f}(x)$ and $f_{x_b}(x)$, respectively) are then estimated and applied to the following membership function:

$$\mu_{\bar{E}}(x) = \frac{f_{x_f}(x)f_{x_b}(x)}{\max_{x \in \mathbb{R}^n} f_{x_f}(x)f_{x_b}(x)} \quad (1)$$

Note that calculating the forward reachable set is non-trivial. When the value of $\mu_{\bar{E}}(x)$ is above a certain threshold, this indicates that a set of flight trajectories is somewhere in the intersection of the forward and backward reachable sets, and thus, that the aircraft can operate within an envelope defined by this intersection.

In development of our flight estimation tools, we focus particularly on the steps of this approach that, as of the time of this writing, require this process to be done offline. In the framework from Ref. [10], sampling and optimization play large roles. First, the framework samples the trim points by setting up a constrained optimization problem and acquiring a solution using a sequential quadratic programming solver. This makes exploration of the trim set limited, based on the number of variables that affect trim and requires significant time to compute. Then, the framework optimizes a weighted sum of control effectiveness, using rates that are described in three translational and three rotational directions. By generating control input samples that explore the command region in a particular direction, based on a random vector, the framework circumvents solving the optimization problem directly, but introduces some uncertainty with respect to search completion times. Finally, to compute the membership function, the PDFs for the forward and backward reachable sets are estimated using kernel density estimation (KDE). Practically, this limits the dimensionality of the estimators and assumes a forward reachable set that has been estimated with high-confidence. In the remainder of this section, we will discuss our approach to leveraging recent advances in machine learning that will mitigate these issues.

A. Physics-Informed Machine Learning for Flight Envelope Estimation

The sampling and estimation methods of the framework from Ref. [10] suffer from uncertain convergence times and solutions. In ICM, we plan to explore several techniques that are guaranteed to produce a solution (if one exists), even if sub-optimal, with real-time speed. Sampling and estimation are two extremely important concepts in machine learning. In the general case, they are often exponentially slow in problem dimensionality and have no concrete algorithms for declaring convergence. Due to the wide range of problems that necessitate contingency management, we have to be able to address these problems in the general case, particularly for estimating envelopes after arbitrary failures. Thus, our approach to a new implementation of the framework will be to provide methods for a common and general case: sampling and estimation from analytically described distributions using constrained conditional models, with a focus on physics-informed neural networks.

In recent years, a practical flaw in deep learning has been revealed that limits its capacity in CONOPS such as contingency management. The flaw empirically demonstrates drifts in accuracy of inference, even when neural networks receive inputs that resemble the training set. Some of the best examples of this were illustrated in previous work that used standard feed-forward deep neural networks (DNNs) to model both the canonical two-body problem and pendulum kinematics [11]. In these scenarios, the predictions of trained, standard, feed-forward neural networks grew in error when attempting to model these simple physics problems. However, the predictive capability of a standard neural network architecture showed significant improvement after explicitly introducing physical symmetries and constraints during training. To better frame this discussion for our approach, we formulate the problem in terms of Hamiltonian

