Database Building and Interpolation for an Online Safe Flight Envelope Prediction System

Y. Zhang*, C.C. de Visser† and Q.P. Chu.‡
Delft University of Technology, Delft, 2629HS, The Netherlands.

Nomenclature

b	=	wing span, m
C_l, C_m, C_n	=	dimensionless aerodynamic coefficient of roll, pitch and yaw moments in body axes
C_D, C_Y, C_L	=	dimensionless aerodynamic coefficient of drag force, sideforce and lift force in wind axes
$\mathbf{D}$	=	damage scale matrix
d	=	model dimension
$\bar{c}$	=	mean aerodynamic chord, m
$\mathbf{I}_n$	=	identity matrix of size n
I_{xx}, I_{yy}, I_{zz}	=	moments of inertia about $X/Y/Z$ axes of the body reference frame, kg $\cdot$ m^2
I_{xz}	=	product of inertia with respect to the X and Z axes of the body reference frame, kg $\cdot$ m^2
N	=	number of primitive operations
n	=	grid resolution of each dimension
p, q, r	=	angular rates (roll, pitch, and yaw) along $X/Y/Z$ axes of the body reference frame, rad/s
S	=	wing area, s^2
V	=	airspeed, m/s
α	=	angle of attack, rad
β	=	angle of sideslip, rad
$\delta_a, \delta_e, \delta_r$	=	control surface deflections of aileron, elevator, and rudder, rad
ϕ, θ, ψ	=	Euler angles from the earth reference frame to the body reference frame along $X/Y/Z$ axes, rad
φ	=	implicit value function of reachable sets
ρ	=	air density, kg/m^3

*PhD Student, Control and Simulation Section, Faculty of Aerospace Engineering, Delft University of Technology; Kluyverweg 1, 2629HS, Delft, The Netherlands. AIAA Student Member
†Assistant Professor, Control and Simulation Section, Faculty of Aerospace Engineering, Delft University of Technology; Kluyverweg 1, 2629HS, Delft, The Netherlands. AIAA Member
‡Associate Professor, Control and Simulation Section, Faculty of Aerospace Engineering, Delft University of Technology; Kluyverweg 1, 2629HS, Delft, The Netherlands. AIAA Member

I. Introduction

Safety is of paramount importance to aviation. Aircraft loss-of-control (LOC) is the dominant cause for a large fraction of aircraft accidents that have happened in the past few decades. Active research can be found in the field of LOC prevention and recovery through safe flight envelope prediction and protection [1].

Currently, the flight envelope protection systems are designed based on a predefined flight envelope. This flight envelope is an implicit set of data resulting from performance and structural limits of the aircraft, which is not explicitly referred to by the protection system in real time, but is incorporated as the basis of the control laws and system designs. Under abnormal conditions, the stability margins and control authority may suddenly reduce due to abrupt system failures and structural damage, which lead to a changed flight envelope. Current envelope protection is performed with respect to different values of relevant states/conditions, resulting from the combination of varies limits, which can vary also based on selected control modes selected by pilots after failure detection.

A database-driven system is proposed in our research [2], which is referred to as the "Database-driven safE Flight ENvelope preDiction system (DEFEND)." The general idea of the DEFEND system is to retrieve the changed flight envelope after failure/damage from an onboard database given current conditions of the aircraft. The database is designed to contain different damage and fault scenarios as well as flight conditions at which the flight envelopes are computed offline. In the proposed novel system, the flight envelope used for envelope protection is an explicit set of data retrieved from a database, and is separate from the baseline controller. By explicitly defining the flight envelope and integrating it into envelope protection as an independent module, the proposed approach can be more flexible than current methods under abnormal conditions. Furthermore, potential exists for partial protection effectiveness, in particular with degraded aircraft and control system status. More importantly, with modular design of flight envelope protection system, only envelopes that are explicitly referred to by the controller need to be computed, and the re-design of the overall protection system is not necessarily needed. The design of database allows for a wider range of abnormal cases and saves a large amount of online computation time.

This paper addresses offline construction of the database. Meanwhile, online interpolation is always needed when the required safe flight envelopes can not be directly retrieved from a sparse database in order to achieve the requirement of flight safety. In this paper, the reference aircraft is the Cessna Citation II, which is a twin-jet business aircraft. The Delft University of Technology operates an modified Cessna Citation II aircraft as flying laboratory, and its model is used in high-fidelity flight simulations. A drawing of the Cessna Citation aircraft as well as its damage configuration is shown in Fig. 1. The computation of envelopes is based on model parameters and configurations of this type of aircraft. The detailed mass properties and configurations can be found in [3].

The main focus and contribution of this paper is database building and interpolation for the DEFEND system. More importantly, through complexity analysis, the database approach is shown to be feasible for onboard real time implementation within the aircraft control laws.

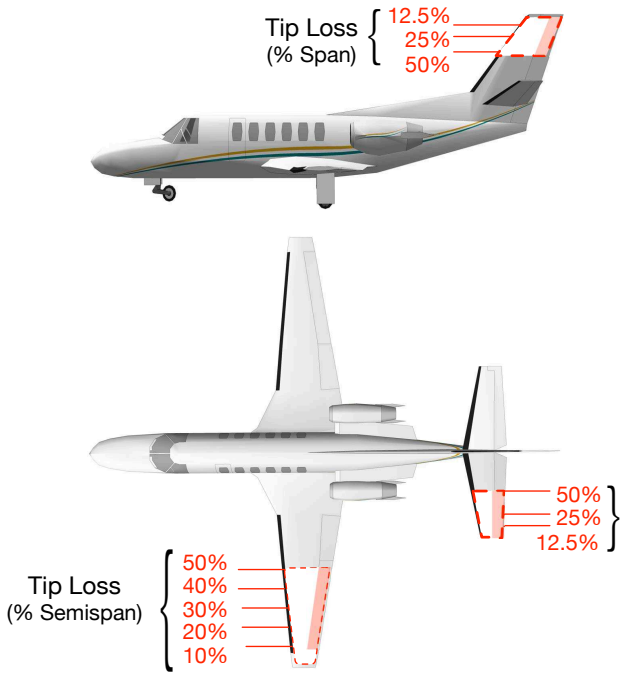


Fig. 1

II. Safe Flight Envelope Computation

Reachability analysis [4] is chosen in this paper as the method to compute safe flight envelopes, since the theory provides a set-valued insight into safety and control design of dynamic systems. One advantage of this method is that all possible trajectories can be computed from all available control strategies and initial states, which naturally meets the safety guarantees [5]. The computed results are called reachable sets, which are defined as a set of states that reach a certain target set within a certian time horizon and current control authority [4]. During the process of predicting the safe flight envelope, two reachable sets are needed, which are normally referred to as the backward reachable set and the forward reachable set [6, 7]. As illustrated in Fig. 2, the intersection between these two reachable sets of a given trim set is defined as the safe flight envelope, which indicates the region in the state space where aircraft can reach the trim set and maneuver freely within a certain time horizon. The trim set is estimated by minimizing the cost function of model states [7]. When failures or damage occur, both forward and backward reachable sets will shrink as well as the trim set due to the changed aircraft model. Therefore, some state trajectories that are part of the reachable sets during normal flight become unreachable after failures or damage, which results in the reduced safe flight envelope.

