# 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 high-fidelity physics-based modeling 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 that through Global Positioning System signals, inertial mass unit outputs, 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

| | |
|---|---|
| $\mathcal{B}_G$ | Discrete belief space |
| $b \in \mathcal{B}$ | Belief state and space |
| $\mathcal{C}$ | Capability set |
| $C, c$ | Constraint and marginalized constraint function |
| $c_s$ | Damage severity parameter |
| $D_{\text{KL}}$ | Kullback-Leibler Divergence |
| $\mathcal{D}_l$ | Damage library |
| $d \in \mathcal{D}$ | Damage state and space |
| $\mathcal{D}_{\text{data}}$ | Data set of sampled points used for SVM and observation model training |
| $\mathbb{E}$ | Expectation operator |
| $f$ | Transition model |
| $f_{\mathcal{X}}$ | Kinematic state transition model |
| $h$ | Observation model |
| $k_{\text{loss}}$ | Material moduli loss of damaged elements |
| $l_c$ | Airframe structure affected by damage |
| $l_{\text{chord}}, w_{\text{chord}}$ | Chordwise location and width of damage |
| $l_{\text{span}}, w_{\text{span}}$ | Spanwise location and width of damage |
| $l_{\text{wing}}$, $c_{\text{wing}}$ | Length and chord of the wing |
| $\mathcal{M}_{b'}$ | $M$ largest contributing $b \in \mathcal{B}_G$ in $\varphi(b', b)$ |
| $m, g$ | Mass and gravitational acceleration |
| $N_p$ | Number of data points in $\mathcal{D}_{\text{data}}$ |
| $N_r$ | Number of response surface basis functions |
| $n_v, n_s$ | Number of quantities in $z_v$ and $z_s$, respectively |
| $O$ | Observation probability distribution |
| $P$ | Engine power |
| $p(\cdot)$ | Probability function |
| $p_{\text{thresh}}$ | Constraint threshold |
| $Q$ | Observation model noise term covariance matrix |
| $R, r$ | Reward and marginalized reward function |
| $S_v$ | Support Vector Machine discriminant |
| $s \in \mathcal{S}$ | Total vehicle state and space |
| $T, D, L$ | Thrust, drag, and lift force |
| $T_{\text{avail}}$ | Available engine thrust |

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

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

| | |
|---|---|
| $T_r$ | Transition probability distribution |
| $t_{\text{depth}}$ | Damage depth into the skin |
| $t_{\text{skin}}$ | Skin thickness of wing |
| $u \in \mathcal{U}$ | Control input and space |
| $\mathcal{V}$ | Capability volume |
| $V^\pi$ | Value function for control policy $\pi$ |
| $V, \psi, \gamma$ | Velocity, heading, and flight path angle |
| $V^*$ | Optimal value function |
| $W$ | Transition model noise term covariance matrix |
| $w$ | Transition model noise term |
| $x \in \mathcal{X}$ | Kinematic state vector and space |
| $x_g, y_g, z_g$ | Global x,y,z-coordinates |
| $z \in \mathcal{Z}$ | Measurement vector and space |
| $z_s \in \mathcal{Z}^s$ | Component of measurement vector relating to vehicle structural state and associated space |
| $z_v \in \mathcal{Z}^v$ | Component of measurement vector relating to vehicle kinematic state and associated space |
| $\alpha, \phi$ | Angle of attack and bank angle |
| $\alpha_1, \alpha_2$ | Interpolation function tuning parameters |
| $\alpha_L^{(n)}, \alpha_D^{(n)}$ | $n^{th}$ response surface basis function coefficient for lift and drag, respectively |
| $\alpha_{z_s}^{(n)}$ | $n^{th}$ response surface basis function coefficient for measurement quantities $z_s$ |
| $\beta_1, \beta_2$ | Probabilistic Support Vector Machine tuning parameters |
| $\Delta t$ | Time step |
| $\epsilon_{\text{allow}}$ | Allowable strain levels in structure |
| $\epsilon_{\text{peak}}$ | Peak strain levels in structure |
| $\eta$ | Appropriate normalizing term |
| $\lambda$ | Discount factor |
| $\nu$ | Observation model noise term |
| $\pi$ | Control policy |
| $\psi_n$ | $n^{th}$ response surface basis function |
| $\tau$ | Bayes posterior |
| $\varphi$ | Interpolation function in Bellman equation |

*Subscripts*

| | |
|---|---|
| $k$ | Quantity at the $k^{th}$ time step |
| trim | Designation of vehicle trim condition variables |

*Superscripts*

| | |
|---|---|
| $(i)$ | $i^{th}$ data point of $\mathcal{D}_{\text{data}}$ |

# I. Introduction

Recent advances and affordability in aerospace sensing technologies have brought forth a new era of large-scale data acquisition in the aerospace industry. Synonymous with Big Data or the Internet of Things (IoT), data from thousands of sensors on board aircraft along with data analytics will enable aerospace agencies to improve upon operational efficiencies of individual aircraft components as well as reduce costs in service and maintenance through monitoring and prognosis. Looking farther ahead, these sensing technologies will enable the next generation of self-aware unmanned aerial vehicles (UAVs) that can make decisions based on the observations they receive.[1] In particular, such vehicles will have higher reliability, survivability, and therefore reduced fiscal risk, by making informed decisions in flight that incorporate real-time monitoring of their health as well as the environment around them. Of particular relevance to the self-aware UAV concept are circumstances where structural damage is accumulated during a mission through discrete events or through fatigue over a long life cycle. Such in-flight decisions can potentially impact mission objectives such as the trajectory the UAV takes. As a consequence, path planning for a self-aware UAV that leverages online sensor information about its health is fundamental to the decision loop and is the topic of this paper.

A graphical depiction of a self-aware UAV is shown in Figure 1. Here onboard sensors embedded across the entire airframe collect data of internal conditions, external signals, and the environment around the UAV. In particular, if structural damage is encountered, it will manifest as a change in sensor output. The UAV will then use the sensor output to infer the change in its flight envelope and re-plan its mission if needed.