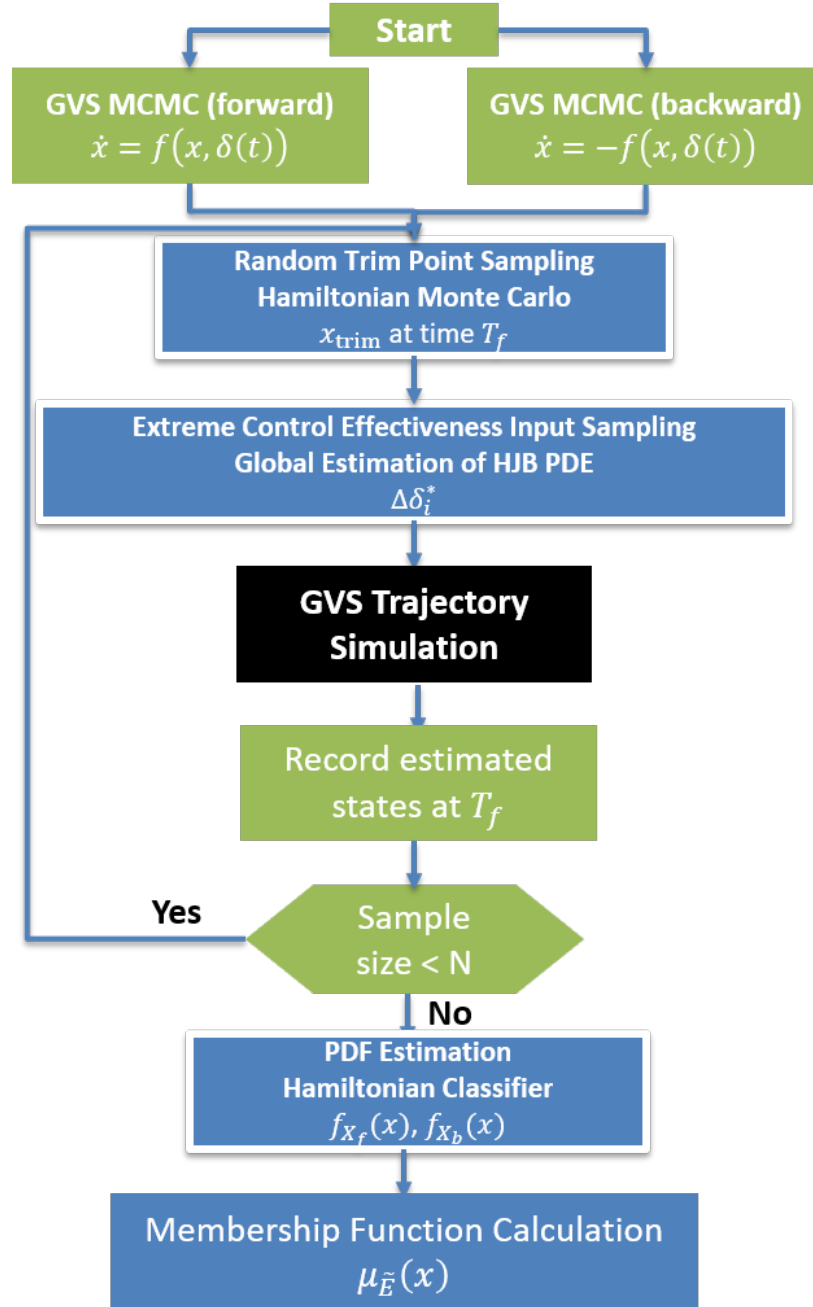


Fig. 3 ICM Implementation of Flight Envelope Estimation for Unstable Over-Actuated Aircraft (boxes in blue are the focus of research for ICM)

mechanics. Hamilton’s equations describe the time evolution of a system as:

$$\frac{dp}{dt} = -\frac{\partial \mathcal{H}}{\partial q} \text{ and } \frac{dq}{dt} = \frac{\partial \mathcal{H}}{\partial p} \quad (2)$$

for the set of canonical coordinates (q, p) representing generalized velocities and conjugate momenta and $\mathcal{H}$, the Hamiltonian equation representing the total energy of the system. If the Hamiltonian is time-independent, then its value is constant:

$$\frac{d\mathcal{H}}{dt} = \frac{\partial \mathcal{H}}{\partial q} \frac{dq}{dt} + \frac{\partial \mathcal{H}}{\partial p} \frac{dp}{dt} = 0 \quad (3)$$

For mechanical systems, this constant typically represents the sum of the kinetic and potential energy of the system that produced the generalized velocity and conjugate momenta. In this research, we focus on using the Hamiltonian for sampling, constraining neural network predictions that help in estimating the current envelope, and directly estimating partial differential equation (PDE) solutions.

First, to address sampling issues, we extend recent work from Stanford, Google AI Perception, and Google Brain in extending Hamiltonian Monte Carlo (HMC) sampling [12, 13]. To review, let w represent a sampling proposal distribution, satisfying some set of mild conditions. Starting from an initial sample (state) x_0 , at each time step t , we sample a new candidate $x' \sim w(\cdot|X_t)$ from the understood state space X_t . In Markov Chain Monte Carlo (MCMC), a Markov Chain detailed balance can be forced using Metropolis-Hastings accept-reject rules [14, 15]. When proposals are drawn from w , the acceptance probability is calculated as:

$$A(x'|x_t) = \min\left(1, \frac{P(x')Q(x_t|x')}{P(x_t)Q(x'|x_t)}\right) \quad (4)$$

If $A(x'|x_t)$ is above a threshold, MCMC accepts x' as x_{t+1} . Otherwise, $x_{t+1} = x_t$.

HMC has recently grown in popularity by extending this concept. By constraining the sample space using an energy function, U , the algorithm characterizes the distribution being sampled in a method aligned with the physics of the system. HMC describes the distribution from which it samples as $P(x) \propto e^{-U(x)}$, treating U like the energy of the physical system. For a momentum vector, p , distributed entirely independently from x , the probability distribution $P(x)$ is computed by approximately integrating the Hamiltonian dynamics on x and p , with $U(x)$ as the potential energy and $e^{\frac{1}{2}p^T p}$ as the kinetic energy. Then, rather than sampling from $P(x)$, HMC samples jointly from $P(x, p)$ and factorizes it:

$$P(x, p) = P(x)P(p) \quad (5)$$

The time evolution of the system is interpolated by leapfrog integration [12], which calculates new position values, using the velocity half a time step ahead of the position, and then updates the momentum. In doing so, the physical dynamics are integrated into sampling in order to traverse long distances while conserving total energy, thus conserving total probability. This allows HMC to make proposals at long distances from the starting point that can be accepted with very high probability.