The boundary of reachable sets can be determined by solving for a value function and obtaining its zero level set [4]. In this paper we use the level set method [8, 9], which has been successfully applied to the reachability analysis of many systems including aircraft [7]. The level set method provides a numerical way of approximating the solution to the Hamilton-Jacobi-Isaacs (HJI) partial differential equation (PDE) formulated by the reachability

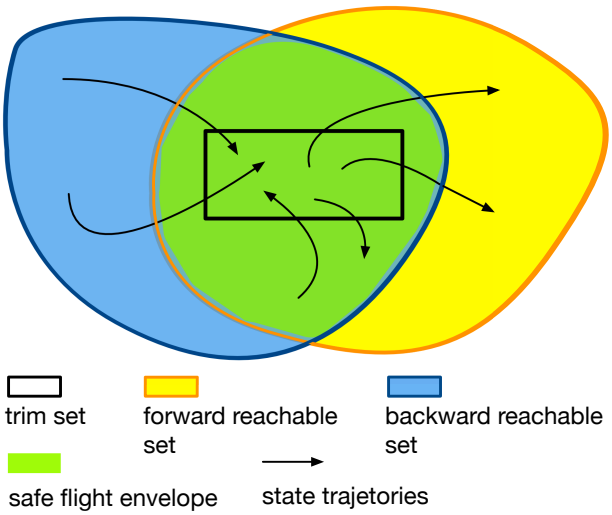


Fig. 2 Illustration of the safe flight envelope based on reachability analysis.

analysis. As a subclass of the Euler method, the level set method discretizes the state space into grids and calculates in a dimension-by-dimension manner, which evolves solving for optimal inputs on each grid node. A list of key steps of the level set method can be found in the appendix, and more technical details are elaborated in literature [8, 9] as well as online tutorials. The nonlinear model of a Cessna Citation is used in this paper to generate safe flight envelopes for the database.

The computational load of calculating reachable sets using the level set method becomes higher with the increase of model complexity and dimensions. So it is necessary to decompose the dynamic system to alleviate the curse of dimensionality without sacrificing optimality [10]. In this paper, the full model of the aircraft is decoupled into equations of longitudinal and lateral/directional dynamics [4, 7, 11]. Three-dimensional safe flight envelope are computed referred to each of the three planes of the aircraft dynamics. Different combinations of states lead to different shapes of safe flight envelopes, which can be stored and selected by pilots. In order to study the influence of aerodynamic model changes on the computed flight envelopes, this paper focuses on aircraft structural damage, which is defined as the departure, or loss of a part of the airframe (e.g., wing, vertical tail, horizontal stabilizers) and control surfaces. The damage results in asymmetric configurations as well as reduced areas of surfaces, which have an intrinsic impact on the aerodynamic and control characteristics [12, 13]. In this paper, structural damage is quantified by the proportion of lost tip span [12, 13], and the effects of damage can be quantified by the changed values of dimensionless aerodynamic coefficients and model structure [14].

In this paper, two sets of envelopes are computed using the level set method toolbox [9]. One is the *pitching envelope*, referred to the motion in the longitudinal plane with horizontal tail damage [12]. The other is the *rolling envelope*, referred to the motion in the lateral plane with wing tip loss [12]. It will be shown in the following examples that the computed envelopes under damage cases will shrink from their original damage-free shape. Starting from

these examples, we can generate a complete offline database containing different abnormal scenarios.

A. Longitudinal Envelopes

Under the condition of trimmed yaw and roll motion with states (p, r, ϕ, ψ, β) and their derivatives to be zero, the dynamics equation of the longitudinal motion can be written as [15]:

$$\begin{aligned}
\dot{V} &= \frac{1}{m}\left[T\cos\alpha - \frac{1}{2}\rho V^2 S C_D - mg\sin(\theta-\alpha)\right] \\
\dot{\alpha} &= -\frac{1}{mV}\left[T\sin\alpha + \frac{1}{2}\rho V^2 S C_L - mg\cos(\theta-\alpha)\right] + q \\
\dot{\theta} &= q\cos\phi \\
\dot{q} &= \frac{1}{2}\rho V^2 S c \frac{C_m}{I_{yy}}
\end{aligned} \tag{1}$$

By assuming constant velocity, a three-dimensional pitch envelope of (α, θ, q) can be computed based on the simplified model. Figure 3(a) and (b) show the safe flight envelopes computed from Eq. 1 under the condition of true airspeed $V_{TAS} = 100m/s$ and load factor $n = 1$ with time horizon T = 1s. Air density ρ is assumed to be constant ρ_0 at sea level. The control inputs are thrust and elevator deflections, which are within the bounds: $T \in [4448.2, 22241]$ N, $\delta_e \in [-25, 25]$ deg. Figure 3(a) shows both forward and backward reachable sets, which form the safe flight envelope displayed in Fig. 3(b) by taking the intersection between them. When one of the horizontal tails is damaged, the dominant effect is on the longitudinal stability. If the integrity of the attached elevators are also affected, the pitch control power, i.e., the control effectiveness may reduce, and the total pitching moment coefficient after damage is modeled as:

$$C_{m_{dmg}} = (\mathbf{I}_n - \mathbf{D}_m)\,\mathbf{C}_m \tag{2}$$

where $\mathbf{D}_m \in \Re^{n\times n}$ is a diagonal matrix, referred to as the damage scale matrix and $\mathbf{C}_m \in \Re^{n\times 1}$ is composed of polynomial terms of the dimensionless pitching moment model. The changed values of aerodynamic coefficients under a pre-defined damage case are determined based on data from a series of wind tunnel tests [12, 13] as well as computational simulations [3]. The tests are conducted on sub-scale aircraft models with partial or total loss of wings, horizontal stabilizers, and vertical tail. The values of stability derivatives under different damage cases are recorded during the tests. By analyzing these test data, the damage scale matrix $\mathbf{D}$ can be determined, which quantifies and models the effects of damage on aerodynamic characteristics. More details on the modeling work can be found in our previous paper [2]. Furthermore, the damage scale matrix can be used to represent different damage/failure combinations, without direct correspondence with a predefined failure.

According to the Cessna Citation model used in [2], $\mathbf{C}_m = [C_{m_0},\ C_{m_\alpha}\alpha,\ C_{m_q}\frac{q\bar{c}}{V},\ C_{m_{\delta_e}}\delta_e]$, and the damage scale matrix $\mathbf{D}_m = diag(d_{m_1},\ d_{m_2},\ d_{m_3},\ d_{m_4})$ can be identified from experimental data. Figure 3(c) and (d) show how the

envelopes reduce with 30% tip loss of one horizontal stabilizer ($\mathbf{D}_m = diag(0.3,\ 0.3,\ 0.3,\ 0.2)$).

The decoupled longitudinal equation is valid under the condition of trimmed yaw and roll motion. In case of asymmetric stabilizer and elevator damage, however, an incremental rolling moment is generated. Therefore, it is assumed that rudder and ailerons are capable of re-trimming the aircraft. The continuous re-trimming process is assumed to be automatically accomplished by the onboard fault tolerant controller so that a large amount of time spent on manual control can be saved.

The 2D contours in Fig. 4 are "slices" taken from 3D figures in Fig. 3 by setting the third state as fixed value. The main purpose of showing these 2D figures is to clearly demonstrate how each state changes after damage, and the values of envelope boundaries. Among all the values within the range of each state, some values result in empty contours, which indicate conditions that are not part of the whole envelope. The values of non-empty contours are selected as conditions in this paper for clarity of presentation.