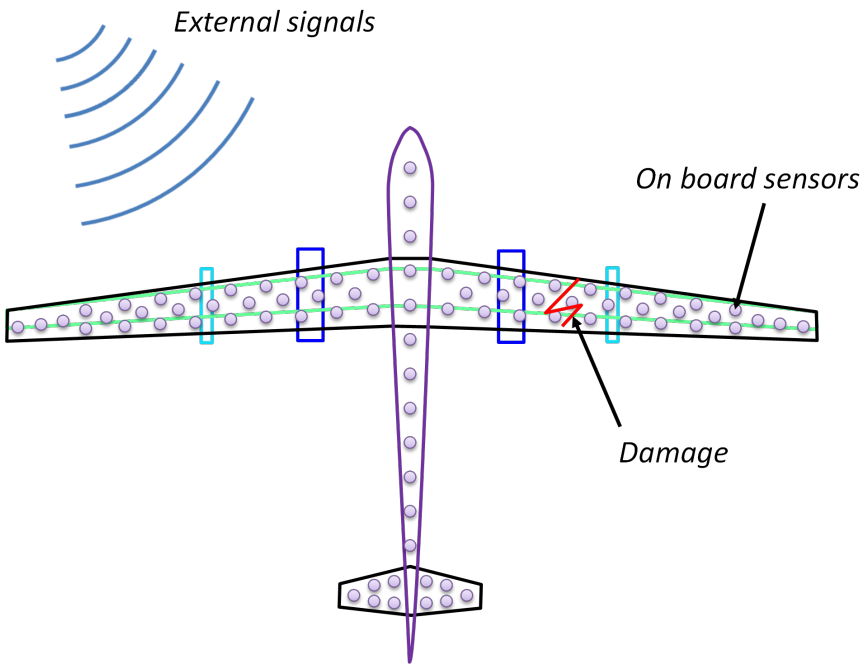


**Figure 1. Envisioned self-aware aerospace vehicle with sensors embedded across its airframe.**

Designing a path-planning strategy for a self-aware UAV foremost requires the ability to convert sensor data of damaged structure into estimates of the modified flight envelope. Such technologies have emerged from a synthesis of smart materials, machine learning, physics-based modeling, statistics, and signal processing within the structural health monitoring (SHM) community. For detection of structural damage in particular, guided waves, ultrasonic, and nonlinear acoustic techniques combined with signal array processing and transducers made of piezoelectric and/or optical fiber materials are becoming popular tools for detecting small cracks or microcracks.[2,3] In application, these tools find use in the broader class of Integrated Vehicle Health Management (IVHM) and Prognostic Health Management (PHM) systems for detection, diagnosis, and prognosis of faults in electrical, software, and structural subsystems.[4,5] Once damage is detected and characterized, the task of determining the modified flight envelope can be formulated as a pattern recognition problem.[6] One such approach is to construct libraries characterizing the flight envelope for a range of different damage events and trim motion primitives using computational or wind-tunnel experiments.[7–10] These libraries can be represented, queried, or modified using techniques in machine learning, filtering methods, or statistical inference for parameter estimation and determination of the modified flight envelope.[11,12] Parameters that are commonly estimated and used for subsequent control of damaged aircraft include stability/control derivatives, lift, drag, stall, and inertia shifts.[8,9]

Upon recognizing that the flight envelope or handling has changed significantly as a result of damage, a self-aware UAV must be able to maintain stability and tracking as well as change course if needed. Stability and tracking control systems for damaged aircraft, or alternatively reconfigurable flight control or fault tolerant flight control, can be accomplished with control laws that derive from adaptive control architectures, model reference control, model predictive control, or neural networks.[13–19] In these works, common damage scenarios include engine-out[20,21] or power loss conditions, loss of airframe structure (e.g., vertical or horizontal tail loss, partial wing loss), structural softening, or jamming of major control surfaces. Such control systems have found successful applications on Boeing UAVs (X-36, X-40A, X-45, T-33), General Dynamics's VISTA F-16, and Eurofighter Typhoon.[4] In the context of path planning, planning solutions generally account for reduced 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.[16,22–25]

Building upon the techniques discussed, this paper focuses in particular on the path-planning problem involved for a single self-aware UAV that becomes damaged but is equipped with sensors that can monitor its dynamics and internal state of health. An illustrative scenario of interest is depicted in Figure 2. Here a

self-aware UAV is initially tasked to navigate through a series of obstacles to the target location in minimum time. 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 path planning solution is inherently data-driven and thus embodies the ideas of Dynamic Data-Driven Application Systems[26] (DDDAS), where observational data are used in a feedback loop to drive the actions the UAV takes. In addition, the path-planning problem is complicated by modeling uncertainty and noise, fusion of sensor data from multiple modalities, and the possibility of a dynamically changing flight envelope based on further actions the UAV takes.

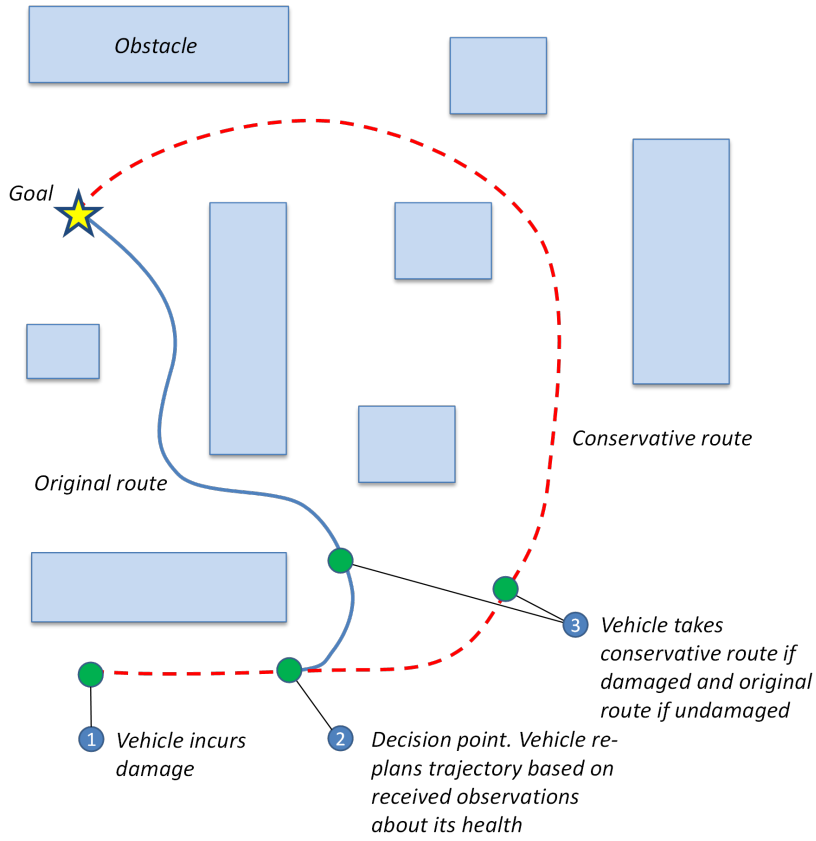


**Figure 2. Illustration of scenario of interest.**

To address this problem, we leverage a combination of physics-based modeling, machine learning, Bayesian statistics, and decision theory. We extend prior work on dynamic capability estimation from Ref. 12 into a probabilistic constraint model for path planning. The novelty in our approach lies in the ability to incorporate offline high-fidelity physics-based simulation results into the constraint model using probabilistic surrogates. As a result, we leverage a combination of both offline and online information sources to achieve gains in vehicle operation. In addition, the method we propose is flexible with respect to the physics-based models describing the aerostructural and dynamical characteristics of the vehicle and measurement processes. We demonstrate our methodology on a 3-D path planning example, where a UAV is tasked to reach a target location while avoiding obstacles and not exceeding available capability as a result of damage. 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.

