Methodology for Path Planning with Dynamic Data-Driven Flight Capability Estimation

Methodology for Path Planning with Dynamic Data-Driven Flight Capability Estimation

Victor Singh* and Karen E. Willcox†

Massachusetts Institute of Technology, Cambridge, MA, 02139

This paper presents methodology to enable path planning for an unmanned aerial vehicle that uses dynamic data-driven flight capability estimation. The main contribution of the work is a general mathematical approach that leverages offline vehicle analysis and design data together with onboard sensor measurements to achieve dynamic path planning. The mathematical framework, expressed as a Constrained Partially Observable Markov Decision Process, accounts for vehicle capability constraints and is robust to modeling error and disturbances in both the vehicle process and measurement models. Vehicle capability constraints are incorporated using Probabilistic Support Vector Machine surrogates of high-fidelity physics-based models that adequately capture the richness of the vehicle dynamics. Sensor measurements are treated in a general manner and can include combinations of multiple modalities such as GPS/IMU data as well as structural strain data of the airframe. Results are presented for a simulated 3-D environment and point-mass airplane model. The vehicle can dynamically adjust its trajectory according to the observations it receives about its current state of health, thereby retaining a high probability of survival and mission success.

Nomenclature

C, c	Constraint and marginalized constraint function	V_π	Value function for control policy π
D_{KL}	Kullback-Leibler Divergence	W	Transition model noise term covariance matrix
N_r	Number of response surface basis functions	Δt	Time step
O	Observation probability distribution	α, ϕ	Angle of attack and bank angle
P	Engine power	α_1, α_2	Interpolation function tuning parameters
Q	Observation model noise term covariance matrix	$\alpha_L^{(n)}, \alpha_D^{(n)}$	n^{th} response surface basis function coefficient for lift and drag, respectively
R, r	Reward and marginalized reward function	$\alpha_{z_s}^{(n)}$	n^{th} response surface basis function coefficient for measurement quantities z_s
S_v	Support vector machine discriminant	β_1, β_2	Probabilistic support vector machine tuning parameters
T, D, L	Thrust, drag, and lift force		
T_{avail}	Available engine thrust	$\epsilon_{\mathrm{allow}}$	Allowable strain levels in structure
T_r	Transition probability distribution	ϵ_{peak}	Peak strain levels in structure
V, ψ, γ	Velocity, heading, and flight path angle	η	Appropriate normalizing term
		λ	Discount factor
V^*	Optimal value function	$\mathbb{E}$	Expectation operator

*Graduate student, Department of Aeronautics and Astronautics, victorsi@mit.edu, Student Member AIAA

†Professor of Aeronautics and Astronautics, kwillcox@mit.edu, Associate Fellow AIAA

$\mathcal{B}_G$	Discrete belief space	l_c	Airframe structure affected by damage
$\mathcal{C}$	Capability set		
$\mathcal{D}_l$	Damage library	m, g	Mass and gravitational acceleration
$\mathcal{M}_{b'}$	M largest contributing $b \in \mathcal{B}_G$ in $\varphi(b', b)$	n_v, n_s	Number of quantities in z_v and z_s, respectively
$\mathcal{V}$	Capability volume	$p(\cdot)$	Probability measure
ν	Observation model noise term	p_{thresh}	Constraint threshold
π	Control policy	$s \in \mathcal{S}$	Total vehicle state and space
ψ_n	n^{th} response surface basis function	t_{depth}	Damage depth into the skin
τ	Bayes posterior	t_{skin}	Skin thickness of wing
φ	Interpolation function in Bellman equation	$u \in \mathcal{U}$	Control input and space
		w	Transition model noise term
$b \in \mathcal{B}$	Belief state and space	$x \in \mathcal{X}$	Kinematic state vector and space
c_s	Damage severity parameter	x_g, y_g, z_g	Global x,y,z-coordinates
$d \in \mathcal{D}$	Damage state and space	$z \in \mathcal{Z}$	Measurement vector and space
f	Transition model	$z_s \in \mathcal{Z}^s$	Component of measurement vector relating to vehicle structural state and associated space
$f_{\mathcal{X}}$	Kinematic state transition model		
h	Observation model		
k_{loss}	Material moduli loss of damaged elements	$z_v \in \mathcal{Z}^v$	Component of measurement vector relating to vehicle kinematic state and associated space
$l_{\text{chord}}, w_{\text{chord}}$	Chordwise location and width of damage		
		Subscripts	
$l_{\text{span}}, w_{\text{span}}$	Spanwise location and width of damage	trim	Designation of vehicle trim condition variables
l_{wing}, c_{wing}	Length and chord of the wing	k	Quantity at the k^{th} time step

I. Introduction

A wealth of *offline* information is generated during the design and analysis of an unmanned aerospace vehicle. This information spans multiple modeling sources of varying fidelity. These sources can include global aerostructural analysis involving finite element codes modeling different vehicle scenarios and damaged configurations, manufacturing design criteria, as well as detailed component analysis of the airframe structure. However, such information is typically not used to inform decisions during online operation of the vehicle. Recent advances in aerospace sensing technologies could potentially allow for *online* monitoring of internal structural strains and sensing of aerodynamic shears, temperatures, and wind speeds across the entire wing surface of an aerospace vehicle.[1] There exists a significant opportunity to leverage a combination of these offline and online information sources, together with machine learning and big data techniques, to achieve gains in vehicle operation.

This paper focuses particularly on the setting of onboard measurements coupled with offline vehicle analysis information for the next generation of self-aware unmanned aerospace vehicles (see Figure 1) that can dynamically adapt mission strategy based on the observations they receive. Our data-driven planning strategy applies the ideas of Dynamic Data-Driven Application Systems (DDDAS), where observational data are used in a feedback loop with computation and simulation. Our approach has the potential to improve vehicle reliability and survivability, and therefore fiscal risk, by allowing the vehicle to make use of knowledge of its own design in conjunction with online sensors to estimate its current health beyond its baseline performance and react accordingly. In this way, a vehicle mimics the behavior of a biological organism, using and updating knowledge about itself with information it has collected over time to act promptly in favorable conditions and conservatively otherwise.[2]

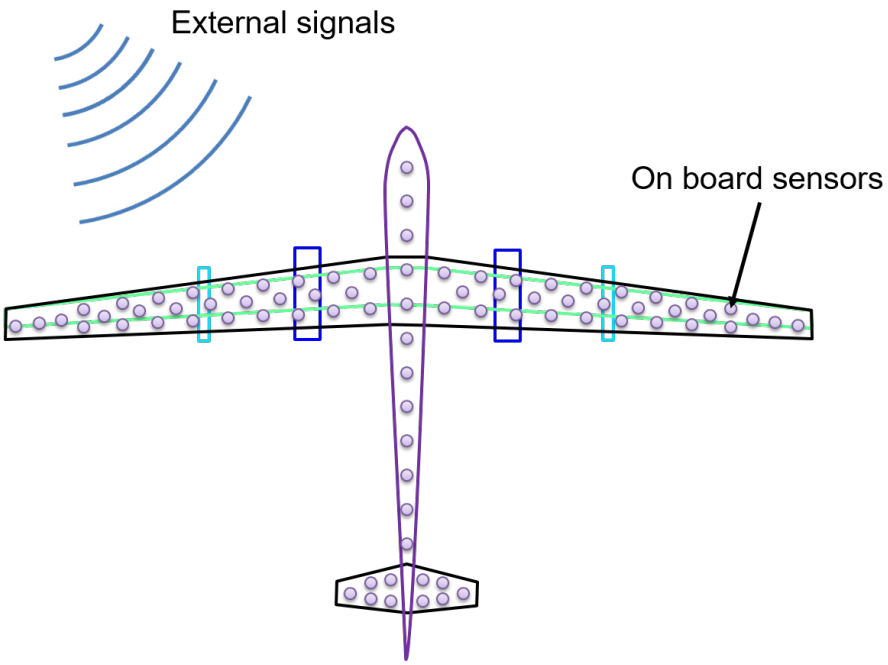

Figure 1. Envisioned self-aware aerospace vehicle with sensors embedded across its airframe.

Enabling a self-aware aerospace vehicle requires four key components:

1. Sensory output about the vehicle's internal and external conditions as well as its position and orientation in its environment.

2. Conversion of sensory output of internal conditions to characterization of the vehicle's current health.

3. Capability estimation that provides quantifiable metrics of the vehicle's abilities based on its current health.

4. A path planner that can respond accordingly to changes in the vehicle's estimated capability.

In the paragraphs that follow, we briefly focus attention on items 2-4.

Conversion of sensory output to characterization of vehicle health can be associated with damage detection, identification, and monitoring of the vehicle airframe structure. These are common tasks in the structural health monitoring and operational loads monitoring community (see e.g., Refs. 3–6 and 7).

In the area of capability estimation, one approach is to construct libraries characterizing the flight envelope for a range of different damage events. The library can be represented and queried using techniques such as least squares, filtering methods, or statistical inference for parameter estimation and determination of the modified flight envelope.[8] A data-driven approach leverages offline information about the design of a vehicle, such as aerodynamic and structural limits, together with online sensor information to provide a dynamic estimate of the vehicle capability.[9] A large body of literature addresses changes in aircraft performance parameters such as stability/control derivatives both with simulation and wind-tunnel experiments.[10] Parameters such as lift, drag, stall, center-of-gravity, and inertia shifts are explored where wing structure damage is modeled as a loss of stiffness in the underlying vortex lattice and lumped element models used to represent the aircraft structure and aerodynamics.[11,12]

Work in path planning and control for damaged aircraft has commonly focused on settings where the damage is known a priori. A common damage scenario includes an engine-out emergency where one engine is failed or has reduced thrust and the planner must find a safe strategy to land.[13,14] Other works investigate the effects of softened panels or removal of entire airframe sections as in damage-tolerant control.[15] A strong

focus is placed on stability and tracking of the developed control laws which derive from adaptive control schemes, model reference control, model predictive control, and neural networks.[16–18] Path planners generally account for degraded capability (structural/actuator, sensor, communication, and fuel) as an increased risk, usually in a probabilistic sense, to find safe trajectories or to change mission strategies.[19–21] In these works, the emphasis is on the planning strategy and not so much on damage identification or capability estimation. One work considers the process of flight envelope evaluation and path planning using trim motion primitives in order to find safe landing trajectories.[22]

A fusion of capability estimation and path planning with large-scale sensory information from multiple modalities remains sparse in the literature and is the focus of this paper. Our goal is to develop a data-driven methodology that takes advantage of the wealth of offline vehicle design and analysis information coupled with online sensor data in planning and executing missions with a high success rate. We propose a methodology that is flexible with respect to the physics-based models describing the aerostructural and dynamical characteristics of the vehicle and measurement processes. The methodology incorporates vehicle damage, uncertainty due to modeling and noise, and multiple sensor modalities. We demonstrate our methodology on a 3-D path planning example, where an unmanned aerial vehicle (UAV) is instructed to reach a target location while avoiding obstacles and not exceeding available capability as a result of damage.