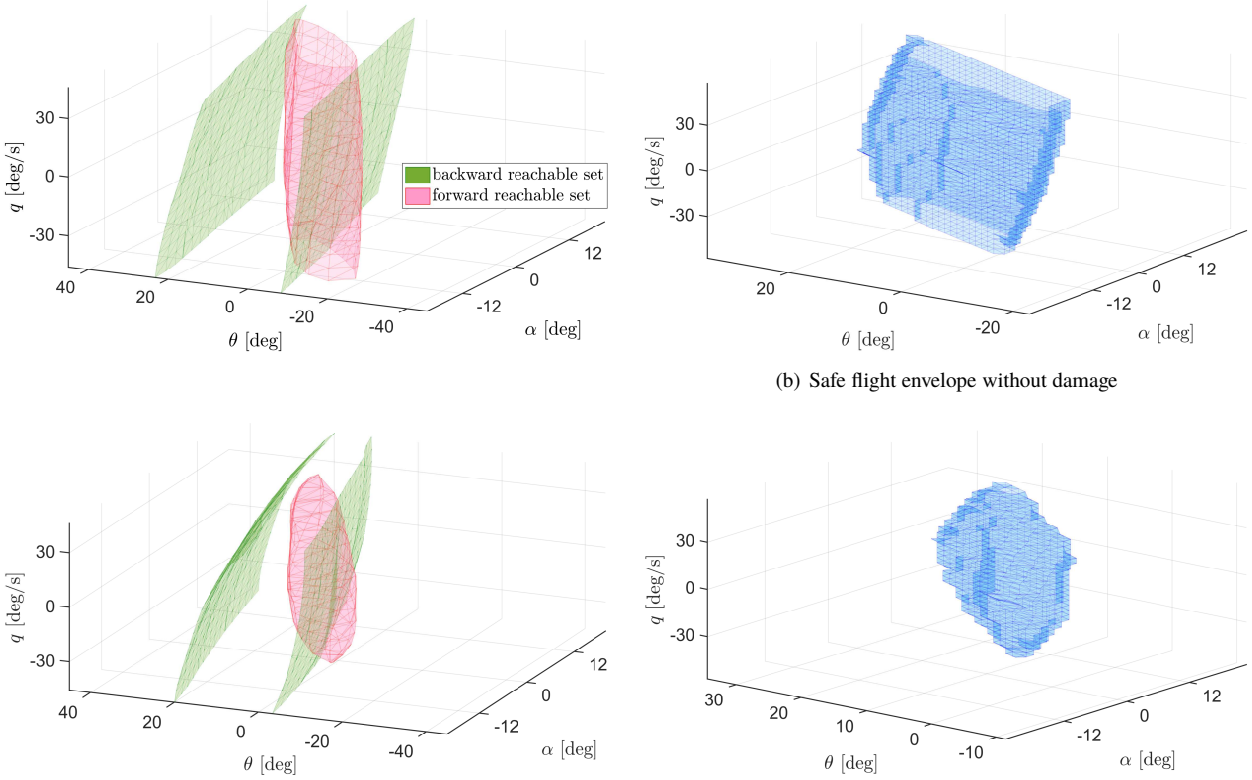
Some envelopes are not completely represented is because that the computed envelope boundaries have exceeded the computation range, which indicates the limits of flight states. Envelope beyond this computation range still exists, but it has no physical meaning in protecting the aircraft. Therefore, the computation range can be regarded as a larger flight envelope that are defined by the physical limits of the aircraft, and the incomplete envelope shown in this paper is the intersection between this larger envelope and the computation result.

B. Lateral Envelopes

The lateral/directional motion of aircraft with zero pitch rate $q = 0$ is defined as [15, 16]:

$$\begin{aligned}
\dot{\beta} &= p\sin\alpha - r\cos\alpha + \frac{1}{mV}\left(-T\cos\alpha\sin\beta + \frac{1}{2}\rho V^2 S C_Y + mg\sin\phi\right) \\
\dot{r} &= \frac{1}{2(J_{xx}J_{zz} - J_{zx}^2)}\rho V^2 S b(J_{zx}C_l + J_{xx}C_n) \\
\dot{p} &= \frac{1}{2(J_{xx}J_{zz} - J_{zx}^2)}\rho V^2 S b(J_{zz}C_l + J_{zx}C_n) \\
\dot{\phi} &= p + r\tan\theta\cos\phi
\end{aligned} \tag{3}$$

The rolling envelopes are mostly influenced by the integrity of wings and the attached ailerons. To compute a rolling envelope with states (β, p, ϕ) on the assumption of zero yaw rate ($\dot{r} = r = 0$), the rudder is assumed to function as normal within the bound of $\delta_r \in [-22, 22]$ deg, in order to maintain a steady heading sideslip maneuver. The deflections of aileron are set within the bounds: $\delta_a \in [-33, 33]$ deg. The two reachable sets and the resulting safe flight envelope in normal condition are illustrated in Fig. 5(a) and (b) respectively. If one side of the wing is damaged, unequal lift forces are induced, generating an incremental rolling moment ΔC_l that increases with α in the angle of attack range in which the aerodynamic characteristics are linear. The corresponding rolling moment increases with α,

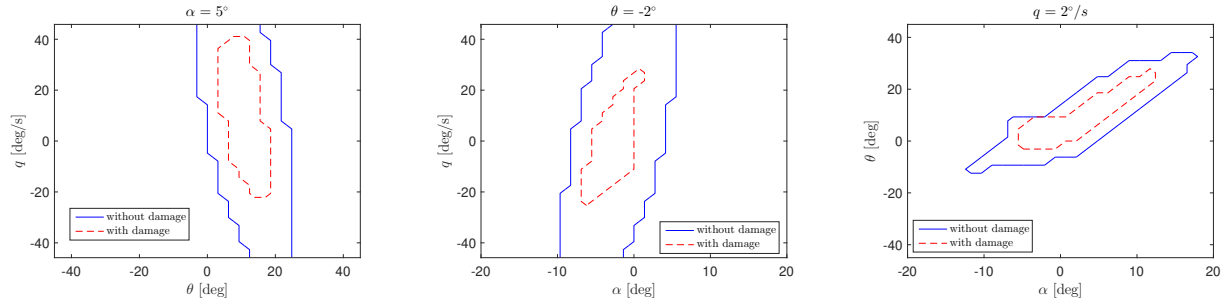


(b) Safe flight envelope without damage

(c) Forward and backward reachable sets with damage

(d) Safe flight envelope with damage

Fig. 3 Comparison of (α, θ, q) envelopes under 30% horizontal stabilizer damage



(a) (b) (c)

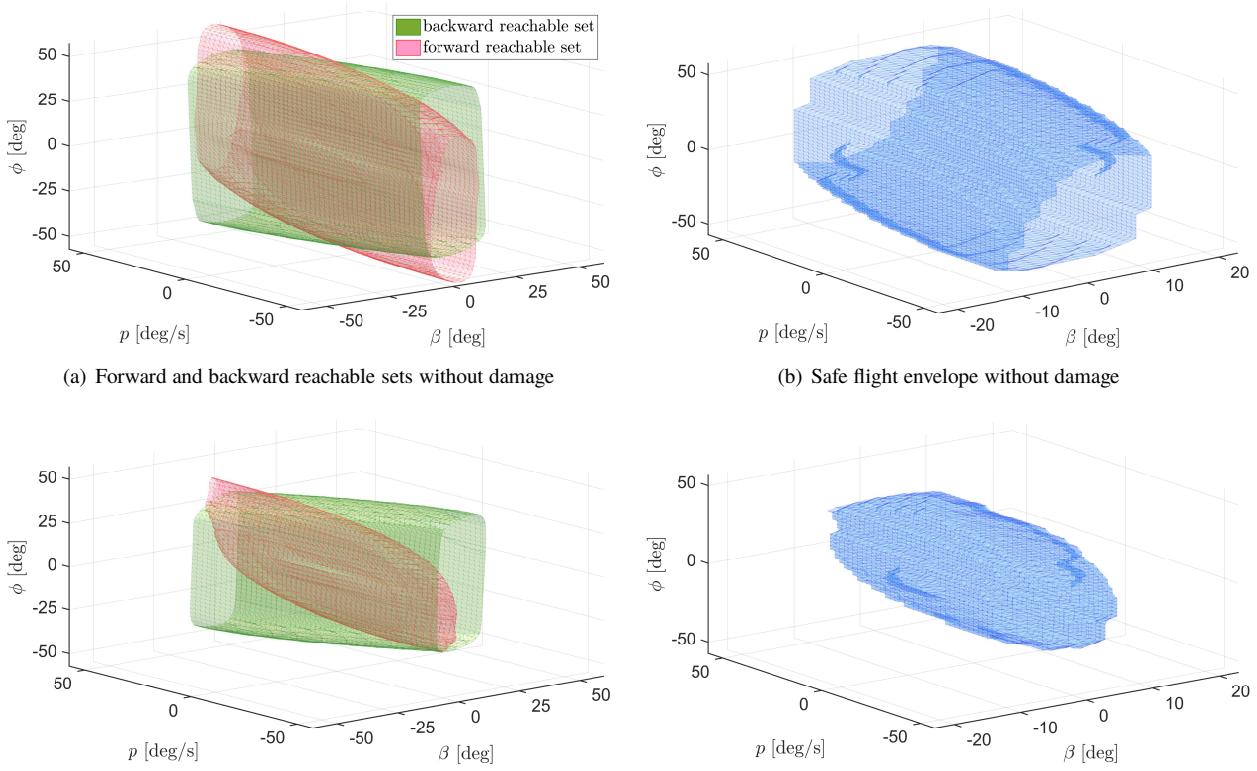
Fig. 4 2D-projected (α, θ, q) envelopes with and without 30% horizontal stabilizer damage

assuming constant dynamic pressure.