To compute updates for the leapfrog operator, the L2HMC (Learning to Hamiltonian Monte Carlo) algorithm focuses on learning a parametric leapfrog operator, L_θ , over an augmented state space [13]. L2HMC defines a loss and training method for neural networks to estimate the necessary functions for performing the leapfrog step, as there is often not enough time or data to compute them directly. The authors of L2HMC structure the algorithm around a latent generative model (variational autoencoder) based on this process for sampling, enabling the decoder to make use of a more expressive posterior and the sampler to identify samples that correlated with the energy function much faster than MCMC or HMC approaches. In other research [16], a separate Hamiltonian Variational Autoencoder (HVAE) was created to use the reparameterization trick to calculate unbiased gradients of the evidence lower bound with respect to the parameters of interest, by using an HMC importance sampling to sample from the latent space during decoding. The authors introduce a tempering step that adds inhomogeneity to system dynamics and better permits exploration of isolated modes of the target.

Second, we explore the utility of the Hamiltonian function in formulating the loss function of a neural network to make predictions about aerodynamic control when producing flight envelope estimates. In Ref. [11], the researchers explore the use of the Hamiltonian to create a loss function for a neural network that estimates the parameters of the Hamiltonian as:

$$L(q, p; \theta) = \left\| \frac{\partial \mathcal{H}_\theta}{\partial p} - \frac{dq}{dt} \right\|_2 + \left\| \frac{\partial \mathcal{H}_\theta}{\partial q} + \frac{dp}{dt} \right\|_2 \quad (6)$$

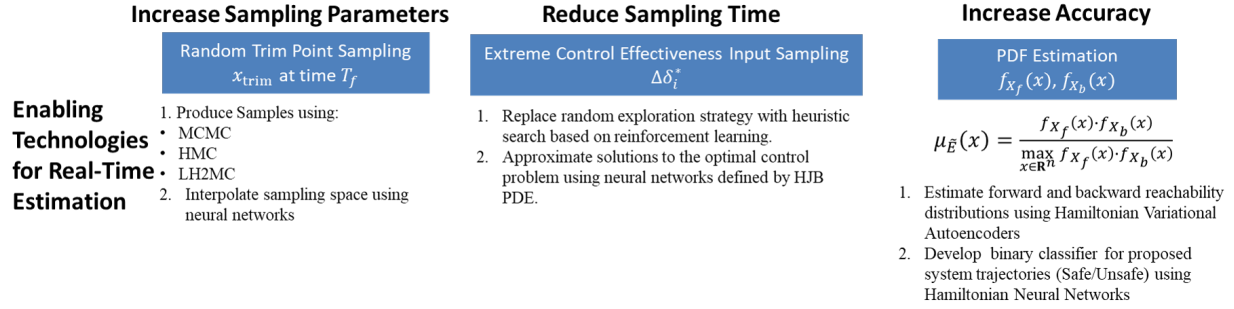


Fig. 4 Summary of enabling technologies from which we will derive tools for real-time flight estimation, numbered by the order in which each method will be used for experimentation.

This loss function is used to train a generative mapping, at time t , from (q_t, p_t) to (q_{t+1}, p_{t+1}) . While the initial experiments in Ref. [11] use this Hamiltonian neural network (HNN) model for simple examples such as the two-body problem and simulated/real pendulums, research in predictions that leverage similar loss functions have increased in complexity over the years. In Ref. [17], a variant of this loss function is used to model the Hénon-Heiles potential. The authors demonstrate that through their Hamiltonian-based loss function, the neural network achieves the ability to predict or respond to chaos in a system. Lagrangian dynamics, as well as other equation-conserving neural networks, have also shown accurate capability in modeling phenomena with well-defined physical constraints [18–21]. Finally, recent work has given guidance on methods of estimating the Hamiltonian that may be extended to time-dependent systems [22, 23].

Third, we explore the possibility of approximating viscosity solutions to PDEs used for envelope estimation (i.e. Hamilton-Jacobi, Hamilton-Jacobi-Bellman, Hamilton-Jacobi-Isaacs) through deep learning. The notion of creating a neural network approximator based on PDEs has been widely explored, even before the age of deep learning [24–27]. More recently, in studies such as Ref. [28, 29], authors identify that a key attribute of a physics-informed network is that PDEs can be solved using relatively small datasets. In Ref. [30], authors explore neural network architectures that produce optimal results for the solution to the Hamiltonian-Jacobi (HJ) viscosity solution. The uncertainty of generative models to produce a generative solution to PDEs has also been explored [31]. This work leveraged convolutional variational autoencoders and a loss function defined by the variational functional of the PDE for training.