An illustrative scenario of interest is depicted in Figure 2. Here a UAV is initially tasked to navigate through a series of obstacles to the target location. However, it becomes damaged and must quickly learn of its reduced maneuverability as a result of damage and re-plan its trajectory if necessary. Multiple routes are possible that require different degrees of vehicle capability in order to perform successfully. If the vehicle recognizes itself to be mildly damaged then the vehicle will take shorter and more aggressive routes, whereas if the vehicle recognizes itself to be severely damaged it will take longer and more conservative routes. The vehicle must ultimately decide which route to take after determining its health. This determination is based on collected measurements as well as offline information embodied in scenario libraries.

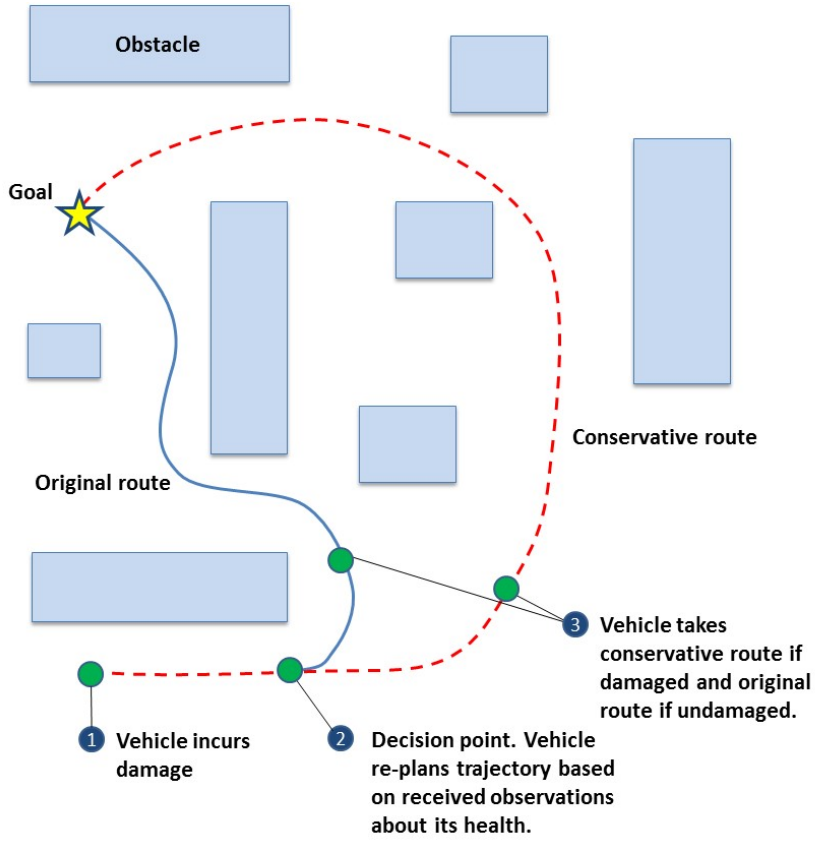

Figure 2. Illustration of scenario of interest.

The remainder of the paper is organized as follows. Section II presents the data-driven methodology. Section III gives demonstration and implementation of the proposed approach on an example scenario. Section IV provides results and discussion of the example scenario. Finally, Section V provides concluding remarks.

II. Methodology

This section describes the mathematical aspects of our methodology. We begin by defining the notion of a self-aware aerospace vehicle and an outline of the computational roadmap in Subsection II.A. We next define the concepts of vehicle state, control, and measurements used throughout the paper in Subsection II.B. Subsection II.C describes the models for state transitions and measurements while Subsection II.D describes state estimation. Vehicle damage and capability representation is described Subsection II.E. Construction of the damage library used in the approach is discussed in Subsection II.F. Finally, the path planning formulation is described in Subsection II.G.

II.A. Path Planning for a Self-Aware Aerospace Vehicle

A self-aware aerospace vehicle is one that can collect information about its internal conditions and its surroundings through measurements from onboard sensors, and then use this information in real-time dynamic decision-making. In this work, the internal conditions considered are related to damage incurred to the vehicle such that its flight envelope is reduced. The path-planning solution is then one that determines the modified flight envelope through an inference problem and uses this knowledge to re-plan the trajectory in flight if necessary.

The proposed approach divides computational effort between offline and online phases, using the general approach proposed in Ref. 9. In the offline phase, we compute probabilistic damage libraries characterizing capability using high-fidelity physics-based models as well as determine allowable control actions for different vehicle state distributions. In the online phase, we infer change in capability and the modified flight envelope due to damage, using noisy sensor data and the precomputed library database in a Bayesian inference problem. The updated capability estimate is then used to re-plan the vehicle's trajectory if necessary. Figure 3 shows the path-planning process for the offline/online approach.

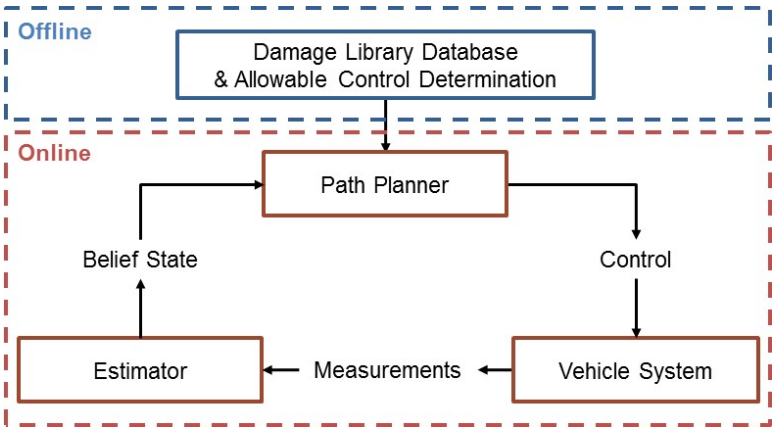

Figure 3. Offline/online approach to proposed dynamic path planner.

II.B. Vehicle State, Control, and Measurement Definitions

The vehicle state represents both the vehicle kinematics and the vehicle's current damage state. To that end, we define the vehicle state by two components. The first component describes the kinematic quantities of the the vehicle (global position, velocity, heading, flight path angle, etc.) and is given by a continuous vector called the kinematic state $x \in \mathcal{X} \subseteq \mathbb{R}^n$, where $\mathcal{X}$ denotes the kinematic state space. The second component represents a particular damage state (parametrized by depth, location, size, etc.) and is given by $d \in \mathcal{D}$, where $\mathcal{D}$ denotes the damage state space. The total vehicle state is given by the vector $s = [x, d]^T \in \mathcal{S} \subseteq \mathcal{X} \times \mathcal{D}$, where $\mathcal{S}$ denotes the total state space.

The vehicle is controlled through an input vector $u \in \mathcal{U} \subseteq \mathbb{R}^m$, where $\mathcal{U}$ denotes the space of control inputs. These inputs can include deflections of the ailerons, elevators, rudder, or other control surface components or signals.

The vehicle receives measurements in the form of a vector $z = [z_v, z_s]^T \in \mathcal{Z}$ where $z_v \in \mathcal{Z}^v \subseteq \mathbb{R}^{n_v}$ contains n_v quantities relating to the vehicle kinematic state and $z_s \in \mathcal{Z}^s \subseteq \mathbb{R}^{n_s}$ contains n_s quantities relating to the vehicle structural state. The space of all measurements is denoted by the set $\mathcal{Z} \subseteq \mathcal{Z}^v \times \mathcal{Z}^s$, where $\mathcal{Z}^v$ is the space associated to z_v and $\mathcal{Z}^s$ is the space associated to z_s. Examples of z_v are Global Positioning System readings, accelerometer readings, velocity, heading and other quantities related to the kinematic state. Examples of z_s are readings from sensor strain gauges located throughout the airframe structure.

II.C. Transition and Observation Models

The vehicle system ("Vehicle System" block in Figure 3) is described using a transition model and an observation model. The transition model, which governs the evolution of the vehicle state, is given by a model of the form:

$$s_{k+1} = f(s_k, u_k, w_k), \tag{1}$$

where $f : \mathcal{S} \times \mathcal{U} \times \mathbb{R}^p \to \mathcal{S}$, $w \in \mathbb{R}^p$ is the transition model noise term accounting for model uncertainty and disturbances, and the subscript $k \in \mathbb{N}$ denotes the value of a quantity at the k^{th} time step. The transition

model encodes how the state of the vehicle transitions from one state to another given a particular control input. The kinematic state will change through the vehicle dynamics while the damage state will change as result of progressive damage induced by aggressive vehicle dynamics or by external events.