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 the belief state representation. 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. 12. 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.

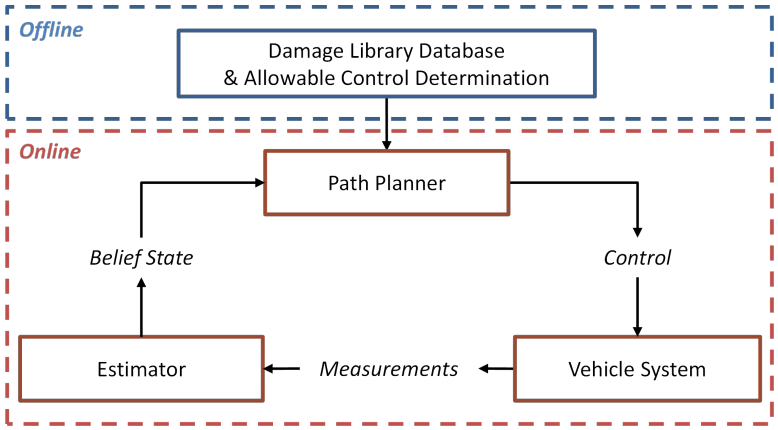


**Figure 3. Offline/online approach to proposed dynamic path planner.**

## 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 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 (which may be represented by a finite parametrization involving severity, location, and size; or may be represented by a field quantity such as that used in continuum damage mechanics) 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 $k \in \mathbb{N}_0$ 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. Belief State Representation 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.[27] 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:[27]

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

where $\eta_k^{-1} = p(z_k|b_{k-1}, u_{k-1})$ is the normalizing term. The function $\tau : \mathcal{B} \times \mathcal{U} \times \mathcal{Z} \to \mathcal{B}$ is the Bayes posterior and can be thought of as the transition model for the belief state. The belief state together with $\tau$ provides an equivalent representation of the vehicle system, originally expressed with the two models $f$ and $h$, in terms of a single new model given by $\tau$. This representation is used in the path planning formulation.

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,[28] the control can be expressed as a function of the current belief state.

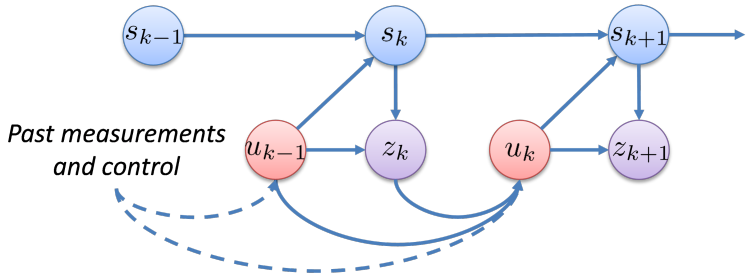


**Figure 4. Hidden Markov model for state evolution. The vehicle states (blue) are hidden and so the control action is a function of past measurements and control. It can be expressed as a function of the current belief state.**

## 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. Here $[x, u]$ corresponds to a particular maneuver such as pull-up, pull-down, banked turn, or any other combination of vehicle kinematic state and control. A maneuver is feasible if it does not exceed structural, engine, or aerodynamic limits such as stall, maximum velocity, or maximum structural component strengths. Thus, $\mathcal{C}$ is the space of feasible maneuvers.

To quantify capability, we assign a probability of whether a maneuver is feasible given a particular damage state configuration by the following probability distribution:

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

Note we could have just as easily defined a generative model where we infer $p([s, u] \in \mathcal{C})$. However, constructing and working with the conditional distribution Eq. 5 is of more interest to us since vehicle constraints subject to specific damage finds additional practical relevance for design tasks (structural sizing and controller design), an example being for certification of rotor-burst conditions required by the Federal Aviation Administration.[29] Eq. 5 is evaluated for each damage state $d$ of interest as outlined in Subsection II.F and is used to form the offline damage library.

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

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

To put Eq. 5 in a form amenable to calculation using the damage library, we marginalize over the damage state $d$ using the Law of Total Probability and apply the Markov model assumption of Subsection II.D, yielding:

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

which is a weighted average of Eq. 5 with respect to the marginal posterior distribution of $d$. Note that this expectation changes in time based on the evolution of the marginal posterior distribution $p(d|z_{0:k}, u_{0:k-1})$ and hence encodes the notion of "dynamic capability" through a weighted average of this marginal posterior, as shown in Figure 5. This figure illustrates how the estimated capability (here parametrized in terms of velocity and bank angle) changes as a function of the marginal posterior damage state distribution. When this distribution is more heavily weighted towards states of lower damage, the estimated capability is higher since the contributions to the expectation in Eq. 7 come from states that have higher overall capability. In contrast, when the distribution is more heavily weighted towards states of higher damage, the estimated capability is lower since the contributions come from states that have reduced capability.

Alternatively, one can think of the marginal posterior of $d$ in the context of a model selection problem, where the marginal posterior is the probability that the vehicle damage state is described by the model given by $d$ 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.

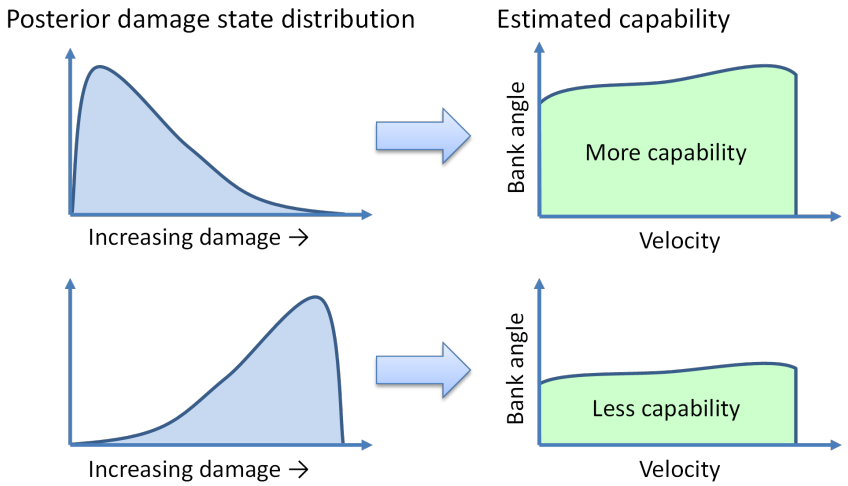


**Figure 5. Illustration of dynamic capability as a weighted average with the marginal posterior damage state distribution. Plots are notional.**