Besides, available roll control power is reduced due to structural damage of ailerons as well as the fact that part of the aileron control is used to compensate for roll asymmetry. Therefore, the non-dimensional coefficient of rolling moment $C_{l_{dmg}}$ after damage is composed of two parts: one is formed by the original vector of lateral aerodynamic terms $\mathbf{C}_l$ scaled by the rolling damage scale matrix $\mathbf{D}_l$; and the other one is the incremental term, which is:

$$C_{l_{dmg}} = (\mathbf{I}_n - \mathbf{D}_l)\,\mathbf{C}_l + \Delta C_{l_\alpha}\,\alpha \tag{4}$$

where $\Delta C_{l_\alpha}\alpha$ denotes the incremental rolling moment non dimensional coefficient due to asymmetric wing damage. According to the Cessna Citation model used in [2], the vector of the lateral aerodynamic terms is $\mathbf{C}_l = \left[C_{l_0}, C_{l_\beta}\beta, C_{l_p}\frac{p\bar{c}}{V}, C_{l_r}\frac{r\bar{c}}{V}, C_{l_{\delta_a}}\delta_a, C_{l_{\delta_r}}\delta_r\right]$. Under 20% wing tip loss, the diagonal damage scale matrix for rolling moment is $\mathbf{D}_l = diag(0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.4,\ 0)$ and the incremental coefficient $\Delta C_{l_\alpha} = 0.02\,\text{rad}^{-1}$, which are derived from wind tunnel tests [12].



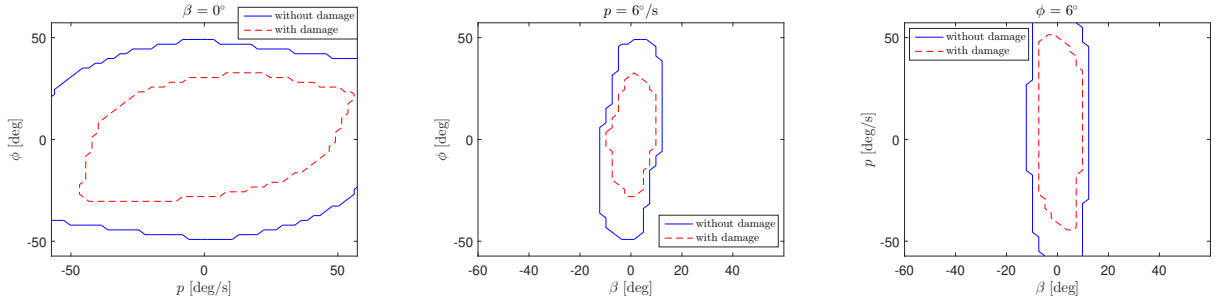
(a) Forward and backward reachable sets without damage

(b) Safe flight envelope without damage

(c) Forward and backward reachable sets with damage

(d) Safe flight envelope with damage

Fig. 5 Comparison of (β, p, ϕ) envelopes with and without 20% wing damage



(a) (b) (c)

Fig. 6 2D projected (β, p, ϕ) envelopes with and without 20% wing damage

III. Information Retrieval and Interpolation

A. Database Design and Structure

Based on the computed safe flight envelopes, a database containing offline calculated envelopes under different fault and damage scenarios can be designed and established. Table 1 shows the main attributes that should be included in the database, which shows how much information needs to be specified in order to retrieve an envelope. It should be noted that for each attribute, only a few example values are listed in Table 1 as an indication. The first three columns in Table 1 describe the damage cases defined by the damage modeling experiments, and the corresponding aircraft models that are used to compute the safe flight envelopes. The following four columns indicate the time span and flight states under which the retrieved flight envelope is valid.

An unique flight envelope $E_{(e_1,e_2,...,e_r)}$ in the last column is determined by attributes in the database, where $(e_1, e_2, ..., e_r)$ denotes the variables of attributes in the database. For example, e_1, e_2, e_3 describes the current fault and damage by its location, type and severity. Based on this, several flight envelopes can be retrieved under different flight conditions (e.g. time horizon, pressure altitude) defined by $e_4, e_5, ..., e_r$, among which only one envelope will be retrieved if values of these attributes can be determined.

Table 1 Preliminary Design of Safe Flight Envelope Database

Location	Type	Severity	Time [s]	Model	True Airspeed [m/s]	Pressure Altitude [m]	Envelope
right horizontal stabilizer	tip loss	30%	1	(α, q, θ)	80	sea level	$E_{(e_1,...e_r)}$
	tip loss	50%	2	(α, q, θ)	90	5000	$E_{(e_1,...e_r)}$
right elevator	stuck	15°	1	(α, q, θ)	100	10000	$E_{(e_1,...e_r)}$
	loss of effectiveness	30%	2	(α, q, θ)	110	5000	$E_{(e_1,...e_r)}$
left wing	tip loss	40%	1	(β, p, ϕ)	120	sea level	$E_{(e_1,...e_r)}$
	tip loss	60%	2	(β, p, ϕ)	130	5000	$E_{(e_1,...e_r)}$
vertical tail	tip loss	20%	1	(β, r, ψ)	90	10000	$E_{(e_1,...e_r)}$
	tip loss	40%	2	(β, r, ψ)	100	5000	$E_{(e_1,...e_r)}$

As introduced in Sec. II, the safe flight envelope can be considered as a set-valued result of an optimal-control based reachability problem. The resulting reachable set is represented as the zero-level set of an implicit surface function φ, i.e., a set of points where $\varphi = 0$. Without an explicit analytic expression, the surface function is approximated by values in grid points, which are propagated from an initial set in a certain velocity field. Therefore, the envelope can be regarded as a closed lower-dimensional interface that divides the state space into interior and exterior regions [8]. For a certain point $P(i, j)$, the sign of its value function $\varphi_{i,j}$ determines which side of the interface the point is on. That is, P is inside the interfaces when $\varphi_{i,j} < 0$ and outside when $\varphi_{i,j} > 0$. If $P(i, j)$ and its neighboring grid points satisfy

[17]:

$$\max(\varphi_{i,j}, \varphi_{i+1,j}, \varphi_{i,j+1}, \varphi_{i+1,j+1}) < 0 \quad \text{OR} \quad \min(\varphi_{i,j}, \varphi_{i+1,j}, \varphi_{i,j+1}, \varphi_{i+1,j+1}) > 0 \tag{5}$$

it means that $P(i, j)$ is guaranteed to be inside or outside the interface. Such points are ignored and the rest of the grid points are regarded to lie on or in the vicinity of the interface. In this way, a narrow band around the interface is formed by the points that approximate the boundary.

Envelopes stored as grid points can be readily used for further calculation, but may suffer from bad scaling with the dimension. Therefore, it is also important to design the number of data points for each attribute so that the database can provide as much information as possible while keeping a reasonable data volume for onboard devices. Furthermore, other storage formats of the flight envelopes, function approximators for instance, might be needed to compress the data volume stored in the database. Research on this topic will be included in the future work.

B. Safe Flight Envelope Interpolation

In real applications, safety should be included as the primary consideration. Since the finite number of safe flight envelopes in the database is in accordance with the number of abnormal cases and discrete flight states, there are inevitably situations where the true flight envelope falls in between the two neighboring categories in the database. In this case, it is necessary to interpolate between two envelopes to fill in the gaps in Table. 1 and obtain more accurate results [2].