We do not anticipate that approximations based on such models will produce exact, actionable inferences to be used in our framework. Instead, we intend to use measurements from these approximations to both narrow search spaces during sampling and to inform uncertainty measurements. In particular, we will explore quantification of consistency and stability properties that can be derived from these measurements to interpret reliability of the estimates over the entire framework.

In summary, we explore the use of Hamiltonian approximation, through deep learning, for flight envelope estimation through:

- Sampling - HMC [12], L2HMC [13]
- Predictions based on system energies - HNN [11, 17], HVAE [16]
- Approximating solutions to the optimal control problem - Neural PDE Estimators [30, 31]

In the following sections, we describe how we will implement and experiment against the GVS to enable real-time operation for the process described in Fig 3. Figure 4 summarizes the enabling technologies that we propose to support this experimentation.

B. Random Trim Point Sampling using Hamiltonian Monte Carlo

The goal of trimming is to find equilibrium points of the system, typically by formulating a constrained optimization problem (that constrains rotational and translational accelerations to zero). In Ref. [10], this problem is solved by interpolating trim points by cubic splines and solving the optimization problem using Sequential Quadratic Programming methods. This solving is performed iteratively for different mass configurations to compare the envelope estimation result. In their work, for efficiency, a small set of vehicle-specific effectors (all-moving wing tips (AMTs), elevons, and pitch thrust vector) were chosen as the control effectors that are used to trim the aircraft. The authors formulate the problem as identifying a trim set, x_{trim} , over time horizon T_f . What is not well understood (or well-defined, in terms of

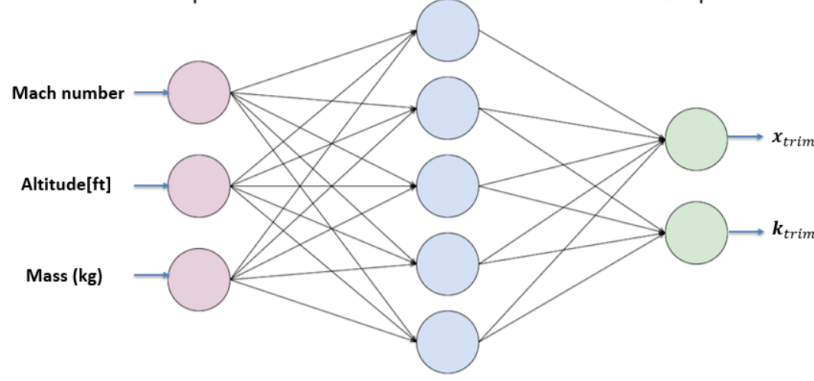


Fig. 5 Neural network for interpolating trim points at varying speeds, altitudes, and mass configurations.

formulation) is if these are sufficient for arbitrarily computing and sampling the trim set.

When trimming, the perceived energy coming from the aircraft sensors, which can be derived from the Hamiltonian, should be consistent over T_f . The first step in our work is to verify this claim empirically. Energy equations for aircraft are historically and consistently well-defined [32–35] and can serve as good tools for model training and comparing to estimates. Next, to identify trim points, we propose applying both the HMC and L2HMC methods to give us the capacity to sample the trim space defined in terms of all available control effectors. We will design experiments to increase the number of control effectors used by each sampling algorithm, describing the performance of how each algorithm mixes over time. The benefits of L2HMC are anticipated to permit the larger set of control effectors to be considered, without significant increase in computational complexity.

For varying altitudes, speeds, and mass configurations, instead of using polynomial interpolation on accepted trim points to define the trim space, we propose the use of neural networks for interpolation of more complex spaces. We will explore simple feed-forward and convolutional neural networks for this interpolation, such as the one illustrated in Figure 5, where x_{trim} is a trim point and k_{trim} is a weighting factor for the trim state. In addition, we can describe a physics-informed loss function based on the known trim constraints, such as requiring the derivatives of all the angles and rates to be equal to 0.

C. Control Input Sampling using Hamiltonian Neural Networks

The framework in Figure 3 replaces the high-dimensional optimal reachability problem, which is computationally infeasible, with Monte Carlo (MC) estimation, which is much more computationally feasible [10]. The framework addresses the sub-optimality of Monte Carlo by proposing a new control input sampling strategy. In our research, we will examine the derived equation in the optimal control framework and sampling strategy, to explore two methods of reducing the overall sampling time.