### 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. Construction of the capability and the observation model for each $d$ is as follows (the reader is referred to Ref. 12 for more details):

1. In the preliminary step, a data set $\mathcal{D}_{\text{data}}(d) = \{x^{(i)}, u^{(i)}, z^{(i)}, l^{(i)}\}_{i=1}^{N_p}$ for each damage state $d$ is generated where $(i)$ designates the $i^{th}$ data point of the $N_p$ total points in the data set, $[x^{(i)}, u^{(i)}]$ is a particular maneuver sampled from the combined vehicle state-control space $\mathcal{X} \times \mathcal{U}$, while $l^{(i)}$ is a label assigned $+1$ if a maneuver is feasible or $-1$ if a maneuver is infeasible. Infeasibility of a maneuver is assessed by running high-fidelity physics based models at the specified maneuver and determining whether that maneuver satisfies the requirements of feasibility given by our earlier definition. Initial sampling is performed using a design of experiments for $N_p$ sample points. The observation model $h$ is constructed for each $\mathcal{D}_{\text{data}}$ using a response surface or other technique.

2. In the second step, a Support Vector Machine (SVM) is generated for each $\mathcal{D}_{\text{data}}$. An adaptive sampling algorithm can be used to refine the SVM boundaries and augment the data sets if necessary.[30]

3. In the final step, the SVM discriminant is used as an argument to a sigmoid to generate the 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 6 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 7 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 $c_s : \mathcal{D} \to [0, 1]$ 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.

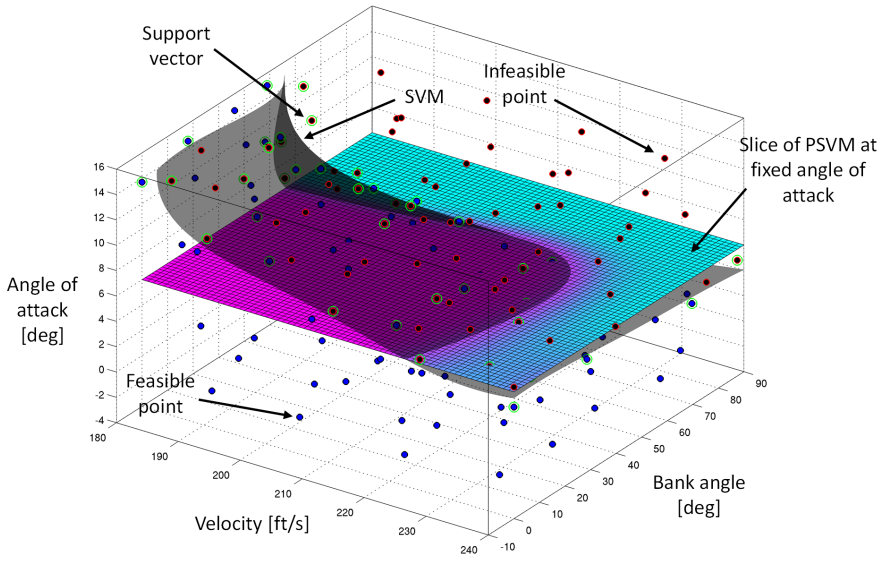


**Figure 6. Capability boundary and PSVM contour slice for a given vehicle damage state. Points underneath the boundary lie in the feasible maneuver region while points above lie in the infeasible region of the combined vehicle state-control space.**

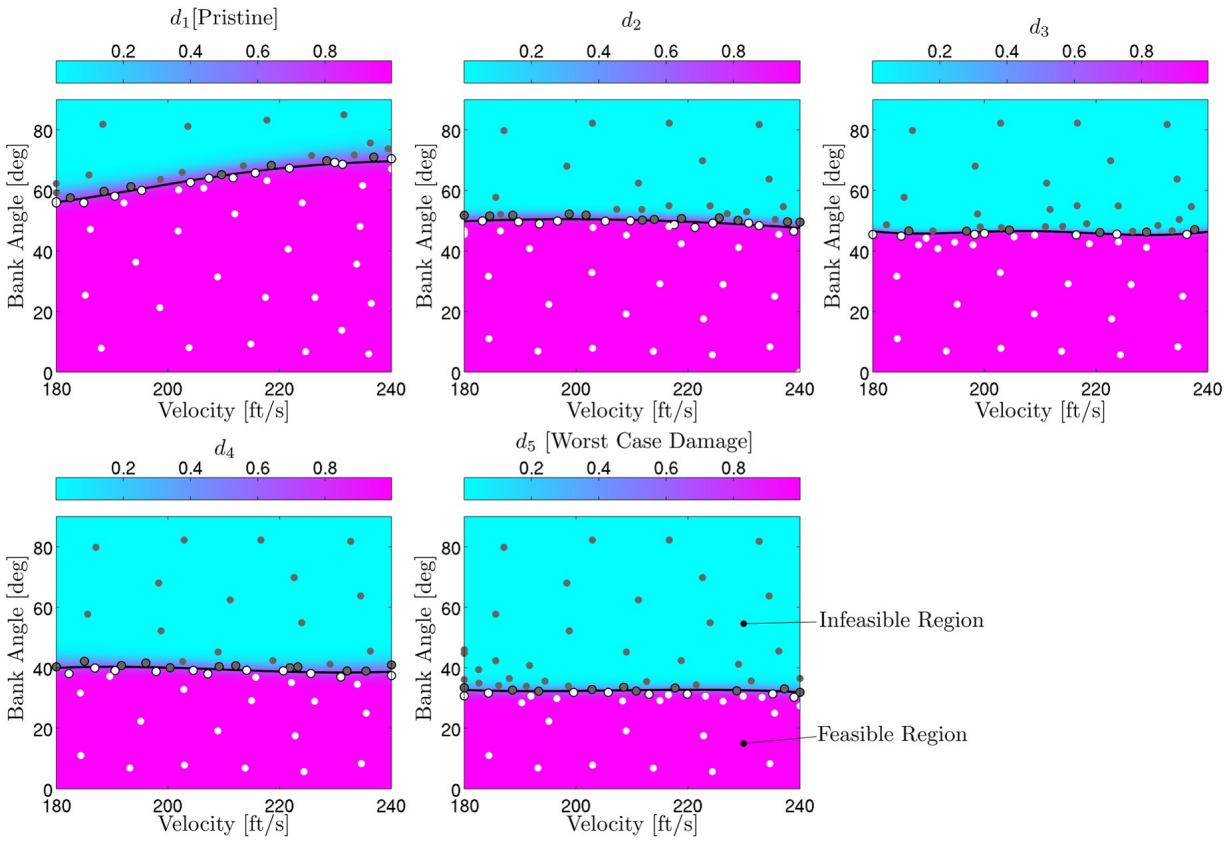


**Figure 7. Comparison of PSVM capability contours at a fixed angle of attack for different damage states. Damage states are ordered based on increasing $c_s$ from left to right and top to bottom.**