Interpolation that is commonly used in look-up tables mostly works with two points. However, in this application, interpolation needs to be done between two envelopes based on certain connections. It can be observed that the safe flight envelopes of different damage severity or flight conditions often share similar geometrical shapes with different scales of expanding, shrinking or shifting. Since flight envelopes can be considered as contours and surfaces, the method used for interpolation in this paper is inspired by research on surface reconstruction and image matching using the level set method and the fast marching method [18–20]. The basic idea of these methods is to build up the geometrical features of contours and surfaces and track the propagation of their fronts made up of data points.

For two envelopes composed of two sets of data points, the mapping is implemented pair-wisely through the optimal path between each two points calculated by the fast marching method. For simplicity and a proof of concept, only two-dimensional envelopes are investigated in this paper, since a three-dimensional envelope can be projected into a $2D$ plane. The extension to higher-dimensional hyper-surfaces will be included in future work.

Generally, given two points in $\mathfrak{R}^n$, the path in between can be defined by a vector-valued function $\gamma(\tau) : [0, \infty) \rightarrow \mathfrak{R}^n$, i.e., each dimension of $\gamma(\tau)$ is defined by a separate function parameterized by τ. If τ is the arclength parameterization of γ, then it has the property of $||\frac{d\gamma(\tau)}{d\tau}||_2 = 1$.

In two dimensions, given a cost function $F(x, y)$, one goal in optimal path planning is to construct the path between

(x_0, y_0) and (x, y) that minimizes the cost of travel between these two points. Let $T(x, y)$ be the minimal cost, which is:

$$T(x, y) = \min_{\gamma} \int_{(x_0,y_0)=\gamma(0)}^{(x,y)=\gamma(L)} F(\gamma(\tau))d\tau \tag{6}$$

where L is the total arclength of path. The level set $T(x, y) = c$ is the set of all points that can be reached from (x_0, y_0) with minimal cost c, and the minimal cost paths are orthogonal to the level curves. Hence,

$$||\nabla T||_2 = F(x, y) \cdot |\frac{d\gamma(\tau)}{d\tau}||_2 = F(x, y) \tag{7}$$

which forms an Eikonal equation [21]. The equation can be interpreted in another way. Imaging a wave front expanding from the starting point (x_0, y_0) with unit speed $F = 1$ until it reaches the end point (x, y). Then the solution $T(x, y)$ is the distance to the starting point, and the level curve of $T(x, y) = c$ denotes the wave front that is located a distance c away. The gradient $||\nabla T||_2$ must be orthogonal to these level curves, implying the path followed by the wave is the shortest path in time. If the end point is not specified and we let the wave propagate through the entire grid, then each grid point should have a value of optimal distance from the starting point, and these values form a distance map of the grid.

The fast marching method provides a numerical scheme for computing solutions to the Eikonal equation based on entropy-satisfying upwind schemes and fast sorting techniques [21]. Readers can refer to Refs. [19, 21] for more details. In this paper, the single starting point A is extended to a set of starting points located on the boundary of one retrieved envelope E_0. For each grid point (x, y), $T(x, y)$ is computed, which denotes the shortest travel distance from point (x, y) to its nearest starting point. The values of $T(x, y)$ thus formulate a distance map, which is stored and retrieved together with its envelope E_0 correspondingly. The size of distance should be designed to cover the largest envelope in the database so that for any envelope E_1, the shortest path can be constructed for points on the boundary through tracing along the smallest value of $T(x, y)$ until one of the points on E_0 is reached.

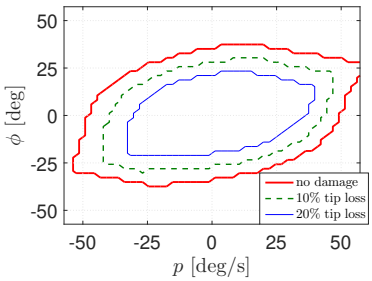
Once optimal path planing is repeated for each point on E_1, linear interpolation can be implemented on each optimal path between data points of the two envelopes. It is assumed that points forming the interpolated contour also lie on the optimal path, and their specific locations along the path are determined by the interpolation weights. These weights are heuristically chosen based on different conditions in which the envelopes are computed.

In Fig. 7, the lateral envelopes under wing damage are used as the first example for demonstration. Figure. 7(a) shows three 2D envelopes in (p, ϕ) computed by the level set method. The largest contour is computed without any damage and the smallest one is computed with 20% wing tip loss. The contour in between them corresponds to the envelope of 10% tip loss, which is used and displayed to test the performance of the interpolation with the method proposed in this section. The retrieved envelopes of zero damage and 20% damage are approximated by data points

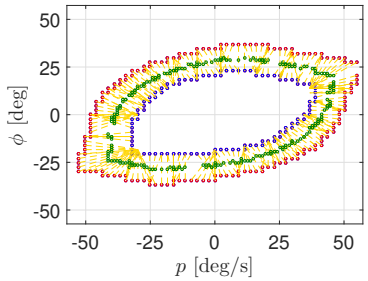
(fig. 7(b)) based on Eq. 5. Then, optimal paths between the points on the largest and the smallest contour are computed, which are represented in yellow dashed lines in fig. 7(b). After the mapping is built between two envelopes, linear interpolation is performed to find the points on the optimal path that approximate the intermediate envelope, which is shown in 7(b). It can be observed in fig. 7(c) that the interpolated contour accurately approximates the envelope of 10% wing damage computed by the level set method.

Damage level is not the only decisive factor in the database. Some other attributes, like true airspeed, pressure altitude and time horizon are also important in building the database. As shown in Table. 1, the value of these attributes are also discrete, which means that interpolation is necessary when specific values of certain states are required. The second example demonstrated in Fig. 8 deals with envelopes of different velocities. Figure. 8(a) shows three 2D envelopes in (β, r) computed by the level set method. The inner and outer envelopes correspond to $V_{TAS} = 60m/s$ and $V_{TAS} = 100m/s$ respectively. The task is to find out the contour of $V_{TAS} = 80m/s$ by interpolating between the two known contours. Following the similar steps of the fast marching method used in the first example, the intermediate contour is found, which approximates the contour computed by the level set method with satisfying accuracy, as shown in Fig. 8(c).

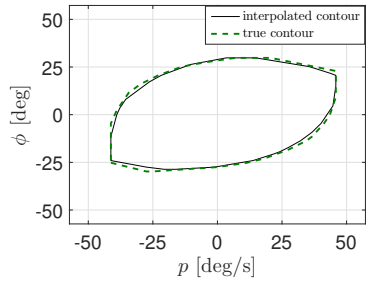
The performance of interpolation can be influenced by the geometric characteristics of the envelopes/contours. It is noticed in the figures that each established path starts from one point on the outer contour and traces back to one of the inner contour along the distance map. Since the distance map is computed based on the inner contour, each "wave front" that propagates from each point on the inner contour may reach more than one point on the outer contour. In this way, not all of the inner points are connected to the outer contour since multiple paths may end up in the same inner point. This may result in some areas with concentrated paths and others with sparse paths. Such inconsistency gives rise to the accumulation of error, especially around the corners of the shape. It is also important to note that there is no direct relation between the computation of optimal path γ and the dynamics models of the aircraft, since the establishment of F and T of the fast marching method treats flight envelopes as ordinary geometric shapes.