The control inputs generated by this method are a result of exploring the control input space in a particular direction, based on a random variable $\mathbf{W}$ that is sampled at each time step. $\mathbf{W}$ is defined as a vector with a component mapping to each of the velocities and rates such that $\mathbf{W} = [W_u \ W_v \ W_w \ W_p \ W_q \ W_r]$. In effect, the framework tries to circumvent solving the optimization problem directly, by fixating on specific areas of a command region. But the problem of identifying the best direction is unsolved. In our research, we propose to introduce a way to strategically select $\mathbf{W}$ that shrinks the search time for control effectors. We will leverage a recurrent neural network to construct a reinforcement learning (RL) framework for identifying $\mathbf{W}$ on each iteration of our sampling, turning this process into a more informed heuristic search.

We will explore several continuous reinforcement learning algorithms. For example, we can define the problem in terms of the Deep Deterministic Policy Gradient (DDPG) [36, 37]. In the DDPG formulation, the RL state variable s will be defined as $s = [\dot{x}; x_{trim}(T_f)]$, where $\dot{x}$ represents the known forward and backward dynamics of the aircraft and $x_{trim}(T_f)$ is the sampled trim set described in the previous section, on the time horizon T_f . The action a is the vector $\mathbf{W}$. The formulation of the reward function will be derived from a function of the angular rates. The collective policy will result in an RL agent that guides the control effectiveness input sampling search. We will compare the performance of this reward function to another that is a function of the estimated system change in energy (potential and kinetic). Both formulations will optimize our ability to search the command region and shorten the time for finding a solution.

Because the state information is bound to some Hamiltonian constraints, the Hamiltonian can be integrated into the loss function for the underlying network.

The second method to reducing the sampling time will involve approximation of the solution to the HJB equation for the optimal control strategy, using the neural network approximators for PDEs described in Section V.A. The research goal, here, is to identify the appropriate simulation parameters for generating training data, such that the HJB equation can be approximated with reliable accuracy. Once we sufficiently train these networks on the control effectiveness PDE, we must define a template for statistical verification of these models. We will do so by defining adversarial inputs for the aircraft to determine whether the model infers a solution to the PDE. Not every partial differential equation has an exact solution. The ability of the network to identify when a solution to the HJB equation exists and to produce that solution can be used to evaluate our network. An unstable or inconsistent solution can imply a level of uncertainty about a command region being explored or the system understanding of its own state space dynamics. We will develop adversarial inputs, to evaluate the performance and training of networks that can estimate solutions to the HJB equation.

D. Reachable Set Estimation using Hamiltonian Variational Autoencoders

The notion of the classical flight envelope, in this work, will be extended to the fuzzy set formulation from Ref. [10]. The membership function for this set is formulated as the joint probability of a flight trajectory, x , being within the forward and backward reachable sets. To determine whether or not a trajectory is within the envelope (meaning it is in the intersection of both the backward and forward reachable sets), the sampling described in Figure 4 is used to obtain PDFs that produce the membership function in Equation 1. The KDE algorithm, used in the original work, can suffer from under-smoothing or over-smoothing due to bandwidth selection. While a number of data-driven algorithmic methods have attempted to address the bandwidth problem [38–41], KDE is not meant to be applied to high-dimensional spaces [42]. We explore the utility of variational autoencoders, both because of their capacity for multivariate density estimation, and because of recent efforts to stabilize their estimates using a maximum posteriori (MAP) estimation, to define the objective function [43]. We will develop a loss function based on the regularized version created by MAP and the loss defined by Equation 6. This will produce a robust density estimate that corresponds to the Hamiltonian mechanics of the system, while permitting assessments with respect to changes in the energy profile of the system.

E. ICM Vehicle Simulator Experiments

In order to characterize and compare the practical performance of these enabling technologies in a real-time environment, we will develop experiments comprised of algorithmic runs of the GVS. The main goal of experimentation is to characterize the necessary parameters, training data, and operational data for real-time operation. We will begin experimentation with data collection and training for each of the technologies described in Figure 4. Experimentation will occur in two stages, with a focus on the enabling technologies in Figure 4. At each stage, experimentation of each technology will be executed in two phases: (1) Training and (2) Model Performance Assessment. Figure 6 is a notional architecture of the framework for collecting the necessary data needed for training the first machine learning models described in Figure 4. Figure 6 illustrates how we provide simulation inputs to the GVS, based on the data that is necessary to train a specific set of models. Scenarios in which the vehicle is damaged can also be introduced such that the physics of this damage is appropriately modeled. In some cases, we will want to provide the simulator a trajectory based on an analysis of the necessary training data for our models. Therefore, we develop an informed software component (ML Training Inputs) to select simulation inputs, run simulations in batch, and provide data to the appropriate data repository for offline training.