## II.G. Path Planning

We formulate the path planning problem as a Constrained Partially Observable Markov Decision Process (CPOMDP).[27,31] Formally, a CPOMDP is a tuple $\langle \mathcal{S}, \mathcal{U}, \mathcal{Z}, T_r, O, R, C, \lambda \rangle$ 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 $s'$ since what is of importance is that the transitioned state remains under the current dynamically estimated capability.

- $\lambda \in [0, 1)$ is the discount factor.

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:

$$\begin{aligned} \max_{\pi} V^{\pi}(b) = &\sum_{k=0}^{\infty} \lambda^k \mathbb{E}\big[r(s_k, u_k)|\pi, b_0 = b\big] \\ &s.t. \quad \mathbb{E}\big[c(s_k, u_k)|\pi, b_0 = b\big] \geq p_{\text{thresh}} \quad \forall k \in \mathbb{N}_0, \end{aligned} \tag{10}$$

where

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

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 $\pi$ and the initial belief state $b_0$ being equal to the input belief state $b$.

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,[32,33] and search-based algorithms.[34–37]

## 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 while 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 8. 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.

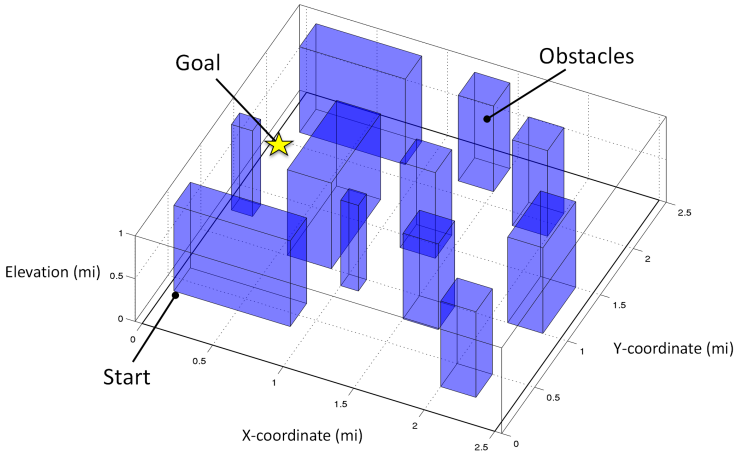


**Figure 8. 3-D environment for example problem.**

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.), $l_{\text{span}}$ is the spanwise location of damage, $w_{\text{span}}$ is the spanwise width of damage, $l_{\text{chord}}$ is the chordwise location of damage, $w_{\text{chord}}$ is the chordwise width of damage, $t_{\text{depth}}$ is the damage depth into the skin, and $k_{\text{loss}}$ is the material moduli 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 9. 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 10. Damage parametrizations 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] |
| $k_{\text{loss}}$ | 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 |
| $k_{\text{loss}}$ | 0 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |

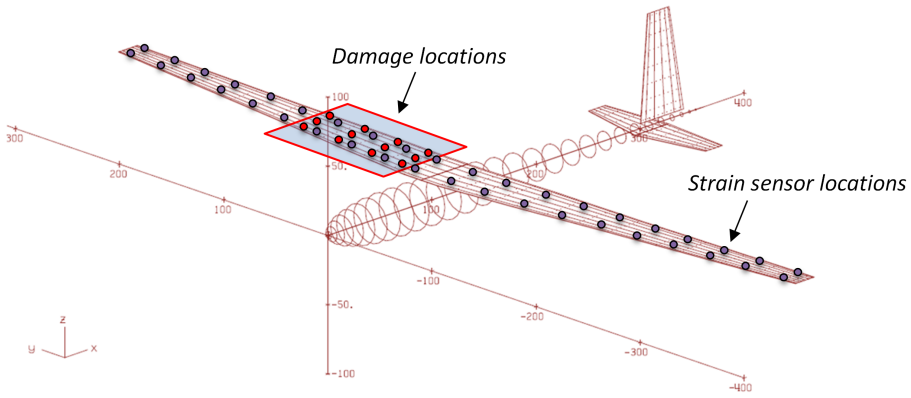


**Figure 9. Location of damage to the UAV and strain sensor placement.**

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 $\epsilon_{\text{peak}}$ across the airframe structure exceeds allowable levels $\epsilon_{\text{allow}}$ 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.

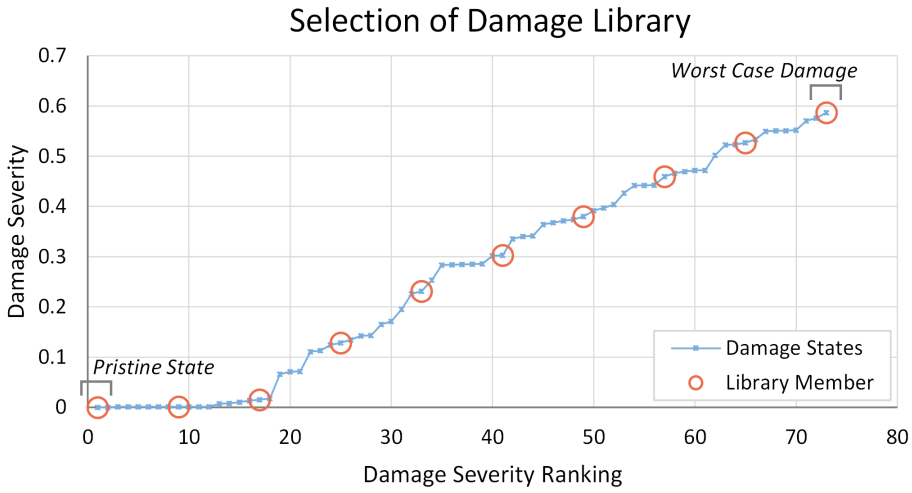


**Figure 10. Selection of library. Damage states used for the library are circled in red.**

## 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[38] and Variational Asymptotic Beam Cross-Sectional Analysis (VABS).[39] 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. 12. 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 $k_{\text{loss}}$ parameter. More sophisticated damage models can be described with higher-fidelity modeling such as continuum damage models of thin shell composites.[40]

### *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:[41]

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

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, $\gamma$ is the flight path angle, $\psi$ is the heading angle, $T$ is the thrust force, $\alpha$

is the angle of attack, $\phi$ is bank angle, $L$ is the lift force, $D$ is the drag force, $m$ is the mass, $g$ is the gravitational acceleration, $\Delta t$ 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, $\psi_n$ 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 $u_{\text{trim}}$ term represents the control component that ensures trim conditions of the aircraft ($\dot{\gamma} = 0$ and $\dot{V} = 0$), and $\Delta u$ 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 $\alpha_{\text{trim}}$ 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 $T_{\text{avail}}$ 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).[42,43] 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 $\nu$ 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 9.