(a) Two-dimensional (p, ϕ) envelopes computed by the level set method

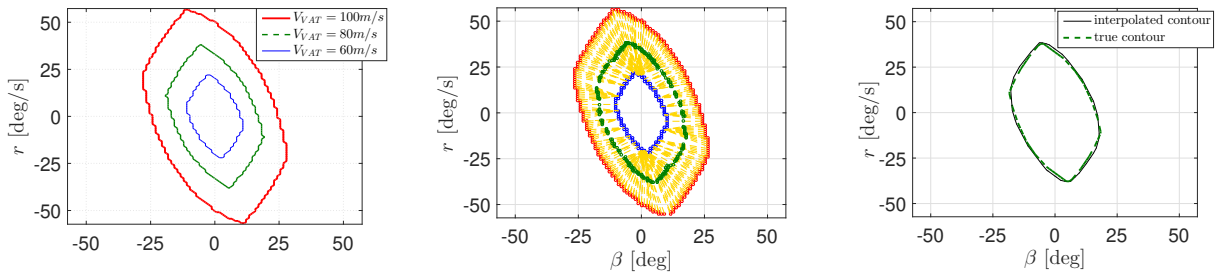


(b) Computed optimal paths and the resulting interpolated contour



(c) Comparison between the interpolated contour and the true contour of 10% wing damage

Fig. 7 Interpolation between envelopes of different damage levels based on the fast marching method



(a) Two-dimensional (β, r) envelopes computed by the level set method

(b) Computed optimal paths and the resulting interpolated contour

(c) Comparison between the interpolated contour and the true contour of $V = 80m/s$

Fig. 8 Interpolation between envelopes of different velocities based on the fast marching method

As mentioned earlier, the interpolation of flight envelopes in our research is initiated by a safety-related problem [2]. Take structural damage for example, it is desired to have more damage levels so that more accurate safe flight envelopes can be predicted. On the other hand, the number of designed damage levels is restricted by physical limitations of modeling experiments like wind-tunnel tests and simulations. Additionally, as indicated in Sec. III A, volume limitations on the database also exist if it needs to be carried onboard. Therefore, in order to obtain a better design of the database, it is important to determine the number of envelopes needed for each damage location by evaluating the performance of the interpolation under different database configurations.

Given two envelopes, one is interpolated from two neighboring envelopes retrieved from the database, and the other envelope for comparison is directly computed from the level set method. The interpolation error is calculated point-wise between the interpolated envelope and the compared envelope. For any point (x_i, y_i) on a 2D interpolated envelope, the error between (x_i, y_i) and any point (x_j, y_j) on the compared envelope is calculated by:

$$err(i,j) = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sqrt{x_j^2 + y_j^2}} \tag{8}$$

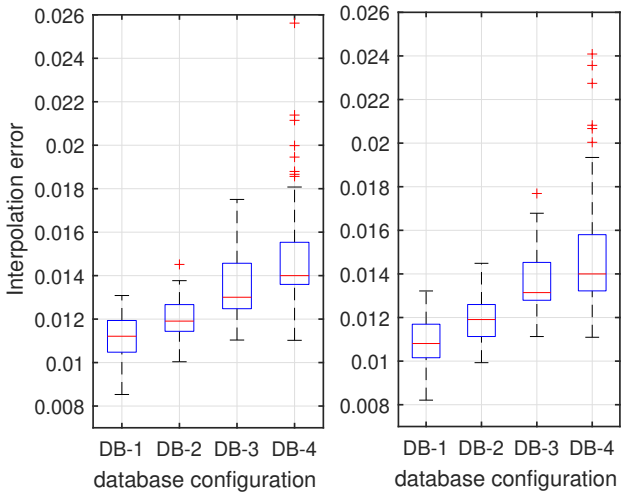
Among all the $err(i,j)$, the point that gives the smallest $err(i,j)$ is regarded as the nearest point on the compared envelope, and this smallest error is defined as the error of point (x_i, y_i) on the interpolated envelope, which is: $err(i) = \min_j\{err(i,j)\}$. The calculated errors for each point on the interpolated envelope forms a data set $\{err(i)\}$ that represents its interpolation error set. In this paper, the interpolation errors of certain damage cases are evaluated on four database configurations.

In this paper, the interpolation errors are evaluated on four database configurations based on the number of damage cases, as shown in Fig. 9(a). Between the damage scale of 0% and 60% tip loss, the number of stored envelopes can be reduced by enlarging the interval between two stored envelopes. In Fig. 9(a), four databases (denoted as DB-1 to DB-4 in the figure) are designed to store different numbers of envelopes. The notations E0 to E6 in the figure represent

envelopes of different damage scales ranging from no damage (0%) to 60% tip loss for each damage location.

Based on these four configurations, the envelope interpolation errors of 25% and 45% wing tip loss calculated by Eq. 8 are shown in the box chart in Fig. 9(b). Significant increase of variance and outliers (denoted by cross mark) is observed in database 4 with the maximum scale interval. The interpolations on DB-1 and DB-2 show similar results in both cases, which indicates that the design with 4 damage levels (DB-2) is sufficient for obtaining accurate safe flight envelopes through interpolation. In this way, a lot of storage space can be saved, and the accuracy can be guaranteed at the same time.

(a) Four databases with different number of envelopes stored



(b) The interpolation error of 25% wing tip loss (left) and 45% wing tip loss (right) evaluated on different database configurations

Fig. 9 Database design of envelope numbers based on interpolation error

C. Complexity Analysis

In order to show the feasibility of online application of the database approach, it is necessary to compare its computational complexity with that of the level set method. *Computational complexity theory* is concerned with how much computational resources in terms of time and space are required to solve a given task [22]. In this paper, the complexity analysis is focused on the time complexity, or computational efficiency of the algorithm. The efficiency of an algorithm is quantified by the number of basic operations as a function of input size [22]. Basic operations, also called primitive operations, correspond to low-level instructions (e.g., addition, multiplication) with constant execution

time. Thus, the actual running time of an algorithm is expected to be proportional to the number of primitive operations [23].

For a given computational grid of d dimensions with equal resolution of n for each dimension, it takes at least $N(n,d) = K_{ls}(d)\cdot n^{d+1}$ primitive operations to compute a reachable set using the level set method, where $K_{ls}(d)$ denotes the number of primitive operations required for each grid point ("ls" represents "level set"). It should be noted that the increase of dimension d means that more complex system models might be used with an increased number of flight states as well as control inputs. If nonlinear programming is required to compute optimal control inputs, the complexity will grow significantly compared with the simple linear cases. Therefore, $K_{ls}(d)$ is an implicit, monotonically increasing function of dimension d, which may increase with more state equations of a higher-dimensional system model as well as with the complexity of the optimal controller. Detailed analysis of $N(n,d)$ of the level set method can be found in the Appendix.

If the envelope is retrieved from the database instead of computed, the time complexity is not dependent on n or d but only the structure of the database, which is trivial compared to the level set method. When interpolation between two retrieved envelopes is required, instead of computing to the entire grid like the level set method, only a subset of points on the envelope boundary are needed. We first consider the simplest two-dimensional case, in which the interpolation between two envelopes, E_1 and E_2, requires $N(n) = M(n) \cdot N_{in}(n)$ primitive operations. $M(n) = \lambda \cdot n$ denotes the number of points on the boundary of E_1, which is proportional to the input size n with a scale factor λ. For each point on the boundary of E_1, the computation of the optimal path between two contours based on the offline generated distance map of E_2 (as described in Sec.III B), requires $N_{in}(n) = K_{in} \cdot n$ primitive operations with $K_{in} < 1$ ("in" indicates "interpolation"). Therefore, the total number of primitive operation is $N(n) = K_{in} \cdot \lambda n^2$. Since a 3D envelope can be decomposed into slices of 2D envelopes, the total number of primitive operations can be approximated by:

$$N(n) = \sum_{i=1}^{n} K_{in}(i) \cdot \lambda(i)\, n^2 \leq n^3 \cdot \max_i K_{in}(i) \cdot \max_i \lambda(i) \tag{9}$$

where i is the index to the grid point of the third dimension. Based on this, the interpolation between higher dimensional envelopes approximately requires $N(n,d) = K_{in} \cdot \lambda\, n^d$ primitive operations by decomposing them into lower-dimensional envelopes, where $K_{in} \cdot \lambda$ is the average value over the whole grid.