The observation model, which relates the vehicle state and control to a measurement, is given by the following model:

$$z_k = h(s_k, u_{k-1}, \nu_k), \tag{2}$$

where $h : \mathcal{S} \times \mathcal{U} \times \mathbb{R}^r \to \mathcal{Z}$ and $\nu \in \mathbb{R}^r$ is the observation model noise term accounting for model uncertainty and disturbances. Measurements considered here include combinations of different modalities ranging from output about the vehicle kinematics to that of the internal structural state. This observation model allows flexibility to what quantity can be considered a measurement. A measurement can come from output of an arbitrary combination or arrangement of sensors but more importantly can come from quantities that may be the result of a post-process (e.g., Fourier analysis, pattern recognition, statistical inference techniques, etc.) provided a model is available that relates the vehicle state and control back to those quantities. The latter becomes important where strain-only information is insufficient to detect and characterize damaged structure, in which case some form of spectral analysis of structural response is required.

II.D. State Estimation and Hidden Markov Model Assumption

Due to partial observability, the true state of the vehicle is hidden and unavailable. Instead, we track a quantity known as the belief state.[23] Formally, the belief state is the probability distribution of the state conditioned on the history of measurements and control:

$$b_k(s) = p(s|z_{0:k}, u_{0:k-1}), \tag{3}$$

where $b_k \in \mathcal{B}$ is the belief state at time step k and lives in the belief space $\mathcal{B}$, p is a probability distribution, and the subscript $0 : k$ denotes quantities for each time step from 0 up to k. The belief state is the output of the "Estimator" block in Figure 3. To simplify estimation, we assume the total vehicle state evolves according to a hidden Markov model (see Figure 4). Under this model, the belief state updates according to the definition of the Bayes filter:[23]

$$b_k(s) = \tau(b_{k-1}, u_{k-1}, z_k)(s) = \eta_k p(z_k|s, u_{k-1}) \int_{s' \in \mathcal{S}} p(s|s', u_{k-1}) b_{k-1}(s') \, \mathrm{d}s', \tag{4}$$

where $\eta_k^{-1} = p(z_k|b_{k-1}, u_{k-1})$ is the normalizing term and $\tau : \mathcal{B} \times \mathcal{U} \times \mathcal{Z} \to \mathcal{B}$, the Bayes posterior, can be thought of as the transition model for the belief state. The control action is a function of the data, namely the history of measurements and control. However, since the belief state is a sufficient statistic,[24] the control can be expressed as a function of the current belief state.

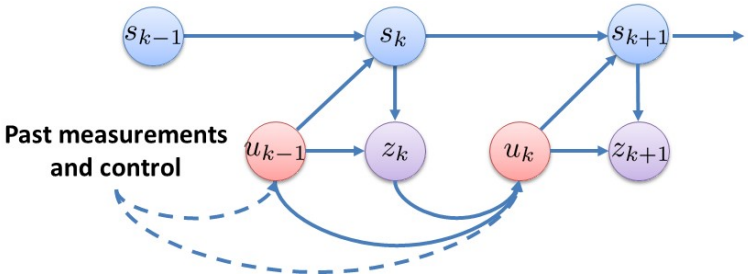

Figure 4. Hidden Markov model for state evolution. The vehicle states (blue) are hidden and so the control action is a function of past measurements and control. It can be expressed as a function of the current belief state.

II.E. Damage and Capability Representation

We treat damage as a reduction in capability where capability is defined as a set $\mathcal{C} \subseteq \mathcal{X} \times \mathcal{U}$ such that the pair $[x, u] \in \mathcal{C}$ is feasible, where $[x, u]$ corresponds to a particular maneuver. Example maneuvers include pull-up, pull-down, banked turn, or any other combination of vehicle state and control. The set $\mathcal{C}$ is the set of maneuvers that will not lead to vehicle failure. The boundary of $\mathcal{C}$ is defined by the physics of the vehicle and will include structural, engine, or aerodynamic limits such as stall, maximum velocity, and/or maximum structural component strengths.

To quantify capability at the k^{th} time step, we assign a probability of whether a maneuver is feasible given a particular damage state configuration by the following probability distribution:

$$p([x, u] \in \mathcal{C} | d_k). \tag{5}$$

However, since the true damage state of the vehicle is unknown, we instead consider the probability of whether a maneuver is feasible given the history of measurements and control:

$$p([x, u] \in \mathcal{C} | z_{0:k}, u_{0:k-1}). \tag{6}$$

Conditioning this expression on the damage state d_k and applying the Markov model assumption, this expression can be rewritten as

$$p([x, u] \in \mathcal{C} | z_{0:k}, u_{0:k-1}) = \mathbb{E}\big[p([x, u] \in \mathcal{C} | d_k)\big], \tag{7}$$

which is a weighted average of Eq. 5 with respect to the marginal posterior distribution of d_k. Note that this expectation changes in time based on the evolution of the marginal posterior distribution $p(d_k | z_{0:k}, u_{0:k-1})$ and hence encodes the notion of "dynamic capability" through a weighted average of this marginal posterior. Another interpretation of the marginal posterior is that of a model selection problem, where the marginal posterior is the probability that the vehicle damage state is described by the model given by d_k conditioned on the history of data. These probabilities or weights change as new information is made available, reflecting the growing confidence of one model over another.

II.F. Damage Library Construction and Surrogate Modeling

The offline phase of the methodology shown in Figure 3 involves computation using high-fidelity physics-based models of a damage library database to be used for inference online. This amounts to computing $p([x, u] \in \mathcal{C} | d)$ and the observation model h for each damage state d of interest. Generation of vehicle capability for a given damage state d is described as follows (the reader is referred to Ref. 9 for more details):

1. Run high-fidelity physics-based models at points that adequately sample the boundary between infeasible regions and feasible regions in the combined vehicle state-control space $\mathcal{X} \times \mathcal{U}$ for a particular damage state d. An adaptive sampling algorithm can be used to pick points intelligently.[25] For each feasible maneuver point $[x, u]$, assign the value $+1$ and for each infeasible maneuver point, assign the value -1. In addition, for the measurement quantities of interest, record their values for each maneuver point $[x, u]$. These recorded values will serve as the training data to build the observation model using standard machine learning techniques such as response surface modeling or kriging.

2. Fit the feasible/infeasible points with a Support Vector Machine (SVM) or some other binary classifier.

3. Use the SVM discriminant in a Probabilistic Support Vector Machine (PSVM). The PSVM defines the capability for a given damage state d as:

$$p([x, u] \in \mathcal{C} | d) = \frac{1}{1 + e^{\beta_1(d) S_v(s,u) + \beta_2(d)}}, \tag{8}$$

where $S_v : \mathcal{S} \times \mathcal{U} \to \mathbb{R}$ is the SVM discriminant and $\beta_1, \beta_2 : \mathcal{D} \to \mathbb{R}$ are tuning parameters.

The use of a PSVM provides an inexpensive and data-source-flexible surrogate that captures the richness of the vehicle constraints. Moreover, it allows for direct application of dynamic capability in the Bayesian context as specified in Eq. 7. Figure 5 shows an example capability PSVM generated over the combined vehicle-control space for one particular damage state using the above procedure. For this case, the combined vehicle kinematic state and control space is parametrized in terms of velocity, bank angle, and angle of attack. Here we see that with either a high velocity, bank angle, and/or angle of attack, there is low probability given by the PSVM of being able to perform the corresponding maneuver due to reaching of aerodynamic stall or exceeding structural limitations. Figure 6 illustrates how the PSVM contours change as a result of increased damage to the vehicle. Here, PSVM contours for 5 example damage states are shown ranked from lowest severity (pristine) to highest (worst case damage). The parametrization of the combined vehicle kinematic state and control is again velocity and bank angle but the angle of attack is fixed. The immediate observation is that the region underneath the SVM discriminant equal to 0 shrinks as damage is worsened. This motivates the following definition of a damage severity parameter to characterize the extent of damage. We define the damage severity parameter $c_s(d) \in [0, 1]$ by the following relation:

$$c_s(d) = \frac{\mathcal{V}_{\text{pristine}} - \mathcal{V}(d)}{\mathcal{V}_{\text{pristine}}}, \tag{9}$$

where $\mathcal{V} \in \mathbb{R}_+$ represents the volume underneath the associated capability boundary (SVM discriminant equal to 0) for damage state d and the pristine structural state denoted by the subscript "pristine". We use the damage severity parameter to rank and downselect damage states for the library made available to the path planner. This library, which we denote by $\mathcal{D}_l \subset \mathcal{D}$, serves as the limited set of damage states over which we perform inference and take expectations. Note that this set is incomplete in that it does not contain every conceivable damage state. However, the assumption is that the library contains enough damage states to adequately account for the extent of damage in which the vehicle finds itself. New conditions that emerge as a result of recurrent inspection and maintenance of the vehicle can be easily added to the library over time, reflecting a growing and maturing knowledge database for that vehicle over its lifecycle.

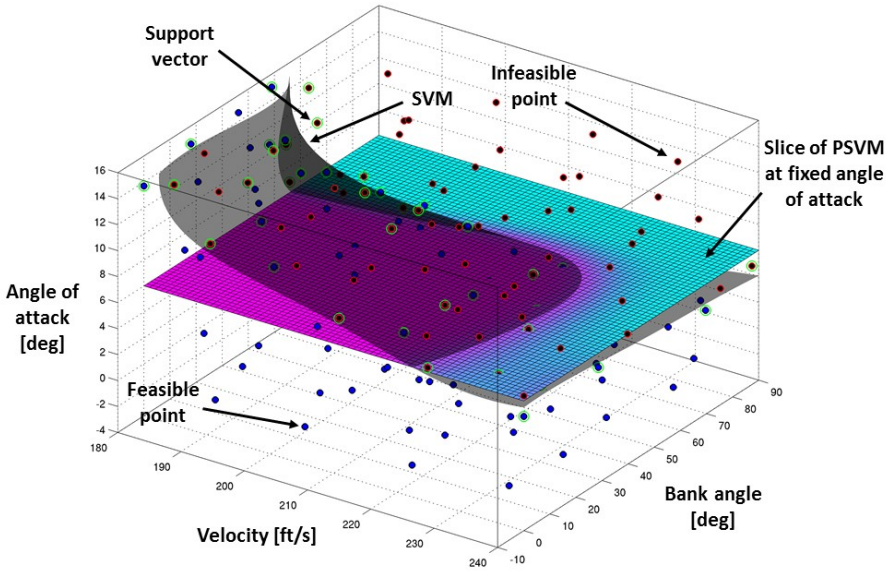

Figure 5. Capability boundary and PSVM contour slice for a given vehicle damage state. Points underneath the boundary lie in the feasible maneuver region while points above lie in the infeasible region of the combined vehicle state-control space.