### III.D. Path Planning Solution 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

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

This equation can be solved approximately by using a grid-based approach and interpolation scheme[44] 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.

Following Ref 44, 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 $\tau$ 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 11.

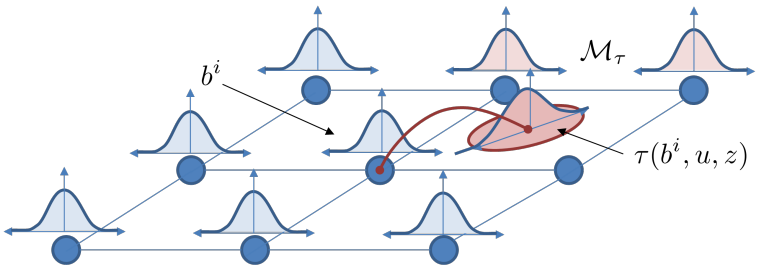


**Figure 11. Discretization of belief states of $\mathcal{B}_G$ and interpolation of transitions.**

The interpolation scheme takes the form:

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

where $\varphi$ 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. This then sets up a sparse coupling between states in the Bellman equation leading to memory efficiency. For the interpolating function, the following is used:

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

Here $D_{\mathrm{KL}}$ denotes the Kullback-Leibler Divergence, and the variables $\alpha_1$ and $\alpha_2$ 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 ($0 \le \eta_\tau\varphi(\tau,b^j) \le 1$ and $\sum_{j\in\mathcal{M}_\tau}\eta_\tau\varphi(\tau,b^j) = 1$),[44] 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 yields:

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

where the arguments of $\tau$ have been suppressed for brevity. To compute the expectation over the measurements, Monte Carlo simulations are performed following the sampling procedure of the MC-POMDP algorithm in Table 16.3 of Ref. 27. The approximate dynamic program represented in Eq. 24 is solved using

value iteration. This computation is performed offline where once solved, each belief point $b^i$ will have associated to it an approximate value of $V^*(b^i)$ and an action to take $u(b^i)$. Online, the control $u$ for a given belief distribution $b$ is determined by averaging over the offline control values using the same interpolation scheme. That is,

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

where $b$ comes from the state estimator. State estimation is performed using a Particle Filter.

## IV. Results

This section provides the results of the example problem using the proposed methodology. Figure 12 shows realizations of the vehicle trajectory for the pristine case and a heavily damaged case for $p_{\text{thresh}} = 0.9$. 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.

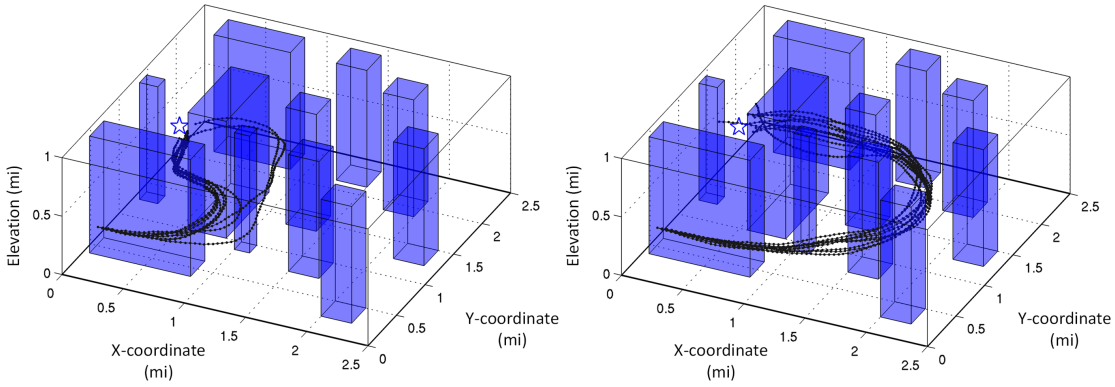


**Figure 12. Realizations of vehicle trajectory for the cases where the vehicle is in the pristine state (left) and a heavily damaged state (right).**

Survivability of the UAV across all 3650 missions is summarized in Figure 13. 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 14 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 all but one mission. The location of failures is visualized in Figure 16. 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.

