Blade Element Theory Model for UAV Blade Damage Simulation

de Alvear Cardenas, J.I.; de Visser, C.C.

DOI
10.2514/6.2024-2816

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the AIAA SCITECH 2024 Forum

Citation (APA)
de Alvear Cardenas, J. I., & de Visser, C. C. (2024). Blade Element Theory Model for UAV Blade Damage Simulation. In *Proceedings of the AIAA SCITECH 2024 Forum* Article AIAA 2024-2816 (AIAA SciTech Forum and Exposition, 2024). American Institute of Aeronautics and Astronautics Inc. (AIAA). https://doi.org/10.2514/6.2024-2816

AIAA SciTech Forum
8-12 January 2024, Orlando, FL
AIAA SCITECH 2024 Forum

10.2514/6.2024-2816

Blade Element Theory Model for UAV Blade Damage Simulation

José Ignacio de Alvear Cárdenas[*][†]
San Jose State University Research Foundation, Moffett Field, California, 94043, United States

Coen C. de Visser[‡]
Delft University of Technology, Delft, Zuid Holland, 2629HS, The Netherlands

From fault-tolerant control to failure detection, blade damage simulation is integral for developing and testing failure-resilient modern unmanned aerial vehicles. Existing approaches assume partial loss of rotor effectiveness or reduce the problem to centrifugal forces resulting from the shift in the propeller centre of gravity. In this study, a white-box blade damage model based on Blade Element Theory is proposed, integrating both mass and aerodynamic effects of blade damage. The model serves as plug-in to the nominal system model, enables the simulation of any degree of blade damage and does not require costly experimental data from failure cases. A complementary methodology for the identification of the airfoil lift and drag coefficients is also presented. Both contributions were demonstrated with the Bebop 2 drone platform and validated with static test stand wrench measurements obtained at 3 levels of blade damage (0%, 10%, 25%) in a dedicated wind tunnel experimental campaign with velocities up to 12 m/s. Results indicate high accuracy in simulating a healthy propeller. In the presence of blade damage, the model exhibits a relative error between 5% and 24% at high propeller rotational speeds and between 15% and 75% at low propeller rotational speeds.

I. Nomenclature

BD = Blade damage, %
BL, BS = Blade and blade section
C_d, C_l = Airfoil drag and lift coefficients
c_c, c_r, c_t = Longest chord length, root chord and tip chord, m
D = Drag force, N
$\overrightarrow{d}$ = Conversion matrix from rotational rates to linear velocities
d_r = Blade section length, m
F = Force, N
g = Gravitational acceleration, m/s^2
h = Trapezoid height, m
i_p = Propeller incidence angle, °
k_x, k_y = Linear inflow weighting factors
L = Lift force, N
l, b = Distance from the propeller centre of rotation to the body coordinate frame x- and y-axes, m
M = Moment, Nm
m = Mass, kg
$\dot{m}$ = Mass flow, kg/s
n_b, n_{bs} = Number of blades and blade sections
n_t = Number of trapezoids in which a blade is divided
P = Propeller
Q = Torque, Nm
q = Number of data samples
R = Propeller radius, m

---

[*] Project Associate, Human Systems Integration Division, San Jose State University Research Foundation, jose.dealvearcardenas@sjsu.edu

[†] Work performed as MSc Student, Faculty of Aerospace Engineering, Control and Simulation Division, Delft University of Technology

[‡] Associate Professor, Faculty of Aerospace Engineering, Control and Simulation Division, Delft University of Technology, AIAA member

r_{CG} = Distance between the propeller centre of rotation and centre of gravity, m
T = Thrust, N
V = Linear velocity, m/s
V_A = True airspeed, m/s
V_R = Airspeed at the rotor, m/s
V_w = Wind speed, m/s
v_0, v_i = Uniform and linear induced velocities, m/s
$\bar{y}_c$ = Span-wise centroid location, m
α = Angle of attack, rad
α_d = Angle of attack of the rotor disk relative to the oncoming flow, rad
γ = Gradient-descent optimisation learning rate
ϵ = Model error
ζ = Rotation direction boolean
θ = Pitch angle, rad
θ_{tw} = Blade twist rate per rotor radius, rad/mm
λ_j = Angle between the blade j with its propeller's x-axis
μ_x = Tip speed ratio or advanced ratio
ξ_{jk} = Damage indicator boolean
ρ = Air density, kg/m^3
σ = Standard deviation
ϕ = Roll angle, rad
χ = Wake skew angle, rad
ψ = Yaw angle, rad
ψ_k = Blade section azimuth angle, rad
ω = Propeller rotational speed, rad/s
Ω = Vehicle angular velocity, rad/s