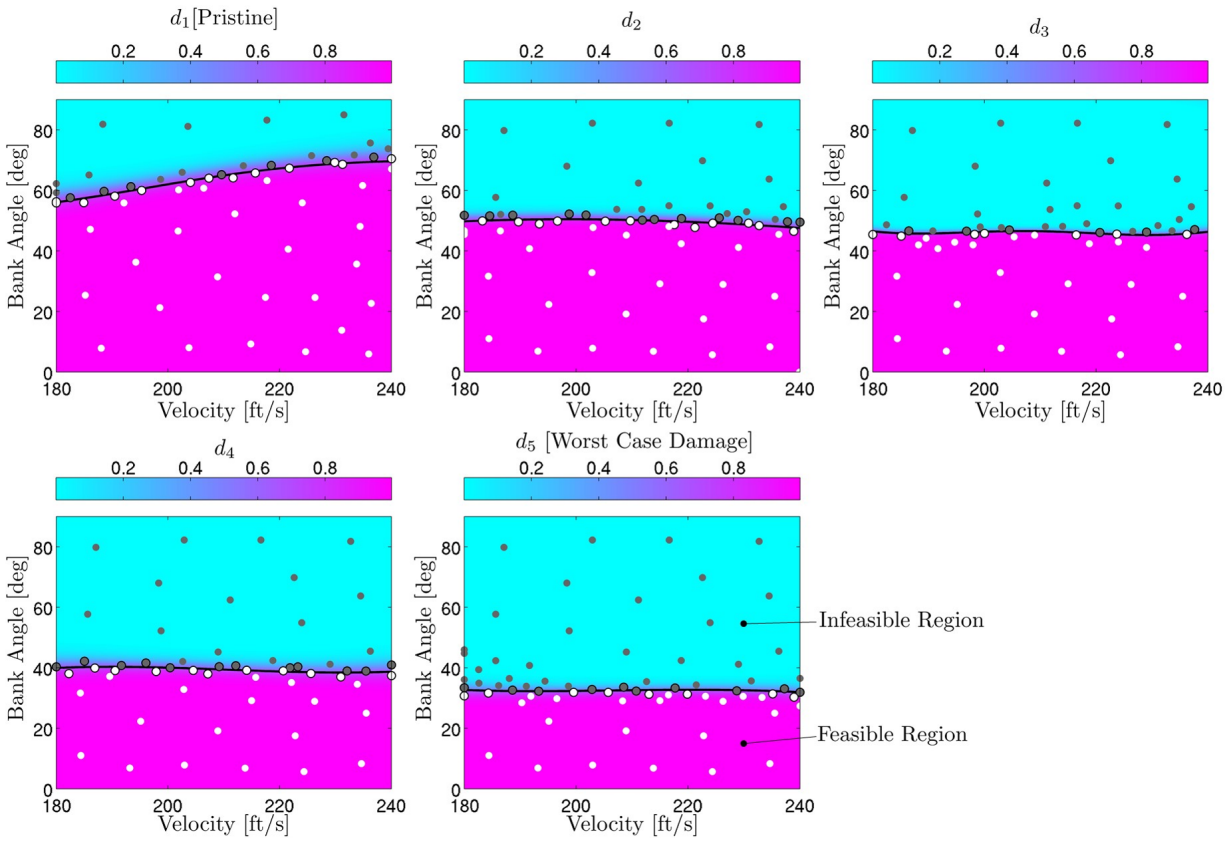

Figure 6. Comparison of PSVM capability contours at a fixed angle of attack for different damage states. Damage states are ordered based on increasing c_s from left to right and top to bottom.

II.G. Path Planning

We formulate the path planning problem as a Constrained Partially Observable Markov Decision Process (CPOMDP).[23,26] Formally, a CPOMDP is a tuple $\langle \mathcal{S}, \mathcal{U}, \mathcal{Z}, T_r, O, R, C \rangle$ where:

- $\mathcal{S}, \mathcal{U}, \mathcal{Z}$ is the vehicle state space, control input space, and measurement space as defined in Subsection II.B.
- $T_r(s, u, s') : \mathcal{S} \times \mathcal{U} \times \mathcal{S} \to \mathbb{R}_+$ is the transition probability distribution associated with the vehicle transition model and gives the probability of transitioning to state s' from the initial state s after applying control input u. It is defined as $T_r(s, u, s') \equiv p(s'|s, u)$ and describes the statistics of the transition model f.
- $O(s, u, z) : \mathcal{Z} \times \mathcal{U} \times \mathcal{S} \to \mathbb{R}_+$ is the observation probability distribution associated with the observation model and gives the probability of observing z after applying control input u when the vehicle is in state s. It is defined as $O(s, u, z) \equiv p(z|s, u)$ and describes the statistics of the observation model h.
- $R(s, u, s') : \mathcal{S} \times \mathcal{U} \times \mathcal{S} \to \mathbb{R}$ is the one-step reward function for the vehicle being in state s and applying control action u, as well as the reward for the transitioned state s'. The reward function encodes penalties for obstacles in the environment, rewards for reaching the goal location, and penalties/rewards for other states.
- $C(s, u, s') : \mathcal{S} \times \mathcal{U} \times \mathcal{S} \to \mathbb{R}$ is the constraint function. For our purposes, $C(s, u, s') = p([x', u] \in \mathcal{C}|d')$. Note that the constraint function here considers the control action u and the transitioned state s'.

The objective of a CPOMDP is to find a feedback control policy $\pi : \mathcal{B} \to \mathcal{U}$ that maximizes the total expected discounted reward subject to the requirement that the expectation of the constraint function be above some threshold. That is:

$$\begin{aligned} \text{maximize } V_\pi(b) =& \mathbb{E}\left[\sum_{k=0}^{\infty} \lambda^k r(s_k, u_k) | \pi, b_0 = b\right] \\ s.t. \quad & \mathbb{E}[c(s_k, u_k)] \geq p_{\text{thresh}} \quad \forall k, \end{aligned} \tag{10}$$

where

$$\begin{aligned} r(s,u) &= \int_{s' \in \mathcal{S}} R(s,u,s') p(s'|s,u) \text{ ds}' \\ c(s,u) &= \int_{s' \in \mathcal{S}} C(s,u,s') p(s'|s,u) \text{ ds}'. \end{aligned} \tag{11}$$

The threshold $p_{\text{thresh}} \in [0,1]$ tunes how aggressive or cautious is the resulting control policy and how much on average the constraints are satisfied. A high value ($p_{\text{thresh}} \to 1$) corresponds to a conservative policy, a medium value ($p_{\text{thresh}} \to 0.5$) corresponds to a more aggressive policy, while a low value ($p_{\text{thresh}} \to 0$) corresponds to an aggressive policy that will lead to a high likelihood of vehicle failure. The expectation of the discounted reward is conditioned on the policy π and the initial belief state b_0 being equal to b.

Solving Eq. 10 exactly is intractable and approximate solution techniques remain an active area of research. However, there exist numerous techniques for stochastic path planning or planning in belief space that one can employ to solve the above optimization statement. These techniques include dynamic programming, sampling-based algorithms,[27,28] and search-based algorithms[29–32] .

III. Example Problem

This section applies our approach to an example scenario where a damaged aircraft must navigate its way through a series of obstacles to a target location while using dynamic capability estimation. Section III.A describes the problem setup and the parametrization of damage. Section III.B describes the damage states assessed. Section III.C details the physics-based models as well as process and measurement models used. Section III.D discusses the solution to the path planning formulation.

III.A. Problem Setup

The problem scenario involves a UAV that is tasked to reach a goal location in minimum time while avoiding obstacles, as depicted in Figure 7. However, the UAV encounters damage and must quickly react and replan its trajectory. The path planner is tested for a range of different damage states, some that are in the library and some that are not. The vehicle starts at a lower elevation than the target location and must simultaneously climb and maneuver around obstacles as it makes its way to the target location.

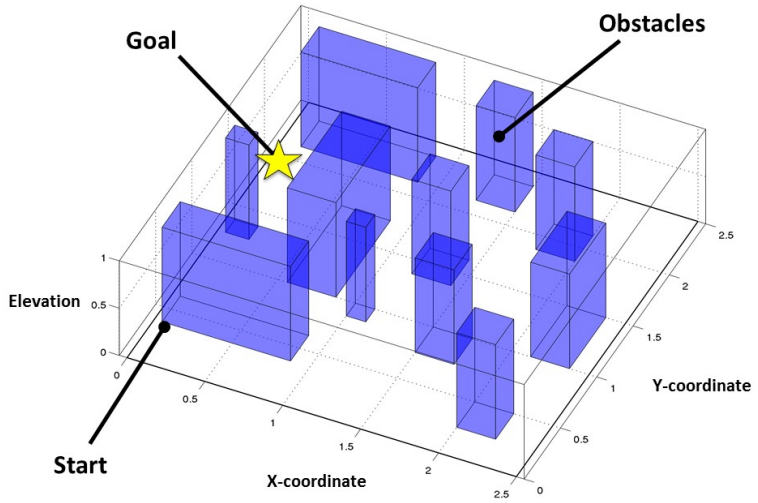

Figure 7. 3-D environment for example problem.