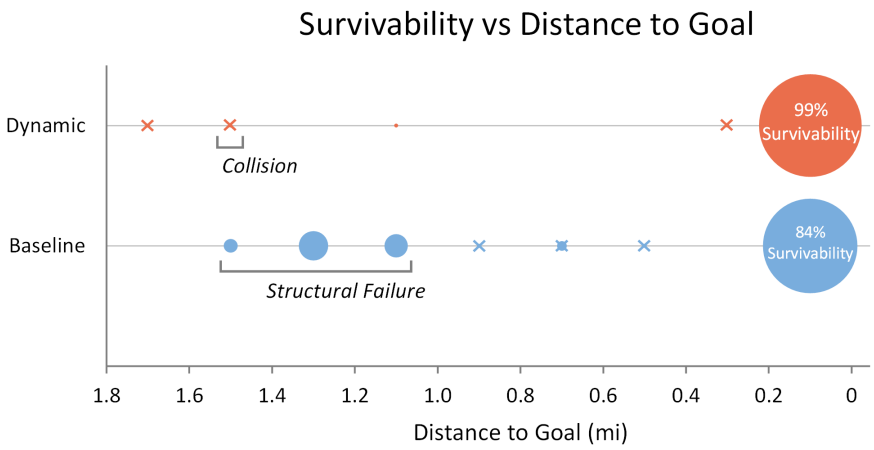


**Figure 13. Survivability comparison of the path planner under the baseline and dynamic capability policy.**

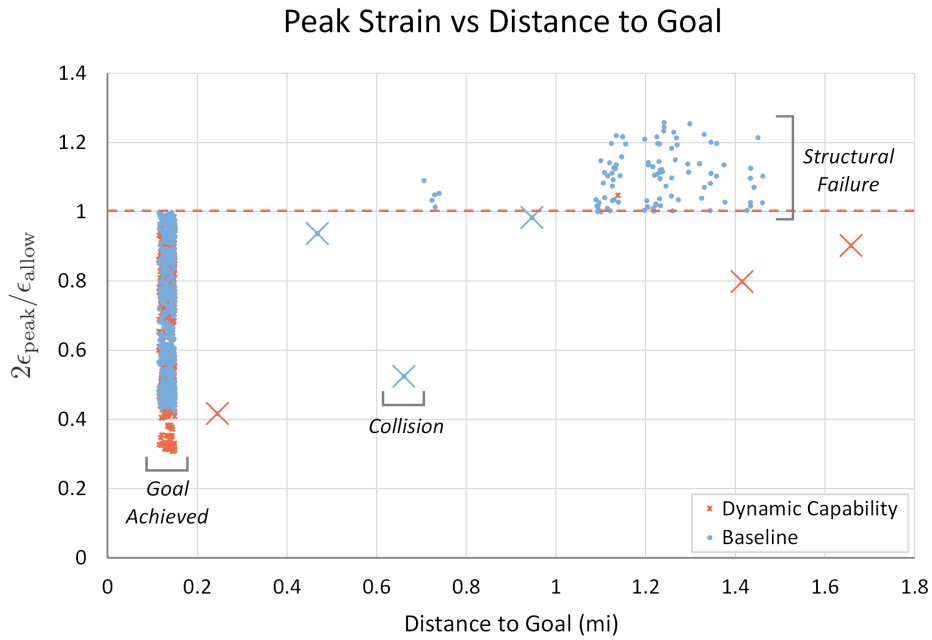


**Figure 14. Peak strain during mission versus achieved distance to goal. "X" markers correspond to obstacle collisions.**

We can understand how the policy under dynamic capability is performing for individual damage cases by looking at Figure 15. Here, we see two clusters corresponding to the two trajectories plotted in Figure 12. 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 15.

The rare collisions observed in the tests are the results of filter instability of the underlying Particle Filter used for state estimation. In these cases, the unstable filter outputs inconsistent state estimates that lead to conflicting control actions. These conflicting control actions then lead to a higher likelihood of obstacle collisions. The source of filter instability can be attributed to standard issues with the Particle Filter related

to degeneracy and re-sampling. These issues can be mitigated by using proper regularization techniques.

As mentioned in Subsection III.B, the offline library for the tests is sparse, which presses upon the assumption that the library contains sufficient depth and richness to adequately estimate all possible vehicle damage states. For cases where the damage test case was not in the library, we noticed three behaviors in the marginal distribution of the damage state as a consequence of this sparsity. The first and the most common was that the highest weight was given to the most similar library member while the other members received moderate to low weight. The second was that the weights were uniform across all members in the library. The third and the least common was rapid fluctuation of the weights resulting in a posterior that failed to converge. The third behavior was caused by the damage test case being different enough from the members in the library such that the likelihoods across all members were low. As a result, these low likelihoods introduced high numerical noise when normalizing to calculate the Particle Filter weights. On a higher level, the second and third behaviors indicate that the vehicle is unable to properly estimate its flight envelope using the current members in its library for the new damage case encountered. In addition, the severity of damage and its impact on the flight envelope for the new case can be unclear, especially if the damage is in a new physical location of the vehicle that is not in the library. The safest strategy to take when observing these behaviors is to assume the worst case scenario, which is what we implemented for the path planner with dynamic capability. Nevertheless, the behavior of the marginal damage state 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.