II. Introduction

Fault is defined as "*an unpermitted deviation of at least one characteristic feature of the system from the acceptable, usual, standard condition*" [1], reducing its capability of performing a required task. Failure and malfunction are the result of the accumulation of one or more faults that lead to the permanent interruption or intermittent irregularity in the performance of a system function under the specified operating conditions.

Failures in Unmanned Aerial Vehicles (UAVs) are categorised based on their occurrence in different components, namely sensor faults, actuator faults, and plant faults [2], being the first two groups those that most literature aim at predicting. On the one hand, sensor faults arise from incorrect readings from system instruments and sensors, encompassing constant bias faults (stuck sensor), drift faults (additive-type), constant gain faults (multiplicative-type), and outlier faults [3, 4]. Actuator failures, on the other hand, result from the total loss or degradation of the propeller, motor, or electronic speed controllers [5]. They are classified into four categories: actuator saturation, actuator lock, actuator fly-off, and propeller damage. Propeller damage, characterised by the chipping or breaking of a blade, presents a unique simulation challenge due to the asymmetrical forces and moments it introduces to the system that go beyond the change in thrust.

To enhance the resilience of multi-rotor and hybrid UAVs to potential failures, work is carried out in multiple fronts, including obstacle avoidance [6], upset recovery [7], fault-tolerant control [8–10] or fault detection and diagnosis [11]; the latter consisting of the fault classification, as well as its location and magnitude identification. For all these tasks, researchers use models to simulate their systems and failures for the training or testing of their approaches before deployment. The complexity of aerospace systems often necessitates the use of grey- or black-box models obtained through system identification [12]. Consequently, simulations are constrained to the failure cases within the flight envelope of the collected data, typically obtained from costly wind tunnel experimental campaigns. Additionally, acquiring sufficient data for system identification from highly damaged cases is challenging due to concerns about operator safety and system survivability in the event of a potential loss of control. These limitations restrict the range of captured failures available for simulation, relying on the interpolation and extrapolation of the few experimentally tested scenarios.

Previous literature in the field of fault diagnosis has exploited simplified simulations of blade damage. Avram et al. [13] treat quadrotor actuator faults, such as structural damage to the propellers or rotor degradation, as a partial loss of effectiveness — a partial loss of thrust generated by the damaged rotor. This is simulated by multiplying the commanded rotor angular velocity by a factor lower than one in order to obtain the "true" rotor angular velocity. This approach, however, overlooks vibrations resulting from the unbalance of forces and moments in the system.

Another strategy, proposed by Ghalamchi et al. [14], introduces sinusoids in force signals to simulate vibrations caused by propeller unbalance. The sinusoids only consist of the decomposition of the centrifugal force in the x and y components caused by the displacement of the propeller centre of gravity due to blade damage. Unfortunately, this approach neglects the vibrations in moment signals and those induced by the changed aerodynamics due to the displacement of the centre of pressure.

Developing more accurate blade damage models is crucial for creating realistic simulations that can reveal subtle data features, potentially improving UAV on-board failure detection and diagnosis capabilities. A promising technique for modelling forces and moments is Blade Element Theory (BET), extensively used in helicopters [15], UAVs [16–18] and wind turbines [19]. BET discretises the propeller radially into a finite number of segments, each producing a differential thrust and torque, and it is based on the assumption that the wrenches generated by the complete (rotor) blade can be computed by the addition of the individual contributions of each of its span-wise elements. For this purpose, 2D airfoil characteristics are exploited whereas 3D effects are ignored. Previously, this approach has been used to model propeller thrust [20] but it has never been explored for blade damage modelling.