We parametrize a damage state d by the tuple $\langle l_c, l_{\text{span}}, w_{\text{span}}, l_{\text{chord}}, w_{\text{chord}}, t_{\text{depth}}, k_{\text{center}}, k_{\text{edge}} \rangle$. Here, l_c is a categorical variable that indicates the affected airframe structure (wing, horizontal stabilizer, etc.), l_{span} is the spanwise location of damage, w_{span} is the spanwise width of damage, l_{chord} is the chordwise location of damage, w_{chord} is the chordwise width of damage, t_{depth} is the damage depth into the skin, and k_{loss} is the material moduli (elastic and shear) loss of affected elements. An example parametrization for a damage state d is given by

$$\langle \text{Upper Right Wing Surface}, 0.1 l_{\text{wing}}, 0.05 l_{\text{wing}}, 0.3 c_{\text{wing}}, 0.1 c_{\text{wing}}, 0.7 t_{\text{skin}}, 0.99 \rangle,$$

which corresponds to damage of the upper right wing surface at spanwise location 0.1 times the length of the wing, chordwise location 0.3 times the chord of the wing, spanwise width of 0.05 times the length of the wing, chordwise width of 0.1 times the chord of the wing, depth of 0.7 times the skin thickness of the wing, and 99% material moduli loss of affected elements. Other damage states are defined similarly.

III.B. Damage State Selection and Methodology Assessment

We test our methodology by providing the path planner a limited damage library (as mentioned in Subsection II.F) and running the path planner on the vehicle for different damage states. Some of these damage states will be in the library while others will not. Our objective is to characterize how the path planner performs when 1) the vehicle has undergone damage that is in the library and 2) the vehicle has undergone damage that is not contained in the library.

To begin, we restrict ourselves to damage of the upper right wing surface of the vehicle, as shown in Figure 8. We then generate 73 damage states by performing a full factorial exploration of the parameters listed in Table 1. Ten of these states populate the damage library. These 10 states are used to build the PSVM surrogates, observation models, and also to do the dynamic inference. The 10 states for the library are determined by ranking all 73 damage states by increasing values of c_s and selecting states at uniform increments from lowest severity (pristine) to highest (worst case damage), as shown in Figure 9. Damage parametrization of the 10 states in the library are summarized in Table 2.

Table 1. Damage state parametrization. A full factorial exploration is performed with the indicated parameters.

Parameter	Value
l_c	Upper Right Wing Surface
$l_{\text{span}}/l_{\text{wing}}$	[0.1 0.2 0.3 0.4]
$w_{\text{span}}/l_{\text{wing}}$	[0.05]
$l_{\text{chord}}/c_{\text{wing}}$	[0.3 0.5 0.7]
$w_{\text{chord}}/c_{\text{wing}}$	[0.1 0.3 0.5]
$t_{\text{depth}}/t_{\text{skin}}$	[0.7 0.9]
k_{loss}	0.99

Table 2. Parametrization for the 10 members in the damage library. URWS denotes Upper Right Wing Surface.

Parameter	1	2	3	4	5	6	7	8	9	10
l_c	URWS	URWS	URWS	URWS	URWS	URWS	URWS	URWS	URWS	URWS
$l_{\text{span}}/l_{\text{wing}}$	0	0.30	0.30	0.40	0.30	0.30	0.10	0.40	0.20	0.10
$w_{\text{span}}/l_{\text{wing}}$	0	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
$l_{\text{chord}}/c_{\text{wing}}$	0	0.70	0.50	0.50	0.50	0.30	0.30	0.30	0.70	0.70
$w_{\text{chord}}/c_{\text{wing}}$	0	0.30	0.10	0.50	0.30	0.30	0.10	0.30	0.10	0.50
$t_{\text{depth}}/t_{\text{skin}}$	0	0.90	0.70	0.90	0.90	0.70	0.70	0.90	0.70	0.70
k_{loss}	0	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

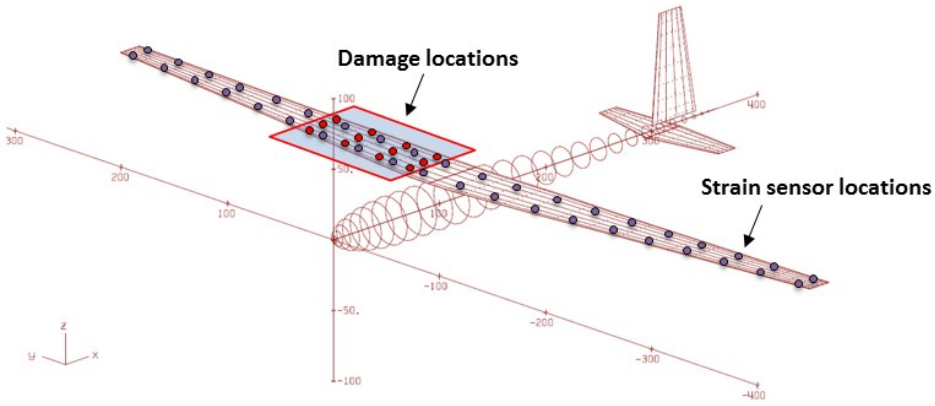

Figure 8. Location of damage to the UAV and strain sensor placement.

Next, for each of the 73 damage states, we apply the corresponding damage to the vehicle and perform the mission of moving the vehicle from the start location to the goal using the path planner with the aforementioned damage library. We repeat this mission 50 times for each damage state for a total of $N = 73 \times 50 = 3650$ missions. For each mission i, we store the following information: mission success (S_i), mission time (termination time) (t_i), and peak strain levels across the airframe structure ($\epsilon_{\text{peak},i}$) during the mission. Mission success $S_i = 1$ if the vehicle reaches the target location without collisions and structural failure, and $S_i = 0$ otherwise. Mission time is the time to reach the target objective or the time of termination as a result of structural failure or collision with an obstacle. Structural failure occurs when the peak strain level ϵ_{peak} across the airframe structure exceeds allowable levels ϵ_{allow} at any time during the mission. Since we are interested in survivability of the vehicle, we quantify survivability as the number of successful missions

divided by the total number of missions:

$$P(S) = \frac{1}{N}\sum_{i=1}^{N} S_i. \tag{12}$$

Thus, the goal of the path planner is to maximize $P(S)$ across all missions.

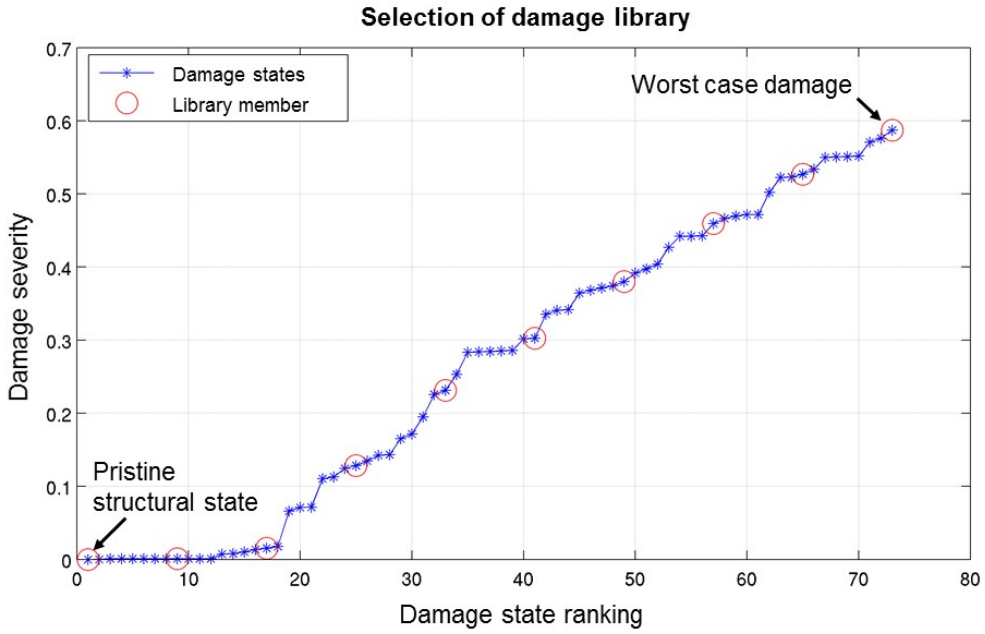

Figure 9. Selection of library. Damage states used for the library are circled in red.

III.C. Vehicle Models

III.C.1. Aerostructural Model

The aerostructural model used to generate the PSVM library as well as the lift, drag, and strain quantities for all damage states assessed, is a combination of ASWING[33] and Variational Asymptotic Beam Cross-Sectional Analysis (VABS).[34] ASWING is a nonlinear aerostructural solver for flexible-body aircraft configurations of high to moderate aspect ratio. We use ASWING to calculate internal wing loads and deflections as functions of input vehicle state and control. We can obtain internal structural loads for both static and dynamic flight conditions but restrict ourselves to quasi-static maneuvers. VABS is used primarily to resolve local effects due to stiffness weakness of the aircraft wing for different damage states. Full implementation details for this aerostructural model are given in Ref. 9. It is important to note that the extent of damage modeling of this tool is reduction in the moduli of damaged elements in the 2D cross section of VABS via the k_{loss} parameter. More sophisticated damage models can be described with higher-fidelity modeling such as isogeometric analysis of thin shell composites.[35]

III.C.2. Vehicle Transition Model

For the vehicle transition model, we restrict ourselves to static damage for the assessment, i.e., $d_{k+1} = d_k$. This model takes the form:

$$f(s,u,w) = \begin{bmatrix} f_{\mathcal{X}}(s,u,w) \\ d \end{bmatrix}. \tag{13}$$

The function $f_{\mathcal{X}}$ represents the transition model for the kinematic state which we take as the point-mass zero-side slip airplane model with an additive noise term:[36]

$$f_{\mathcal{X}}(s,u,w) = x + \begin{bmatrix} V\cos\gamma\cos\psi \\ V\cos\gamma\sin\psi \\ V\sin\gamma \\ \frac{T\cos\alpha - D(s,u)}{m} - g\sin\gamma \\ \frac{L(s,u)+T\sin\alpha}{mV}\cos\phi - \frac{g}{V}\cos\gamma \\ \frac{L(s,u)+T\sin\alpha}{mV\cos\gamma}\sin\phi \end{bmatrix} \Delta t + w, \tag{14}$$

where $x = [x_g, y_g, z_g, V, \gamma, \psi]^T$, $u = [T, \alpha, \phi]^T$, and $w \sim \mathcal{N}(0, W)$. Here x_g, y_g, z_g denote global x,y,z-coordinates, V is the velocity, γ is the flight path angle, ψ is the heading angle, T is the thrust force, α is the angle of attack, ϕ is bank angle, L is the lift force, D is the drag force, m is the mass, g is the gravitational acceleration, Δt is the timestep, and w is the transition model noise term with covariance matrix W. Parameters lift L and drag D are calculated at sampled maneuver points $[x, u]$ for a given damage state d in the library using ASWING+VABS and fit with response surface models for use of the form:

$$\begin{aligned} L(s,u) &= \sum_{n=0}^{N_r} \alpha_L^{(n)}(d)\psi_n(x,u) \\ D(s,u) &= \sum_{n=0}^{N_r} \alpha_D^{(n)}(d)\psi_n(x,u). \end{aligned} \tag{15}$$

where $\alpha_L^{(n)}$ is the n^{th} coefficient in the expansion for lift, $\alpha_D^{(n)}$ is the n^{th} coefficient in the expansion for drag, ψ_n is the n^{th} response surface basis function, and N_r is the number of response surface basis functions in the expansion.