When training for random trim point sampling models (MCMC, HMC, L2HMC), the GVS will be parameterized with a safe reachability set, determined a priori by SMEs. The AMTs, elevons, pitch thrust vectors, and other aircraft-specific control effectors are decided by a trim sampling algorithm, and accepted or rejected based on the acceptance criteria described in Section V.A. Whether the data point is accepted or rejected, it is provided to the training data set, along with its acceptance probability. This training data is then used to train each of the Monte Carlo samplers, producing a trim set that is verified against new simulation scenarios. When training for extreme control effectiveness method, we execute the algorithm as defined in Ref. [5], randomly defining the random variable $\mathbf{W}$ each time to collect an initial dataset of RL states, actions, and rewards. The reward function is computed by determining how close the direction of $\mathbf{W}$ aligns with the optimal control strategy. RL models with continuous state and action spaces are updated over time to improve this variable selection over multiple time horizons. Finally, we will train neural network PDF estimators without a feedback loop. The intent of this part of data collection is to train and compare models that quickly and accurately estimate PDFs and to assess the utility for in-flight transfer learning.

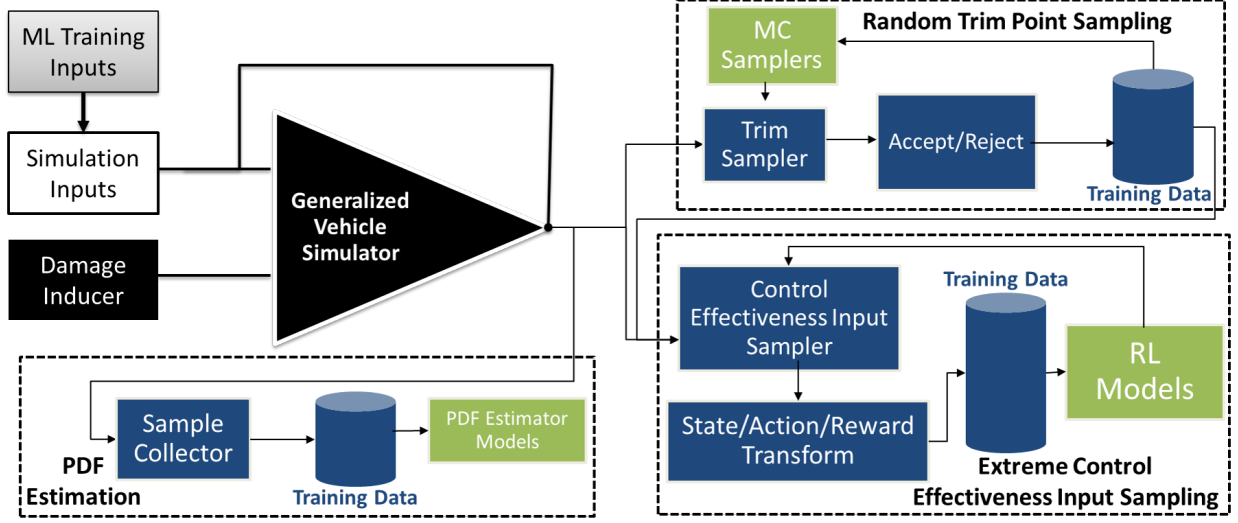


Fig. 6 Notional architecture of the data collection framework for flight envelope estimation.

Design of experiment (DOE) methods will be used to characterize the appropriate parameters and model architectures for each of the models in this architecture. After initial training and experimentation based on the DOE, a model performance assessment will be performed by measuring the training and prediction performance when varying the simulated damages to the aircraft, based on hazards that are characterized in the larger ICM program. In addition, once damages occur, we will measure transfer learning performance and accuracy after models have converged. These measurements will be produced by our experiments with the goal of answering the following research questions regarding the topic of flight envelope estimation:

- Are the non-linear estimators in this study reliable enough to be used for real-time, reliable flight envelope estimation?
- How does an increasing number of control effectors impact the speed and accuracy with which we can recompute and interpolate a trim set?
- Can a neural network approximation of the viscosity solution for the HJB-PDE constrain the search space when recomputing reachable sets?
- What are the minimum data that need to be monitored in-flight to accurately estimate a vehicle's actuation capabilities in off-nominal scenarios?

The answers to these questions will guide the packaging of our flight estimation tools and their use in future CONOPS.

VI. Conclusions

The paper presents a discussion on specific challenges associated with a robust contingency management system decision process and identifies select tools that would be beneficial in supporting it. Additionally, we provide specific development of tools for flight envelope estimation for an aircraft operating in an off-nominal state. The preliminary results of our flight envelope estimation tools show that Hamiltonian mechanics play a significant role. Our contributions to the flight envelope estimation framework for real-time operation require the following research explorations (experiments outlined in Section V.E):

- Online random trim point sampling using HMC methods
- Real-time approximation of control effectors using HNNs and RL
- Accurate estimation of forward and backward reachable set PDFs using HVAEs

The results of these flight envelope estimation research explorations, as well as development of additional autonomous tools are the subjects of future research.