In this paper, a white-box blade damage simulation model based on BET is proposed, implemented, and validated. It complements the identified healthy UAV model, providing the difference in forces and moments with respect to the nominal system. Unlike existing methods, the approach takes into account the effects of both shifts in the centres of gravity and pressure, enabling the injection of any level of failure without the need for costly and dangerous system identification experiments. To the authors' knowledge, this is the first time BET is used for UAV blade damage simulation and the first time mass and aerodynamic effects are modelled together in order to shift research towards more realistic white-box blade damage models. Furthermore, this paper also presents a method for identifying the (mostly unknown) UAV blade lift and drag curves with respect to the angle of attack using BET, an approach never tried before in literature.

The proposed model has been applied to the Parrot Bebop® 2 UAV and validated by comparing its predictions to the wrench signals of a damaged propeller at multiple degrees of failure. The validation data were acquired during a dedicated wind tunnel experimental campaign at the Open Jet Facility at Delft University of Technology. The experimental setup allowed the controlled variation of environmental variables such as the wind speed (ranging from 0 and 12 m/s) and the propeller incidence angle (varying between 0 and $\pi/2$ rad).

The paper is organised as follows. First section III, section IV and section V describe the methodology, where the first two explain the mass and aerodynamic effects and the third addresses the identification of the propeller lift and drag curves. The flow of these computations and the complete approach is illustrated in advance for the reader in Fig. 1. In section VI, the proposed approach is applied to the Bebop 2 UAV to show its potential in characterising blade damage on a real platform. Subsequently, section VII validates the results through static test stand wind tunnel experiments, highlighting some limitations of both the proposed model and test setup. Finally, section VIII presents the conclusions of this research and recommendations for future work.

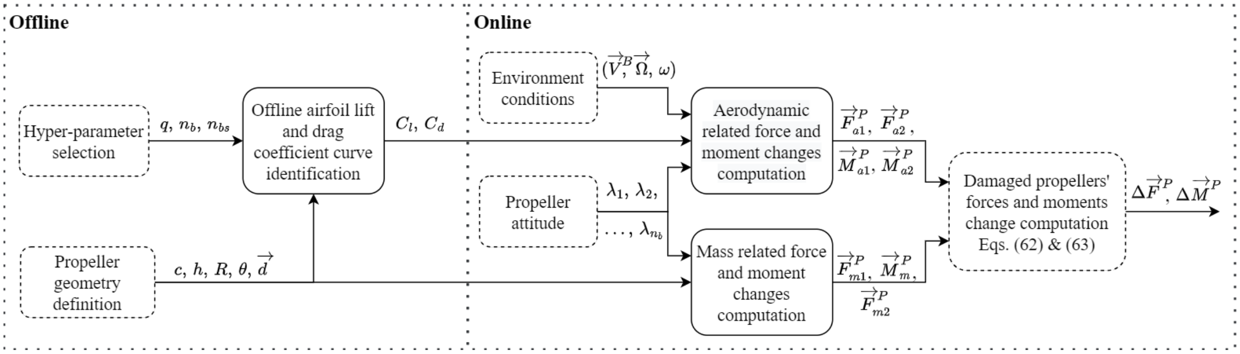


Fig. 1 Flowchart of the damaged propeller offline and online computations. The lift and drag coefficient curves identification takes place offline, whereas the computation of forces and moments due to propeller damage are performed online. The blocks with a solid edge line are further expanded in the methodology sections.

III. Mass effects

This section discusses the moments and forces that emerge from the change in propeller mass and its corresponding shift in centre of gravity for a single propeller with blade damage. They are all measured in the propeller's reference frame, which is the equivalent to the body reference frame translated to the centre of the propeller's hub.