The control is expressed as $u = u_{\text{trim}} + \Delta u$, where the u_{trim} term represents the control component that ensures trim conditions of the aircraft ($\dot{\gamma} = 0$ and $\dot{V} = 0$), and Δu represents the perturbations from these conditions and is the component used for path planning. Trim conditions are found by solving the following set of equations for α_{trim} at each timestep:

$$\begin{aligned} \frac{T_{\text{trim}}\cos\alpha_{\text{trim}} - D(s,u_{\text{trim}})}{m} - g\sin\gamma = 0, \text{ Constant Velocity} \\ \frac{L(s,u_{\text{trim}}) + T_{\text{trim}}\sin\alpha_{\text{trim}}}{m}\cos\phi - g\cos\gamma = 0, \text{ Constant Flight Path Angle} \end{aligned} \tag{16}$$

Note that changes in stability of the UAV as a result of damage (changes in lift and drag as a result of damage) are handled by the trim component of the control. Time integration of Eq. 14 is approximately computed using a fourth-order Runga-Kutta scheme and the trim condition equations in Eq. 16 are solved using Newton-Raphson iteration.

The available thrust T_{avail} derives from a simple constant power (P) engine model ($T_{\text{avail}} = P/V$) where the engines are modeled after two Austro Engine AE300 Series (168 hp peak).[37,38] The total thrust is bounded between zero thrust and the available thrust:

$$0 \leq T_{\text{trim}} + \Delta T \leq T_{\text{avail}}. \tag{17}$$

III.C.3. Vehicle Observation Model

For the vehicle observation model, we use direct measurements of all kinematic state variables and strain sensor output from different locations on the vehicle wing. This model takes the form:

$$h(s,u,\nu) = \begin{bmatrix} x \\ \sum_{n=0}^{N_r} \alpha_{z_s}^{(n)}(d)\psi_n(x,u) \end{bmatrix} + \nu, \tag{18}$$

where $\nu \sim \mathcal{N}(0,Q)$. Here $\alpha_{z_s}^{(n)}$ is the n^{th}coefficient (a vector containing all strain sensor components and locations) in the expansion of the strain quantities and ν is the observation model noise term with covariance matrix Q. As is done for the parameters lift and drag, the strain sensor model is constructed for the damage states in the library by evaluating strains using ASWING+VABS at sampled maneuver points $[x,u]$ and fitting with response surface models, per Step 1 in Subsection II.F. Strain sensors are placed at $0.1l_{\text{wing}}$ spanwise increments at chord-wise locations $0.2c_{\text{wing}}$ and $0.7c_{\text{wing}}$ from the side of body to the wing tip for both the left and right wings, as shown in Figure 8.

III.D. Path Planning using Dynamic Programming

We solve Eq. 10 using dynamic programming. In particular, the optimal value function $V^*(b)$ can be rewritten into the following Bellman equation:

$$\begin{aligned} V^*(b) = &\max_{u\in\mathcal{U}} \left\{ r(b,u) + \lambda \int_{z\in\mathcal{Z}} V^*(\tau(b,u,z))p(z|b,u)\ \mathrm{d}z \right\} \\ &s.t. \quad c(b,u) \geq p_{\text{thresh}}, \end{aligned} \tag{19}$$

where

$$\begin{aligned} r(b,u) &= \int_{s\in\mathcal{S}} r(s,u)b(s)\ \mathrm{d}s, \\ c(b,u) &= \int_{s\in\mathcal{S}} c(s,u)b(s)\ \mathrm{d}s. \end{aligned} \tag{20}$$

This equation can be solved approximately by using a grid-based approach and interpolation scheme[39] as described in the following paragraphs. The grid-based approach is attractive in order to handle the constraint in a simple and direct manner. Furthermore, it converts the continuous state CPOMDP to a grid-based Markov Decision Process (MDP) over the belief states.

We begin by defining a finite set of belief states $\mathcal{B}_G \subset \mathcal{B}$ for which each element $b^i \in \mathcal{B}_G$ defines a single belief distribution over the total vehicle state space. The Bellman equation over $\mathcal{B}_G$ is then expressed as:

$$V^*(b^i) = \max_{u\in\mathcal{U}_i} \left\{ r(b^i,u) + \lambda \int_{z\in\mathcal{Z}} V^*(\tau(b^i,u,z))p(z|b^i,u)\ \mathrm{d}z \right\} \tag{21}$$

where the constraint is now encoded in $\mathcal{U}_i$.

In general, the Bayes posterior τ will not be a member of $\mathcal{B}_G$ and therefore interpolation between belief states in $\mathcal{B}_G$ is done, as illustrated in Figure 10.

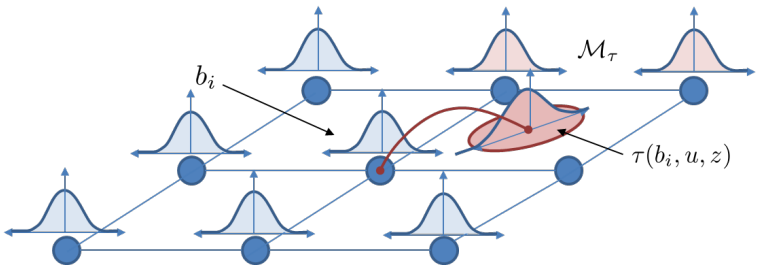

Figure 10. Discretization of belief states of $\mathcal{B}_G$ and interpolation of transitions.

The interpolation scheme we use is the following:

$$V^*(\tau) \approx \eta_\tau \sum_{j \in \mathcal{M}_\tau} \varphi(\tau, b^j) V^*(b^j) \tag{22}$$

where φ is the interpolating function, $\eta_\tau^{-1} = \sum_{j \in \mathcal{M}_\tau} \varphi(\tau, b^j)$, and $\mathcal{M}_\tau$ denotes the set of the M belief states $b^j \in \mathcal{B}_G$ with the largest value of $\varphi(\tau, b^j)$. The purpose of $\mathcal{M}_\tau$ is to truncate the sum in Eq. 22 to those belief states with the highest contribution rather than retaining all belief states in $\mathcal{B}_G$ in the sum. For the interpolating function, we use

$$\varphi(\tau, b^j) = \exp\left\{ \frac{1}{\alpha_1} \left[D_{\mathrm{KL}}(\tau || b^j) + D_{\mathrm{KL}}(b^j || \tau) \right]^{\alpha_2} \right\} \tag{23}$$

Here D_{KL} denotes the Kullback-Leibler Divergence, and the variables α_1 and α_2 are tuning parameters. With the interpolation scheme defined in Eq. 22 and the associated interpolating function in Eq. 23 satisfying the set of convex rules ($0 \leq \eta_\tau \varphi(\tau, b^j) \leq 1$ and $\sum_{j=\mathcal{M}_\tau} \eta_\tau \varphi(\tau, b^j) = 1$),[39] the resulting finite-state Bellman equation is by design a contraction mapping, which yields a unique solution. Substituting the interpolation scheme into the Bellman equation, we arrive at:

$$V^*(b^i) = \max_{u \in \mathcal{U}_i} \left\{ r(b^i, u) + \lambda \int_{z \in \mathcal{Z}} \eta_\tau \sum_{j \in \mathcal{M}_\tau} \varphi(\tau, b^j) V^*(b^j) p(z | b^i, u) \, \mathrm{d}z \right\} \tag{24}$$

where the arguments of τ have been suppressed for brevity. To compute the expectation over the measurements Monte Carlo simulations[23] are performed. Eq. 24 is solved using value iteration.

Online, the control u for a given belief distribution b is determined by averaging using the same interpolation scheme. That is,

$$u(b) = \eta_b \sum_{i \in \mathcal{M}_b} u(b^i) \varphi(b, b^i). \tag{25}$$

IV. Results

This section provides the results of the example problem using the proposed methodology. Figure 11 shows realizations of the vehicle trajectory for the pristine case and a heavily damaged case for $p_{\mathrm{thresh}} = 0.9$. Immediately apparent is that the vehicle selects a longer and less aggressive route to the goal in the heavily damaged case. The realizations also show the stochastic nature of the problem where different routes can be selected based on the belief distribution the vehicle sees, even if the vehicle is in the pristine structural state.