Acknowledgments

This research was supported by the NASA Aeronautics Research Mission Directorate (ARMD), Transformative Tools and Technologies (TTT) project, under the Autonomous Systems / Intelligent Contingency Management subproject. Additional thanks to the NASA Langley Research Center MidRange Computer User's Group for supporting initial exploration and data analysis on the NASA Langley Research Center K-cluster.

References

- [1] Gregory, I. M., Acheson, M. J., Bacon, B. J., Britton, T. C., Campbell, N. H., Cook, J. W., Holbrook, J. B., Moerder, D. D., Murphy, P. C., Neogi, N. A., Simmons, B. M., McMinn, J. D., and Buning, P. G., "Intelligent contingency management for urban air mobility," *2021 AIAA SciTech Forum*, AIAA, 2021.
- [2] Bottasso, C. L., and Montinari, P., "Rotorcraft flight envelope protection by model predictive control," *Journal of the American Helicopter Society*, Vol. 60, No. 2, 2015, pp. 1–13.
- [3] Horn, J. F., Calise, A. J., and Prasad, J., "Flight Envelope Cueing on a Tilt-Rotor Aircraft Using Neural Network Limit Prediction," *Journal of the American Helicopter Society*, Vol. 46, No. 1, 2001, pp. 23–31.
- [4] Horn, J., Calise, A. J., and Prasad, J., "Flight envelope limit detection and avoidance for rotorcraft," *Journal of the American Helicopter Society*, Vol. 47, No. 4, 2002, pp. 253–262.
- [5] Yavrucuk, I., Prasad, J., and Unnikrishnan, S., "Envelope protection for autonomous unmanned aerial vehicles," *Journal of guidance, control, and dynamics*, Vol. 32, No. 1, 2009, pp. 248–261.
- [6] Unnikrishnan, S., Prasad, J., and Yavrucuk, I., "Flight evaluation of a reactionary envelope protection system for uavs," *Journal of the American Helicopter Society*, Vol. 56, No. 1, 2011, pp. 12009–12009.
- [7] Zhang, Y., De Visser, C., and Chu, Q., "Aircraft damage identification and classification for database-driven online flight-envelope prediction," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 2, 2018, pp. 449–460.
- [8] Norouzi, R., Kosari, A., and Sabour, M. H., "Real time estimation of impaired aircraft flight envelope using feedforward neural networks," *Aerospace Science and Technology*, Vol. 90, 2019, pp. 434–451.
- [9] Harno, H. G., and Kim, Y., "Safe flight envelope estimation for rotorcraft: a reachability approach," *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, IEEE, 2018, pp. 998–1002.
- [10] Yin, M., Chu, Q. P., Zhang, Y., Niestroy, M. A., and de Visser, C. C., "Probabilistic Flight Envelope Estimation with Application to Unstable Overactuated Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 12, 2019, pp. 2650–2663.
- [11] Greydanus, S., Dzamba, M., and Yosinski, J., "Hamiltonian neural networks," *Advances in Neural Information Processing Systems*, 2019, pp. 15379–15389.
- [12] Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D., "Hybrid Monte Carlo," *Physics letters B*, Vol. 195, No. 2, 1987, pp. 216–222.
- [13] Levy, D., Sohl-dickstein, J., and Hoffman, M., "Generalizing Hamiltonian Monte Carlo with Neural Networks," 2018.
- [14] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., "Equation of state calculations by fast computing machines," *The journal of chemical physics*, Vol. 21, No. 6, 1953, pp. 1087–1092.
- [15] Hastings, W. K., "Monte Carlo sampling methods using Markov chains and their applications," 1970.
- [16] Caterini, A. L., Doucet, A., and Sejdinovic, D., "Hamiltonian variational auto-encoder," *Advances in Neural Information Processing Systems*, 2018, pp. 8167–8177.
- [17] Choudhary, A., Lindner, J. F., Holliday, E. G., Miller, S. T., Sinha, S., and Ditto, W. L., "Physics-enhanced neural networks learn order and chaos," *Physical Review E*, Vol. 101, No. 6, 2020, p. 062207.
- [18] Mattheakis, M., Sondak, D., and Protopapas, P., "Physical Symmetries Embedded in Neural Networks," *Bulletin of the American Physical Society*, Vol. 64, 2019.
- [19] Bloem-Reddy, B., and Teh, Y. W., "Probabilistic symmetries and invariant neural networks," *Journal of Machine Learning Research*, Vol. 21, No. 90, 2020.