The first forces to be obtained are those caused by the loss of mass. Since the goal is to compute the forces and moments that have to be added to those resulting from the physics model when there is no failure, the force required to be added is in the opposite direction of the gravity vector, as can be seen in Eq. (1). Here, m_{loss} is the lost mass and $\overrightarrow{R}_{PI}$ is the transformation matrix from the inertial to the propeller coordinate frame, which can be seen in Eq. (2). Depending on the drone attitude, the gravity vector can have a value in all three dimensional components of the propeller coordinate frame.

$$\overrightarrow{F}^{P}_{m1} = \overrightarrow{R}_{PI} \begin{bmatrix} 0 \\ 0 \\ -g m_{\text{loss}} \end{bmatrix} \tag{1}$$

$$\overrightarrow{R}_{PI} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{2}$$

Second, the shift in the centre of gravity (CG) causes the appearance of moments around the centre of rotation of the propeller. In order to compute these moments, the arm from the propeller central hub to the new CG location must be computed. For that purpose, the blade has been modelled as a group of trapezoids. As can be observed in Fig. 2, in the case of the Bebop 2 propeller, its blade can be split up in two trapezoids connected at their base, which is situated at the location of the largest blade chord (c_c). c_r and c_t are the chords lengths at the root and the tip, respectively.

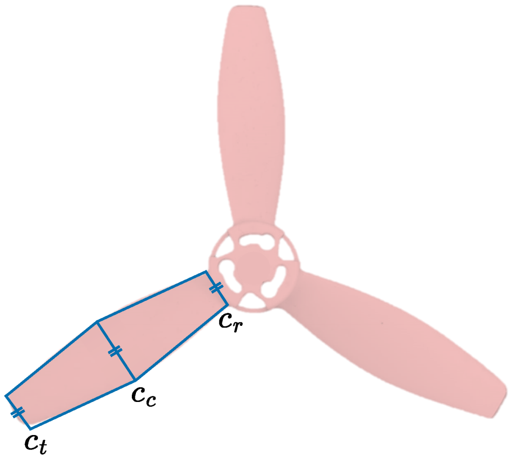


Fig. 2 Bebop 2 propeller top view and trapezoid simplification.

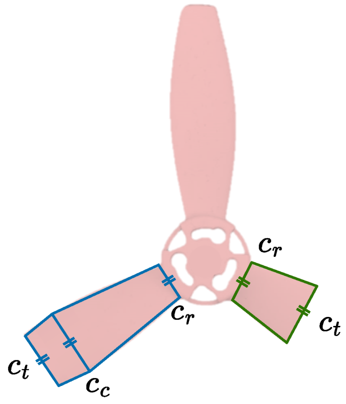


Fig. 3 Damaged Bebop 2 propeller top view and trapezoid simplification.

The CG of each blade is computed separately depending on whether it has damage or not. In the case that there is damage, the tip chord will move along the span of the blade toward the central hub. For the Bebop 2, in the case that damage causes partial blade loss closer to the central hub than the location of c_c, then there would be only one trapezoid in the blade planform and c_c would disappear, as can be seen in Fig. 3.

In the computation of the CG, a constant density assumption is made for the blades, ensuring that the CG coincides with the centroid of each blade. The centroid of each trapezoid is calculated using Eq. (3)* and they are weighted together with their respective areas for the blade centroid calculation using Eq. (4). Figure 4 illustrates the trapezoid geometrical variables and n_t stands for the number of trapezoids within a blade.

---

*For the first trapezoid ($i = 1$), $\sum_{j=1}^{i-1} h_j$ is the empty sum, meaning that it is equal to 0.

$$\bar{y}_{c_{\text{trapezoid}_i}} = \frac{h_i}{3}\frac{2\min(c_i, c_{i+1}) + \max(c_i, c_{i+1})}{c_i + c_{i+1}} + \sum_{j=1}^{i-1} h_j \quad (3)$$

$$\bar{y}_{\text{CG}_{BL}} = \bar{y}_{c_{BL}} = \frac{\sum_i^{n_t} \bar{y}_{c_{\text{trapezoid}_i}} (c_i + c_{i+1}) h_i/2}{\sum_i^{n_t} (c_i + c_{i+1}) h_i/2} \quad (4)$$