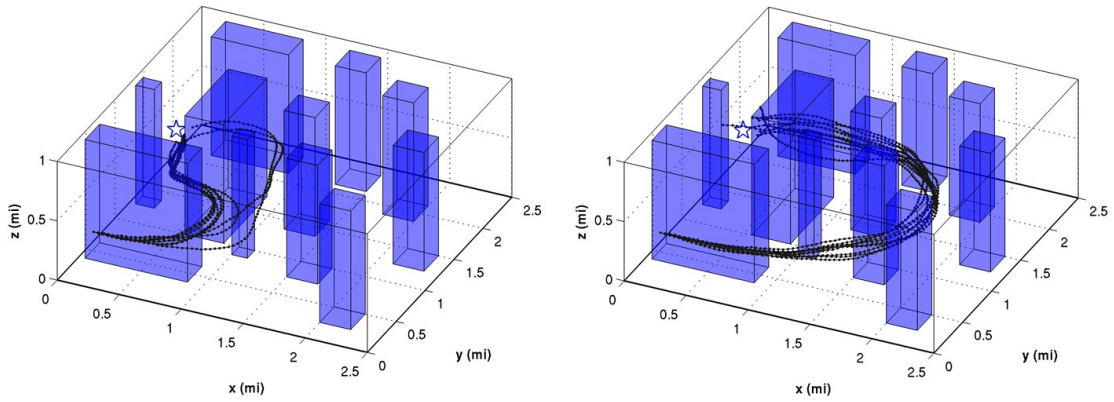

Figure 11. Realizations of vehicle trajectory for the cases where the vehicle is in the pristine state (left) and a heavily damaged state (right).

Survivability of the UAV across all missions is summarized in Table 3. For comparison, we include a baseline policy that makes no use of dynamic capability and performs maneuvers under the assumption that the UAV is in the pristine state at all times. For the test cases run, we see that the path planner with dynamic capability is able to increase total survivability by 15%. To examine further the performance of the path planner, Figure 12 shows the peak strain level during the mission versus the obtained distance to the goal. A successful mission is obtained if the vehicle is able to stay under the allowable strain level (here we have a safety factor of 2 for a threshold ratio $2\epsilon_{\text{peak}}/\epsilon_{\text{allow}} = 1$) while reaching within 0.2 miles of the goal. We see that a large fraction of missions under the baseline policy exceed the threshold peak strain, while the policy using dynamic capability is able to stay below the threshold for nearly all missions. The location of failures is visualized in Figure 14. The majority of failures under the baseline policy occur from exceeding allowable strain levels upon the vehicle entering the first corner between the two large obstacles.

Table 3. Survivability comparison of the path planner under the baseline and dynamic capability policy.

Policy	Survivability
Baseline	84%
Dynamic Capability	99%

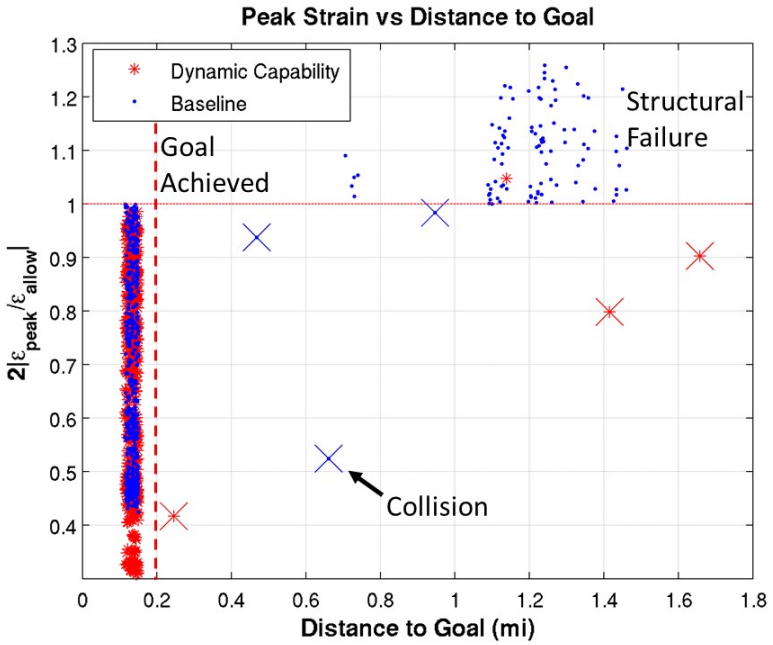

Figure 12. Peak strain during mission versus achieved distance to goal. "X" markers correspond to obstacle collisions.

We can understand how the policy under dynamic capability is performing for individual damage cases by looking at Figure 13. Here, we see two clusters corresponding to the two trajectories plotted in Figure 11. That is, the path planner selects different trajectories based on the particular damage state the vehicle is in, taking longer paths when necessary and taking shorter paths otherwise. In contrast, the baseline policy almost always attempts to take the fastest route and thus has a higher likelihood of exceeding allowable strain values and failing the mission for more heavily damaged scenarios. We note that due to the stochastic nature of the problem, even the baseline policy has a few cases where the path taken can vary, as can be seen by the cluster of points between 45-50 seconds in Figure 13.

In numerical implementation, in certain instances where a damage test case is similar to two cases in the library, the posterior damage state distribution can fluctuate rapidly without converging to a stable distribution. In other cases, a damage test case not similar to any member in the library can result in a posterior where the weights are more or less equal across all damage cases in the library. The first issue can result in conflicting control policies leading to a higher likelihood of collisions (the reason for obstacle collisions for the path planner with dynamic capability) as a result of these policies enforcing different trajectories. Both issues indicate insufficient knowledge about what state the vehicle is in and presses upon the assumption that the library contains sufficient depth and richness to adequately model the true vehicle state. However, fluctuation of this distribution can serve as a useful diagnostic to determine whether the vehicle has been damaged in a new way not captured in the library. This can then motivate further analysis, inspection of the vehicle, and updates to the library for the new cases encountered.

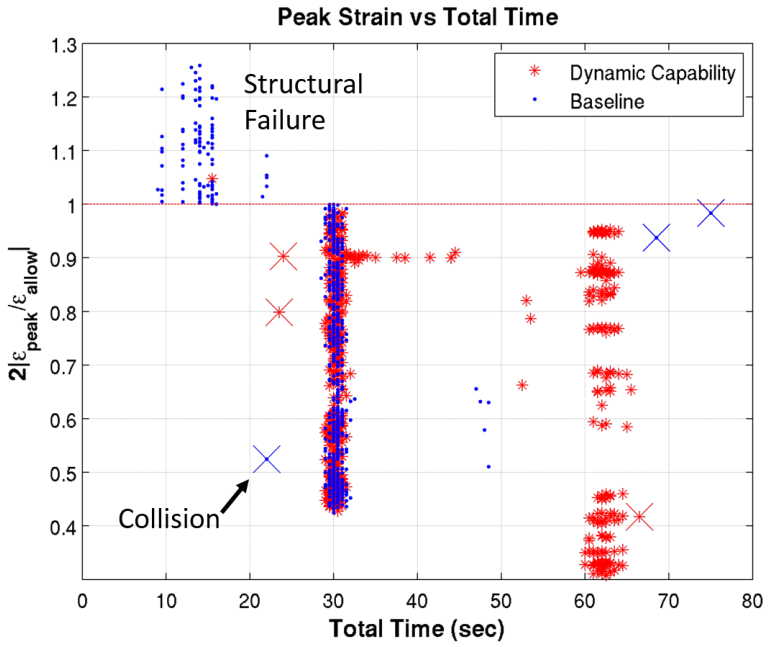

Figure 13. Time to goal vs peak strain during mission. "X" markers correspond to obstacle collisions.

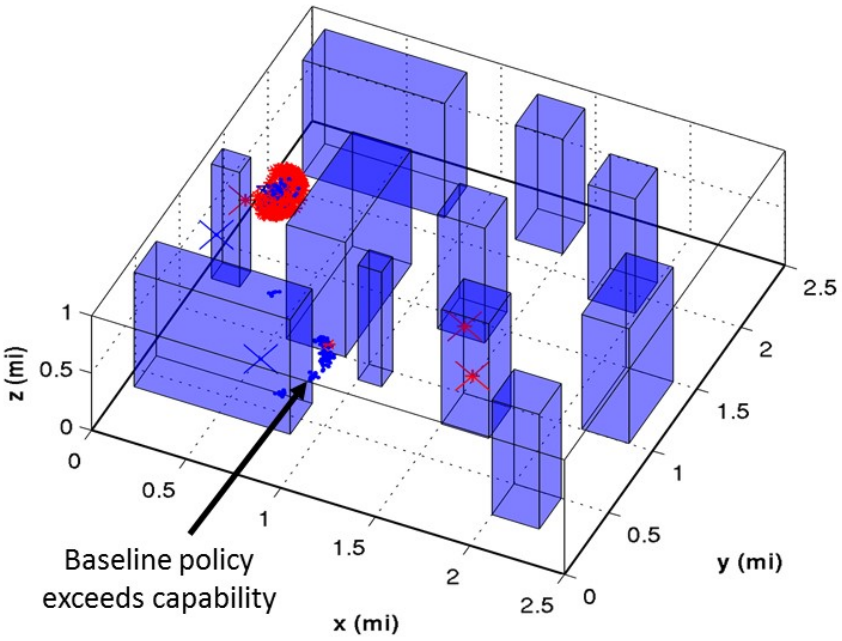

Figure 14. Locations reached before success/failure. The baseline policy fails as the UAV makes its way around the first corner and exceeding allowable strain levels as a result of the aggressive climb and tight turn."X" markers correspond to obstacle collisions.

V. Conclusions

This paper presented a data-driven methodology that leverages offline vehicle design information together with onboard sensor information to achieve dynamic path planning. An illustrative example highlighted the key properties of the approach where a vehicle is tasked to reach a target location while avoiding obstacles

and staying within capability. The example showed the intricate nature of the problem where, based on what the vehicle senses, different trajectories can be taken to the goal location. Heavily damaged cases result in conservative control actions while less damaged cases yield more aggressive actions to the goal. Results show that a policy that uses dynamic capability has a higher chance of survival when compared to a baseline policy that only knows of the pristine structural state. We note that controlling a damaged UAV also involves accommodation of changes in handling characteristics. Modeling such changes requires models of sufficient fidelity. For our implementation, the extent of damage modeling is via stiffness weakness and not drastic changes in the UAV aerodynamic profile (i.e., removal of airframe sections). As a result, lift and drag profiles turn out to be very similar to the pristine case. This is not a limitation of the methodology but of the limitation of the tools used for implementation.