- [20] Finzi, M., Wang, K. A., and Wilson, A. G., “Simplifying Hamiltonian and Lagrangian Neural Networks via Explicit Constraints,” *Advances in Neural Information Processing Systems*, Vol. 33, 2020.
- [21] Kim, B., Chudomelka, B., Park, J., Kang, J., Hong, Y., and Kim, H. J., “Robust Neural Networks inspired by Strong Stability Preserving Runge-Kutta methods,” *European Conference on Computer Vision*, Springer, 2020, pp. 416–432.
- [22] Bertalan, T., Dietrich, F., Mezić, I., and Kevrekidis, I. G., “On learning Hamiltonian systems from data,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, Vol. 29, No. 12, 2019, p. 121107.
- [23] McDonald, K. T., “Hamiltonian with z as the Independent Variable,” Tech. rep., Technical Report, 2015.
- [24] Uchiyama, T., and Sonehara, N., “Solving inverse problems in nonlinear PDEs by recurrent neural networks,” *IEEE International Conference on Neural Networks*, IEEE, 1993, pp. 99–102.
- [25] Aarts, L. P., and Van Der Veer, P., “Neural network method for solving partial differential equations,” *Neural Processing Letters*, Vol. 14, No. 3, 2001, pp. 261–271.
- [26] Djeridane, B., and Lygeros, J., “Neural approximation of PDE solutions: An application to reachability computations,” *Proceedings of the 45th IEEE Conference on Decision and Control*, IEEE, 2006, pp. 3034–3039.
- [27] Niarchos, K., and Lygeros, J., “A neural approximation to continuous time reachability computations,” *Proceedings of the 45th IEEE Conference on Decision and Control*, IEEE, 2006, pp. 6313–6318.
- [28] Liu, D., Wang, D., Wang, F.-Y., Li, H., and Yang, X., “Neural-network-based online HJB solution for optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems,” *IEEE transactions on cybernetics*, Vol. 44, No. 12, 2014, pp. 2834–2847.
- [29] Raissi, M., Perdikaris, P., and Karniadakis, G. E., “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, Vol. 378, 2019, pp. 686–707.
- [30] Darbon, J., Langlois, G. P., and Meng, T., “Overcoming the curse of dimensionality for some Hamilton–Jacobi partial differential equations via neural network architectures,” *Research in the Mathematical Sciences*, Vol. 7, No. 3, 2020, pp. 1–50.
- [31] Zhu, Y., Zabararas, N., Koutsourelakis, P.-S., and Perdikaris, P., “Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data,” *Journal of Computational Physics*, Vol. 394, 2019, pp. 56–81.
- [32] Rutowski, E. S., “Energy approach to the general aircraft performance problem,” *Journal of the Aeronautical Sciences*, Vol. 21, No. 3, 1954, pp. 187–195.
- [33] Johnson, D. T., “Evaluation of energy maneuverability procedures in aircraft flight path optimization and performance estimation,” Tech. rep., AIR FORCE FLIGHT DYNAMICS LAB WRIGHT-PATTERSON AFB OH, 1972.
- [34] Akmeliawati, R., and Mareels, I. M., “Nonlinear energy-based control method for aircraft automatic landing systems,” *IEEE Transactions on Control Systems Technology*, Vol. 18, No. 4, 2009, pp. 871–884.
- [35] You, D.-I., and Shim, D. H., “Design of an aerial combat guidance law using virtual pursuit point concept,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 229, No. 5, 2015, pp. 792–813.
- [36] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M., “Deterministic policy gradient algorithms,” 2014.
- [37] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D., “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971*, 2015.
- [38] Marron, J., and Nolan, D., “Canonical kernels for density estimation,” *Statistics & probability letters*, Vol. 7, No. 3, 1988, pp. 195–199.
- [39] Heidenreich, N.-B., Schindler, A., and Sperlich, S., “Bandwidth selection for kernel density estimation: a review of fully automatic selectors,” *ASTA Advances in Statistical Analysis*, Vol. 97, No. 4, 2013, pp. 403–433.
- [40] Backurs, A., Indyk, P., and Wagner, T., “Space and Time Efficient Kernel Density Estimation in High Dimensions,” *Advances in Neural Information Processing Systems*, 2019, pp. 15799–15808.
- [41] Suga, N., Yano, K., Webber, J., Hou, Y., Higashimori, T., and Suzuki, Y., “Estimation of Probability Density Function Using Multi-bandwidth Kernel Density Estimation for Throughput,” *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, IEEE, 2020, pp. 171–176.

- [42] Theis, L., Oord, A. v. d., and Bethge, M., “A note on the evaluation of generative models,” *arXiv preprint arXiv:1511.01844*, 2015.
- [43] Takahashi, H., Iwata, T., Yamanaka, Y., Yamada, M., and Yagi, S., “Student-t Variational Autoencoder for Robust Density Estimation.” *IJCAI*, 2018, pp. 2696–2702.