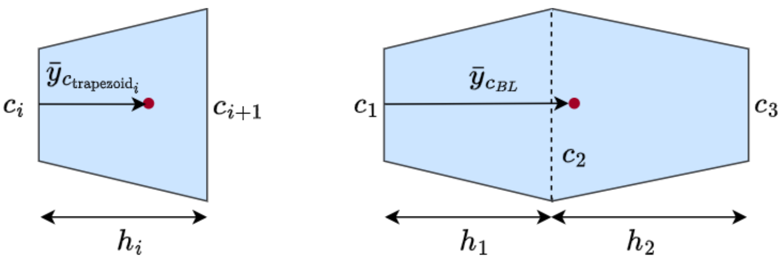


Fig. 4 Blade and trapezoid geometry and centroid.

Similarly, the centroids of individual blades, weighted by their areas, determine the overall propeller centroid. Blade attitude angles, denoted as λ_1, λ_2, ..., λ_{n_b} (where n_b is the number of blades) — the angle each blade makes with respect to the x-axis of the propeller coordinate frame — guide the decomposition of blade centroids into x and y coordinates. Given the propeller's CG to its centre of rotation (Eq. (5)), moments due to the gravity force are computed in Eq. (6), with m_P denoting propeller mass.

$$\vec{r}^{\,P}_{CG} = \begin{bmatrix} x^P_{CG} \\ y^P_{CG} \\ 0 \end{bmatrix} \quad (5)$$

$$\vec{M}^P_m = \vec{r}^{\,P}_{CG} \times \left(\vec{R}_{PI} \begin{bmatrix} 0 \\ 0 \\ g m_P \end{bmatrix} \right) \quad (6)$$

Third, thanks to the shift of the centre of gravity and the rotation of the propeller, a centrifugal force is created. The magnitude of the centrifugal force is computed with Eq. (7), where ω is the rotational velocity and r_{CG} is the distance between the centres of rotation and gravity, as computed in Eq. (8).

$$F_{m2} = m_P \omega^2 r_{CG} \quad (7)$$

$$r_{CG} = \sqrt{(x^P_{CG})^2 + (y^P_{CG})^2} \quad (8)$$

Finally, Eq. (9) shows the centrifugal force decomposed in the x and y components, where θ_{CG} is the angle between $\vec{r}_{CG}$ and the propeller coordinate frame computed using Eq. (10). This centrifugal force is illustrated in Fig. 5.

$$\vec{F}^P_{m2} = \begin{bmatrix} F_{m2}\cos\theta_{CG} \\ F_{m2}\sin\theta_{CG} \\ 0 \end{bmatrix} \quad (9)$$

$$\theta_{CG} = \arctan \frac{y^P_{CG}}{x^P_{CG}} \quad (10)$$

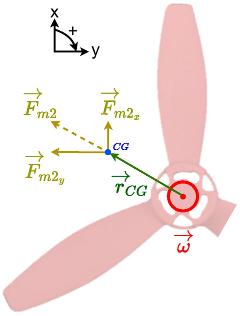


Fig. 5 Damaged propeller centrifugal force.

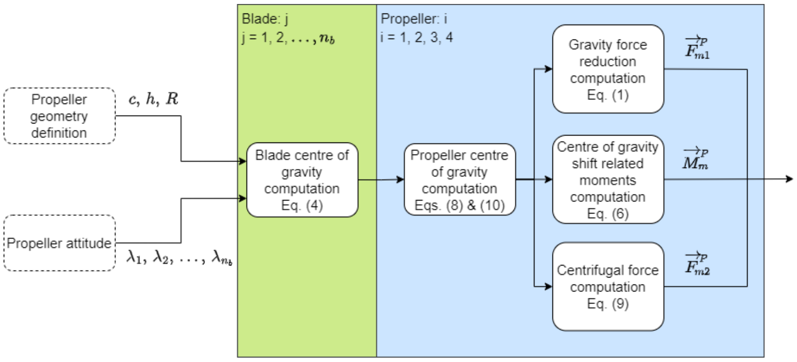


Fig. 6 Flowchart of the computation of the damaged propeller mass related forces and moments at a single time step during simulation.