In the example problem, we solved Eq. 10 approximately using dynamic programming and a grid-based approach. Discretization of the vehicle state space comes with some nuisances. Too fine a discretization suffers from the curse of dimensionality, while too coarse of a discretization results in poor quality of the solution, spurious trajectories, and/or inadequate capturing of the corners of the reward and constraint functions. We note that the literature on path planning and control is vast, however the path planning problem can be broken down into two main sub-problems: motion planning and trajectory tracking. For larger and more sophisticated environments and state spaces, numerous algorithms exist for both and can be used interchangeably to solve the motion planning and trajectory tracking problem inherent in the optimization statement of Eq. 10.

Future work will address larger and more sophisticated environments, wind conditions, different sensor types, and higher-fidelity damage models. Nevertheless, results show promise towards an aerospace vehicle that can dynamically adapt its trajectory according to the observations it receives about its current state of health, thereby retaining a high probability of mission success and survivability.

Acknowledgments

This work was supported by funding from the Arthur Gelb Fellowship, by AFOSR grants FA9550-11-1-0339 and FA9550-16-1-0108 under the Dynamic Data-Driven Application Systems Program (program manager F. Darema), and by the United States Department of Energy Office of Advanced Scientific Computing Research (ASCR) grants DE-FG02-08ER2585 and DE-SC0009297, as part of the DiaMonD Multifaceted Mathematics Integrated Capability Center (program manager S. Lee). The UAV model was provided by Aurora Flight Sciences.

References

[1] "BAE Systems Developing "smart skin" for aircraft," `http://www.gizmag.com/bae-smartskin/33458/`, Accessed: 12/20/2015.

[2] D. Kordonowy and O. Toupet, "Composite Airframe Condition-Aware Maneuverability and Survivability for Unmanned Aerial Vehicles," *Infotech@Aerospace 2011*, American Institute of Aeronautics and Astronautics, Reston, VA, March 2011, No. 2011-1496.

[3] H. Sohn and C. Farrar, "Damage diagnosis using time series analysis of vibration signals," *Smart Materials and Structures*, Vol. 10, No. 3, 2001, pp. 446–451.

[4] Kirk, B., Schagaev, P. I., Wittig, T., Kintis, A., Kaegi, T., Friedrich, F., Ag, E. T., Spirit, S. A., Avenue, S., and Faliro, P., "Active Safety for Aviation," *6th INO Workshop*, EUROCONTROL Experimental Centre (EEC), 2007.

[5] C. Farrar and N. Lieven, "Damage prognosis: the future of structural health monitoring," *Philosophical Transactions of the Royal Society A: Mathematical, Physical, and Engineering Sciences*, Vol. 365, No. 1851, 2007, pp. 623–632.

[6] S. Willis, "OLM: A Hand-On Approach," *in "ICAF 2009, Bridging the Gap between Theory and Operational Practice"*, 2009, pp. 1199–1214.

[7] E. Prudencio, P. Bauman, S. Williams, D. Faghihi, K. Ravi-Chandar, J. Oden, "A Dynamic Data Driven Application System for Real-time Monitoring of Stochastic Damage," *Procedia Computer Science*, Vol. 18, 2013, pp. 2056–2065.

[8] H.J. Koolstra and H.J. Damveld, "Envelope Determination of Damaged Aircraft," *in "AIAA Guidance, Navigation, and Control Conference, 13-16 August, Minneapolis, Minnesota"*, 2012.

[9] M. Lecerf, "Methodology for Dynamic Data-Driven Online Flight Capability Estimation," *AIAA*, Vol. 53, No. 10, 2015, pp. 3073–3087.

[10]G.H. Shaw and M.A. Hill, "Flight Dynamics Modeling and Simulation of a Damaged Transport Aircraft," *in "AIAA Modeling and Simulation Technologies Conference, 13-16 August, Minneapolis, Minnesota"*, 2012.

[11]J. Kim, K. Palaniappan, and P.K. Menon, "Rapid Estimation of Impaired-Aircraft Aerodynamic Parameters," *Journal of Aircraft*, Vol. 47, No. 4, 2010, pp. 1216–1228.

[12]P.K. Menon, P. Sengupta, S. Vaddi, B. Yang, J. Kwan, "Impaired Aircraft Performance Envelope Estimation," *Journal of Aircraft*, Vol. 50, No. 2, 2013, pp. 410–424.

[13]A. Adler, A. Bar-Gill, and N. Shimkin, "Optimal flight paths for engine-out emergency landing," *in Proceedings of the Chinese Control and Decision Conference*, 2012, pp. 2908–2915.

[14]E.M. Atkins, I.A. Portillo, and M.J. Strube, "Emergency Flight Planning Applied to Total Loss of Thrust," *Journal of Aircraft*, Vol. 43, No. 4, 2006, pp. 1205–1216.

[15]D.B. Jourdan, M.D. Piedmonte, V. Gavrilets, D.W. Vos, and J. McCormick, "Enhancing UAV Surviviability Through Damage Tolerant Control," *in "AIAA Guidance, Navigation, and Control Conference, 2-5 August, Toronto, Ontario Canada"*, 2010.

[16]N.T. Nguyen, K.S. Krishnakumar, J.T. Kaneshige, P.P. Nespeca, "Flight Dynamics and Hybrid Adaptive Control of Damage Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 410–424.

[17]V. Stepanyan, S. Campbell, and K. Krishnakumar, "Adaptive Control of a Damaged Transport Aircraft using M-MRAC," *in "AIAA Guidance, Navigation, and Control Conference, 2-5 August, Toronto, Ontario Canada"*, 2010.

[18]C. Edwards, T. Lombaerts, and H. Smaili, "Fault Tolerant Flight Control," *Lecture Notes in Control and Information Sciences*, Vol. 399, 2010.

[19]N. Meuleau, C. Plaunt, D.E. Smith, and T. Smith, "An Emergency Landing Planner for Damaged Aircraft," *in "Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference"*, 2009, pp. 114–121.

[20]I. Lopez and N. Sarigul-Klijn, *MICAI 2009: Advances in Artificial Intelligence*, chap. Intelligent Aircraft Damage Assessment, Trajectory Planning, and Decision-Making under Uncertainty, Springer Berlin Heidelberg, 2009, pp. 99–111.

[21]N.K. Ure, G. Chowdhary, J.P. How, M.A. Vavrina, J. Vian, "Health Aware Planning Under Uncertainty for UAV Mission with Heterogeneous Teams," *in "European Control Conference, 17-19 August, Zurich, Switzerland"*, 2013.

[22]D. Asadi, M. Sabzehparvar, E.M. Atkins, and H.A. Talebi, "Damaged Airplane Trajectory Planning Based on Flight Envelope and Motion Primitives," *Journal of Aircraft*, Vol. 51, No. 6, 2014, pp. 1740–1757.

[23]S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, 2006.

[24]D.P. Bertsekas, *Dynamic Programming and Optimal Control, Vol I, 3rd Edition*, Athena Scientific, 2005.

[25]Basudhar, A. and Missoum, S., "An improved adaptive sampling scheme for the construction of explicit boundaries," *Structural and Multidisciplinary Optimization*, Vol. 42, No. 4, May 2010, pp. 517–529.

[26]L.P. Kaelbling, M. L. Littman, A.R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, Vol. 101, 1998, pp. 99–134.

[27]G. Kewlani, G. Ishigami, and K. Iagnemma, "Stochastic Mobility-based Path Planning in Uncertain Environments," *in "IEEE/RSJ International Conference on Intelligent Robots and Systems, 11-15 October, St. Louis, USA"*, 2009.

[28]A. Bry and N. Roy, "Rapidly-Exploring Random Belief Trees for Motion Planning Under Uncertainty," *in "Proceedings of the IEEE International Conference on Robotics and Automation"*, 2011.

[29]H. Kurniawati, D. Hsu, and W.S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," *in Proc. Robotics: Science and Systems*, 2008.

[30]T. Smith and R. Simmons, "Heuristic search value iteration for POMDPs," *in Proceedings of the 20th conference on Uncertainty in Artificial Intelligence*, 2004, pp. 520–527.

[31]T. Smith and R. Simmons, "Point-based POMDP algorithms: Improved analysis and implementation," *in Proceedings Uncertainty in Artificial Intelligence*, 2005.

[32]A. Somani, N. Ye, D. Hsu, and W.S. Lee, "Despot: Online pomdp planning with regularization," *in Advances in Neural Information Processing Sytems*, 2013, pp. 1772–1780.

[33]Drela, M., "Integrated Simulation Model for Preliminary Aerodynamic, Structural, and Control-Law Design of Aircraft," *in "Proceedings of the 40th AIAA SDM Conference," AIAA, St. Louis, MO, Paper 99-1394*, 1999.

[34]C.E.S. Cesnik and D.H. Hodges, "VABS: A New Concept for Composite Rotor Blade Cross-Sectional Modeling," *Journal of the American Helicopter Society*, Vol. 42, No. 1, 1997, pp. 27–38.

[35]X. Deng, A. Korobenko, J. Yan, and Y. Bazilevz, "Isogeometric analysis of continuum damage in rotation-free composite shells," *Computer Methods in Applied Mechanics and Engineering*, Vol. 284, 2015, pp. 349 –372, Isogeometric Analysis Special Issue.

[36]S. Meng, J. Xiang, Z. Luo, Y. Ren, and N. Zhuang, "A novel trajectory planning strategy for aircraft emergency landing using Gauss pseudospectral method," *Control Theory and Technology*, Vol. 12, No. 4, 2014, pp. 393–401.

[37]"Aurora's Orion MALE UAV Aims for 120-hr. Flight," `http://aviationweek.com/awin/aurora-s-orion-male-uav-aims-120-hr-flight`, Accessed: 08/31/2015.

[38]"AE300 Technical Data Sheet," `http://austroengine.at/uploads/pdf/mod_products1/AE300_Technical_Data_Sheet.pdf`, Accessed: 08/31/2015.

[39]M. Hauskrecht, "Value-Function Approximations for Partially Observable Markov Decision Processes," *Journal of Artificial Intelligence Research*, Vol. 13, 2000, pp. 33–95.