For example, to obtain a yawing envelope of vertical tail damage with two states (β, r) (as shown in Fig. 8), the level set method takes approximately $3 \cdot 10^8$ ($K_{ls} = 307$) primitive operations on a 100×100 two-dimensional grid, whereas the database approach only needs 7695 ($K_{in} = 0.27,\ \lambda = 2.85$) primitive operations by interpolating between two envelopes retrieved from the database with the same grid resolution and dimension. Based on this aircraft model, Fig. 10(a) shows the number of primitive operations $N(n)$ of using interpolation and the level set method with respect to grid resolution ($n \in [10, 100]$, $d = 2$). The figure also shows the plot of n^2 and n^3 for comparison. It can be clearly

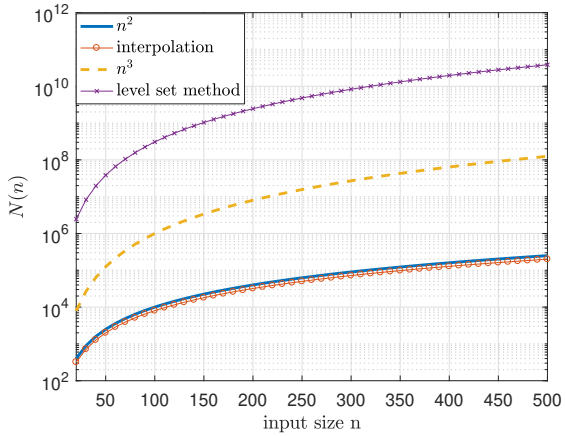
seen that the database approach using interpolation method requires much less computation load compared with the level set method. To make a more straightforward comparison of both methods in terms of dimension, the speedup ratio of the database approach is thus defined as:

$$S_{db}(n,d) = \frac{K_{ls}(d) \cdot n^{d+1}}{K_{in} \cdot \lambda\, n^d} = \frac{K_{ls}(d)}{K_{in} \cdot \lambda} \cdot n \tag{10}$$

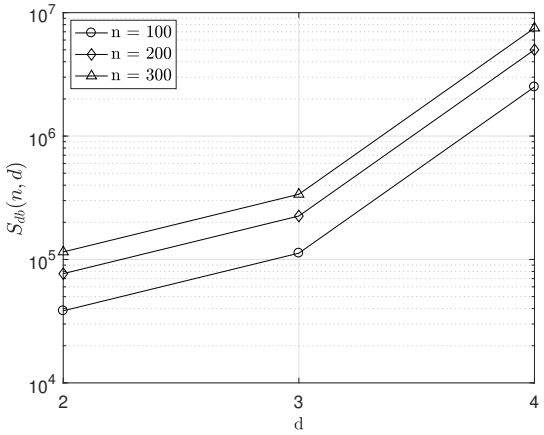
The influence of dimension is mostly on the computation of the level set method described by $K_{ls}(d)$. Based on different models of 2, 3, and 4 dimensions respectively, the results of the speedup ratio for different d and n are displayed in Fig. 10(b). It can be observed that the speedup ratio grows significantly when the dimension reaches 4, indicating that the level set method becomes significantly more complex compared with the database approach. This is largely due to the increased complexity of solving a higher-order optimal control problem, which results in the non-linearity of $K_{ls}(d)$. Despite the high speedup ratio with dimension 4, it indicates the results in relatively simple cases where nonlinear programming is not involved during the computation. Therefore, The results shown in Fig. 10(b) should be interpreted as a lower bound to the speedup ratio of the database method. For more complex case where nonlinear programming is needed, the complexity of the level set method will increase as well as the speedup ratio. The computation of envelopes of such models with dimensions higher than 4 is out of the scope of this paper, but we will look into it in future work.

A key advantage of the database approach is that the required computational load does not depend on the complexity of aircraft system model, which makes it a method that covers a wider spectrum of cases and models. It should be noted that the level set method is not the only way to compute reachable sets and flight envelopes, and the scope of this method is limited. For example, when models of over-actuated aircraft with highly-redundant control effectors are involved, the level set method is incapable of computing the flight envelopes since there is no analytic solutions to optimal control inputs. Instead, other theoretical and practical methods have been proposed and utilized for envelope computation of this kind of model. However, these methods also have high computational complexity that increase with the dimension [24], which can not be used online. Therefore, a database is always needed no matter what method is used offline to compute the safe flight envelopes.

This important feature also guarantees that when the aircraft model is changed due to faults and damage (within the range of the database), the running time of retrieving a different envelope remains unchanged, while the time taken to compute a new envelope using the level set method (if it is possible) increases significantly due to the changed model of the damaged aircraft.



(a) The increase of primitive operations with different n on a two-dimensional grid



(b) The increase of speedup ratio with different model dimensions

Fig. 10 Comparisons of the time complexity between the database interpolation approach and the level set method

IV. Conclusion

This paper discusses the implementation of database building and interpolation for an online database-driven safe flight envelope prediction system. Safe envelopes for both longitudinal and lateral motions are computed using the level set method, demonstrating obvious shrinkage with the occurrence of model changes due to structural damage. The database is constructed from envelopes for different flight conditions and abnormal cases. In order to address the potential safety problem brought by an insufficient number of envelopes stored in the database due to limitations of damage modeling experiments or data volume, a new fast-marching based interpolation scheme is presented for any two envelopes based on their geometrical mapping. By analyzing the interpolation errors for a different numbers of damage cases, the database configuration can be improved while at the same time reducing the required data storage volume.

Computational complexity for both the database approach and the level set method are analyzed with respect to model dimension and input size. The improvement in time complexity of the new database approach compared to the level set method is quantified with a speedup ratio. This speedup ratio is shown to increase polynomially with the grid resolution (n) and exponentially with the dimension (d). The comparison result for a simple two-dimensional example indicates a significant reduction in computational load for the database approach with a speedup ratio of 40000. For a more realistic problem with 4 states ($d = 4$), the speedup ratio for the database approach over the level set method is more than 10^6. This research also indicates that flight envelopes obtained by other experimental or computational methods can be stored in and retrieved from the database. Given its hight efficiency and versatility, the database approach is shown to be feasible for online safe flight envelope prediction. Future work will focus on the implementation and integration of the database into a flight envelope protection system.

Appendix: Complexity Analysis of the Level Set Method

The implementation of the level set method can be divided into five parts. Since it evolves solving an ODE, the calculation is repeated for every time step. We first need to count the number of primitive operations for each time step. For a given grid of d dimensions with the same resolution of n for each dimension, the number of primitive operation can be roughly calculated.

If K_1 denotes the number of primitive operations of calculating the spatial derivatives $p_i = \frac{\partial \varphi_i}{\partial x_i}$ for each dimension i [8], the total number of operations for the entire grid at each time step should be $K_1 dn^d$.

Calculating the analytic Hamilton-Jacobi (HJ) functions $H(x, p) = \min_u F(p, x, u)$ needs $K_2\, n^d$ primitive operations. The function to be minimized is defined as [4]:

$$F(p, x, u) = \sum_{i=1}^{d} p_i\, f_i(x, u) \tag{11}$$

where $f(x, u)$ is the aircraft system model and u is the control input. For each grid point with given values of x and p, K_2 largely depends on the optimization of $F(u)$ using optimal control input u, which is determined by $\frac{dF(u)}{du}$. For linear function where $\frac{dF(u)}{du}$ = constant, the control input is either maximum or minimum value (bang-bang solution in optimal control). For nonlinear function where $\frac{dF(u)}{du} = h(u)$, and $h(u)$ can be polynomial or trigonometric functions. In this case, nonlinear programming methods need to be taken to find the optimal value and the corresponding control inputs, which will significantly increase the computational load of the method.

The numerical approximation of HJ function with dissipation coefficients α using Lax-Friedrichs scheme is [8, 9]:

$$\hat{H}(x, p) = H(x, p) - \alpha(x) \tag{12}$$

The calculation of Lax-Friedrichs needs $(3 + K_3)\, n^d$ primitive operations, where $K_3 = \sum_{i=1}^{d} K_{3_i}$ denotes the total number of steps taken to evaluate the dissipation coefficients α at each grid point. For each dimension i, K_{3_i} steps are required to determine $\alpha_i = \max_u |\frac{\partial H}{\partial p_i}|$ [9], which also depends on the relation between system states and control inputs.

Calculating the step bound for CFL condition of integration takes in the value of the calculated dissipation coefficient in [8]:

$$\Delta t_{max} = \frac{1}{\max_x (\sum_{i=1}^{d} \frac{\alpha_i(x)}{\Delta x_i})} \tag{13}$$

Therefore, the determination of time step requires $3d\, n^d$ primitive operations in total.