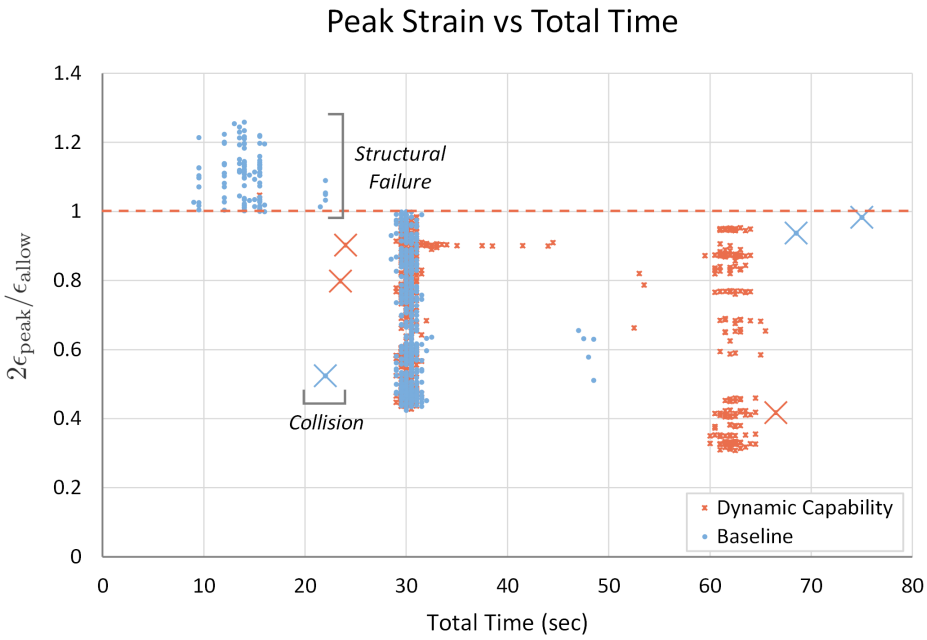


**Figure 15. Peak strain during mission versus total time to goal. "X" markers correspond to obstacle collisions.**

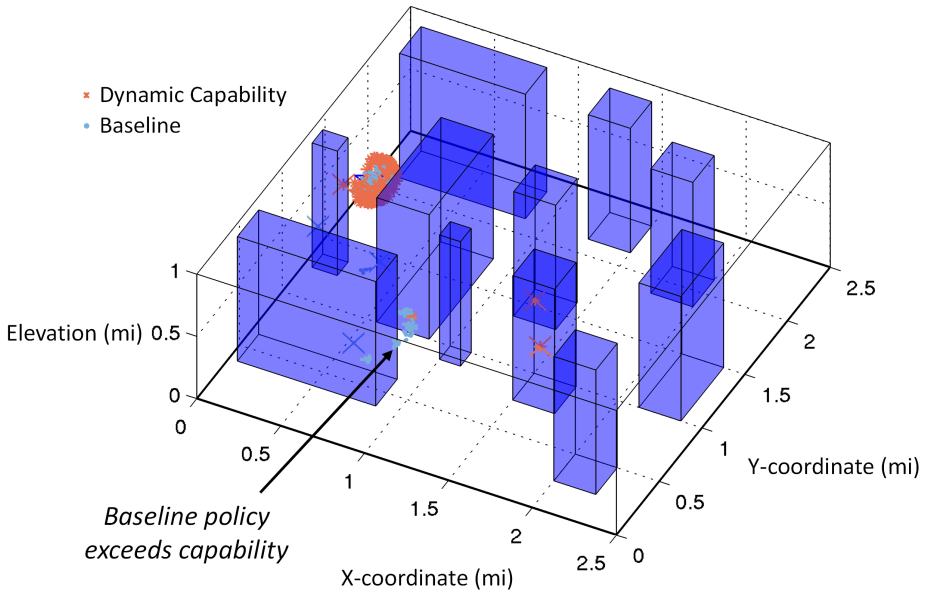


**Figure 16. Locations reached before success/failure. The baseline policy fails as the UAV makes its way around the first corner and exceeding allowable strain levels as a result of the aggressive climb and tight turn."X" markers correspond to obstacle collisions.**

## V. Conclusions

This paper presented a data-driven methodology that leverages offline high-fidelity physics-based models 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, sub-optimal policies, and 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]Kordonowy, D. and Toupet, O., "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.

[2]Staszewski, W., Boller, C., and Tomlinson, G., *Health Monitoring of Aerospace Structures: Smart Sensor Technologies and Signal Processing*, John Wiley & Sons, 2004.

[3]Yuan, F., *Structural Health Monitoring (SHM) in Aerospace Structures*, Woodhead Publishing, 2016.

[4]Esperon-Miguez, M., John, P., and Jennions, I., "A Review of Integrated Vehicle Health Management Tools for Legacy Platforms: Challenges and Opportunities," *Progress in Aerospace Sciences*, Vol. 56, 2013, pp. 19–34.

[5]Benedettini, O., Baines, T., Lightfoot, H., and Greenough, R., "State-of-the-Art in Integrated Vehicle Health Management," *in Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 223, No. 2, 2009, pp. 157–170.

[6]Farrar, C. and Worden, K., *Structural Health Monitoring: A Machine Learning Perspective*, John Wiley & Sons, 2012.

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

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

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

[10]Moncayo, H., Perhinschi, M., and Davis, J., "Artificial-Immune-System-Based Aircraft Failure Evaluation over Extended Flight Envelope," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, 2011, pp. 989–1001.

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

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

[13]Nguyen, N., Krishnakumar, K., Kaneshige, J., and Nespeca, P., "Flight Dynamics and Hybrid Adaptive Control of Damaged Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 751–764.

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

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

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

[17]Chowdhary, G., Johnson, E. N., Chandramohan, R., Kimbrell, M., and Calise, A., "Guidance and Control of Airplanes Under Actuator Failures and Severe Structural Damage," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp. 1093–1104.

[18]Liu, Y., Tao, G., and Joshi, S., "Modeling and Model Reference Adaptive Control of Aircraft with Asymmetric Damage," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 5, 2010, pp. 1500–1517.

[19]Lombaerts, T., Oort, E. V., Chu, Q., Mulder, J., and Joosten, D., "Online Aerodynamic Model Structure Selection and Parameter Estimation for Fault-Tolerant Control," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 3, 2010, pp. 707–723.

[20]Adler, A., Bar-Gill, A., and Shimkin, N., "Optimal Flight Paths for Engine-Out Emergency Landing," *in Proceedings of the Chinese Control and Decision Conference*, 2012, pp. 2908–2915.

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

[22]Lopez, I. and Sarigul-Klijn, N., *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.

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

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

[25]Sarigul-Klijn, N., Rapetti, R., Jordan, A., Lopez, I., Sarigul-Klijn, M., and Nespeca, P., "Intelligent Flight-Trajectory Generation to Maximize Safe-Outcome Probability After a Distress Event," *Journal of Aircraft*, Vol. 47, No. 1, 2010.

[26]Darema, F., "Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements," *Computational Science-ICCS 2004: 4th International Conference*, Springer, 2004, pp. 662–669.

[27]Thrun, S., Burgard, W., and Fox, D., *Probabilistic Robotics*, MIT Press, 2005.

[28]Bertsekas, D., *Dynamic Programming and Optimal Control*, Vol. I, Athena Scientific, 3rd ed., 2005.

[29]Federal Aviation Administration, "Design Considerations for Minimizing Hazards Caused by Uncontained Turbine Engine and Auxiliary Power Unit Rotor Failure," *FAA Advisory Circular No. 20-128A*, 1997.

[30]Basudhar, A. and Missoum, S., "An Improved Adaptive Sampling Scheme for the Construction of Explicit Boundaries," *Structural and Multidisciplinary Optimization*, Vol. 42, No. 4, 2010, pp. 517–529.

[31]Kaelbling, L., Littman, M. L., and Cassandra, A., "Planning and Acting in Partially Observable Stochastic Domains," *Artificial Intelligence*, Vol. 101, 1998, pp. 99–134.

[32]Kewlani, G., Ishigami, G., and Iagnemma, K., "Stochastic Mobility-based Path Planning in Uncertain Environments," *in "IEEE/RSJ International Conference on Intelligent Robots and Systems, 10-15 October, St. Louis, USA"*, 2009, pp. 1183–1189.

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

[34]Kurniawati, H., Hsu, D., and Lee, W., "SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces," *in Proc. Robotics: Science and Systems*, 2008.

[35]Smith, T. and Simmons, R., "Heuristic Search Value Iteration for POMDPs," *in Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 2004, pp. 520–527.

[36]Smith, T. and Simmons, R., "Point-Based POMDP Algorithms: Improved Analysis and Implementation," *in Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, 2005.

[37]Somani, A., Ye, N., Hsu, D., and Lee, W., "Despot: Online POMDP Planning with Regularization," *in Advances in Neural Information Processing Systems*, 2013, pp. 1772–1780.

[38]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.

[39]Cesnik, C. and Hodges, D., "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.

[40]Deng, X., Korobenko, A., Yan, J., and Bazilevs, Y., "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.

[41]Meng, S., Xiang, J., Luo, Z., Ren, Y., and Zhuang, N., "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.

[42]"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.

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

[44]Hauskrecht, M., "Value-Function Approximations for Partially Observable Markov Decision Processes," *Journal of Artificial Intelligence Research*, Vol. 13, 2000, pp. 33–94.