For each time step, the above calculation of $\hat{H}(x, p)$ is repeated for each grid point as the right hand side of the ordinary differential equation [8]:

$$\frac{d\varphi(x)}{dt} = -\hat{H} \tag{14}$$

The number of operations required to solve this ODE is K_4, which depends on the integration scheme. Take Euler method for example:

$$\varphi(t + \Delta t) = \varphi(t) - \Delta \hat{H} \tag{15}$$

where $K_4 = 5$ for each grid point. As stated before, the length of time step Δt is bounded by CFL condition $\Delta t \leq \frac{\Delta x}{\max |V|}$, where V denotes the external velocity that drives the propagation of the implicit surfaces. Hence, the total number of time steps can be calculated as:

$$N_t = \frac{T}{\Delta t} \geq \frac{T \cdot \max(|V|)}{\min_i \Delta x_i} = \frac{T \cdot \frac{\sqrt{\Sigma_{i=1}^{d} X_i^2}}{T}}{\frac{\min_i X_i}{n}} = \frac{\sqrt{\Sigma_{i=1}^{d} X_i^2}}{\min_i X_i} n \geq n \tag{16}$$

where T denotes the time horizon and X is the range of each dimension. Following the discussion above, the total number of primitive operations required to implement the level set method is $N(n,d) = (d\,K_1 + K_2 + K_3 + K_4 + 3\,d)\,n^d\,N_t = K_{ls}(d)\,n^d\,N_t$. Since $N_t \geq n$, the lower bound of $N(n,d)$ is $K_{ls}(d)\,n^{d+1}$.

References

[1] Belcastro, C. M., Foster, J. V., Shah, G. H., Gregory, I. M., Cox, D. E., Crider, D. A., Groff, L., Newman, R. L., and Klyde, D. H., "Aircraft Loss of Control Problem Analysis and Research Toward a Holistic Solution," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 4, 2017, pp. 733–775. doi:10.2514/1.G002815, URL https://arc.aiaa.org/doi/10.2514/1.G002815.

[2] Zhang, Y., de Visser, C. C., and Chu, Q. P., "Aircraft Damage Identification and Classification for Database-Driven Online Flight-Envelope Prediction," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 2, 2018, pp. 449–460. doi:10.2514/1.G002866, URL https://arc.aiaa.org/doi/10.2514/1.G002866.

[3] Nabi, H. N., Lombaerts, T., Zhang, Y., van Kampen, E., Chu, Q. P., and de Visser, C. C., "Effects of Structural Failure on the Safe Flight Envelope of Aircraft," *Journal of Guidance, Control, and Dynamics*, 2018, pp. 1–19. doi:10.2514/1.G003184, URL https://arc.aiaa.org/doi/10.2514/1.G003184.

[4] Lygeros, J., "On Reachability and Minimum Cost Optimal Control," *Automatica*, Vol. 40, 2004, pp. 917–927. URL http://www.sciencedirect.com/science/article/pii/S0005109804000263.

[5] Kaynama, S., Mitchell, I. M., Oishi, M., and Dumont, G. A., "Scalable safety-preserving robust control synthesis for continuous-time linear systems," *IEEE Transactions on Automatic Control*, Vol. 60, No. 11, 2015, pp. 3065–3070. doi:10.1109/TAC.2015.2411872.

[6] van Oort, E. R., Chu, Q. P., and Mulder, J. A., "Maneuver Envelope Determination through Reachability Analysis," *Advances in Aerospace Guidance, Navigation and Control*, Springer Berlin Heidelberg, 2011, pp. 91–102. doi:10.1007/978-3-642-19817-5_8, URL http://link.springer.com/10.1007/978-3-642-19817-5{_}8.

[7] Lombaerts, T., Schuet, S., Acosta, D., Kaneshige, J., Shish, K., and Martin, L., "Piloted Simulator Evaluation of Safe Flight Envelope Display Indicators for Loss of Control Avoidance," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 4, 2016, pp. 948–963. doi:10.2514/1.G001740, URL https://arc.aiaa.org/doi/pdf/10.2514/1.G001740.

[8] Fedkiw, S., *Level Set Methods and Dynamic Implicit Surfaces*, Vol. 153, Springer, 2003. URL http://link.springer.com/content/pdf/10.1007/b98879.pdf.

[9] Mitchell, I. M., "A Toolbox of Level Set Methods (Version 1.1)," Tech. rep., 2007.

[10] Chen, M., Herbert, S., and Tomlin, C. J., "Exact and efficient Hamilton-Jacobi guaranteed safety analysis via system decomposition," *Proceedings - IEEE International Conference on Robotics and Automation*, 2017, pp. 87–92. doi: 10.1109/ICRA.2017.7989015.

[11] Bayen, A. M., Mitchell, I. M., Oishi, M., and Tomlin, C., "Aircraft Autolander Safety Analysis Through Optimal Control-Based Reach Set Computation," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, 2007, pp. 68–77. doi:10.2514/1.21562, URL http://arc.aiaa.org/doi/abs/10.2514/1.21562.

[12] Shah, G., "Aerodynamic Effects and Modeling of Damage to Transport Aircraft," *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008. URL http://arc.aiaa.org/doi/pdf/10.2514/6.2008-6203.

[13] Shah, G., and Hill, M., "Flight Dynamics Modeling and Simulation of a Damaged Transport Aircraft," *AIAA Modeling and Simulation Technologies*, 2012. URL http://arc.aiaa.org/doi/pdf/10.2514/6.2012-4632.

[14] Lombaerts, T., Huisman, H., Chu, Q. P., Mulder, J., and Joosten, D., "Nonlinear Reconfiguring Flight Control Based on Online Physical Model Identification," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, 2009, pp. 727–748. doi:10.2514/1.40788.

[15] Stevens, B., and Lewis, F., *Aircraft control and simulation*, Wiley, 1992.

[16] Cook, M. V., *Flight Dynamics Principles*, 2nd ed., Elsevier, 2013. doi:10.1016/C2010-0-65889-5.

[17] Malladi, R., Sethian, J., and Vemuri, B., "Shape Modeling with Front Propagation: A Level Set Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 2, 1995, pp. 158–175. doi:10.1109/34.368173, URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=368173.

[18] Zhao, H.-k., and Fedkiw, R., "Fast Surface Reconstruction Using the Level Set Method," *Variational and Level Set Methods in Computer Vision*, 2001, pp. 194–201. doi:10.1109/VLSM.2001.938900.

[19] Sethian, J. A., "Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts," *Journal of Computational Physics*, Vol. 169, No. 2, 2001, pp. 503–555. doi:http://dx.doi.org/10.1006/jcph.2000.6657, URL http://www.sciencedirect.com/science/article/pii/S0021999100966579.

[20] Adalsteinsson, D., and Sethian, J., "The Fast Construction of Extension Velocities in Level Set Methods," *Journal of Computational Physics*, Vol. 148, No. 1, 1999, pp. 2–22. doi:10.1006/jcph.1998.6090, URL http://www.sciencedirect.com/science/article/pii/S0021999198960909.

[21] Sethian, J. a., "Fast Marching Methods," *SIAM Review*, Vol. 41, No. 2, 1999, pp. 199–235. doi:10.1137/S0036144598347059, URL http://epubs.siam.org/doi/abs/10.1137/S0036144598347059.

[22] Arora, S., and Barak, B., *Computational Complexity: A Modern Approach*, Cambridge University Press, 2009.

[23] Goodrich, M. T., and Tamassia, R., *Algorithm Design: Foundations, Analysis, and Internet Examples*, Wiley, 2002.

[24] Govindarajan, N., De Visser, C. C., and Krishnakumar, K., "A sparse collocation method for solving time-dependent HJB equations using multivariate B-splines," *Automatica*, Vol. 50, No. 9, 2014, pp. 2234–2244. doi:10.1016/j.automatica.2014.07.012, URL http://dx.doi.org/10.1016/j.automatica.2014.07.012.