In Fig. 6, a flowchart details the discussed computations of the damaged propeller mass related forces and moments.

IV. Aerodynamic effects

This section discusses the moments and forces that emerge from the change in aerodynamics for a single propeller upon blade damage. For their computation, the Blade Element Theory mathematical process is exploited. This method discretises the blade along its span in sections of equal length, determines their individual generated moments and forces, and adds all of them in order to obtain those generated by the complete propeller. When a blade is damaged, the forces and moments that would have been generated by the missing blade sections would be subtracted from those computed by the physics model in the healthy state.

A. Blade Element Theory

The goal of the BET method is the computation of the thrust and torque generated by the complete blade through the sum of the contributions of all its sections. For that purpose, the lift (ΔL_k) and drag (ΔD_k) equations, which can be seen in Eq. (11) and Eq. (12), are applied to each of the blade sections k of span length equal to dr. For the rest of the paper, the subscript i stands for the propeller, j for the blade and k for the blade section.

$$\Delta L_k(r_k, \psi_k) = \frac{1}{2} C_{l_k}(\alpha_k(r_k, \psi_k)) \rho V_{A_k}^2(r_k, \psi_k) c_k(r_k) dr \tag{11}$$

$$\Delta D_k(r_k, \psi_k) = \frac{1}{2} C_{d_k}(\alpha_k(r_k, \psi_k)) \rho V_{A_k}^2(r_k, \psi_k) c_k(r_k) dr \tag{12}$$

ρ is the air density which depends on the altitude at which the drone flies with respect to the sea level. Furthermore, c_k is the blade section average chord, V_{A_k} is the airspeed seen by the blade section perpendicular to its span and C_{l_k} and C_{d_k} are the lift and drag coefficients of the 2D blade airfoil, respectively. As can be seen, these parameters are a function of the angle of attack (α_k), the distance from the blade section centroid to the centre of rotation (r_k) and the blade section azimuth angle (ψ_k). The last one is an angle measured on the propeller plane and it is defined to have a value of zero degrees (ψ=0) in the direction of the drag, increasing its value in the direction of rotation. The r and ψ definitions can be visualised in Fig. 7 and Fig. 8, respectively. When a variable is a function of r and ψ, it will be represented by $(\cdot)$ for readability purposes.

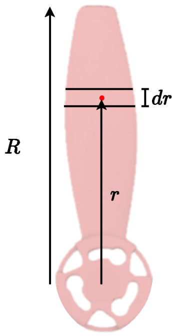


Fig. 7 Blade geometrical parameters.

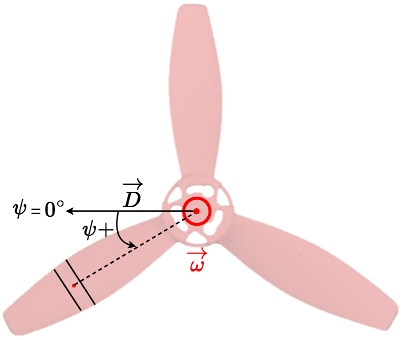


Fig. 8 Azimuth angle visualisation.

The airspeed at each blade section (V_{A_k}) has to be computed taking into account three main components: the vehicle combined linear and angular velocities, the propeller rotational velocity and the induced velocity. First, Eq. (13) is used to compute the linear velocity of the propeller assembly ($\vec{V}^P$) from the body linear ($\vec{V}^B$) and angular velocities ($\vec{\Omega}$). The $\vec{d}$ matrix presented in Eq. (14) is used to convert the rotational rates of the vehicle to linear velocities, exploiting the known drone geometry shown in Fig. 9 [12]. Each row of the d matrix corresponds to each of the drone propellers. The first row corresponds to the front-left propeller and the following rows to the other propellers moving clockwise from a top-down view of the drone.

$$\vec{V}_i^P = \vec{\Omega} \times \vec{d}_i^T + \vec{V}^B \tag{13}$$

$$\vec{d} = \begin{bmatrix} l & -b & 0 \\ l & b & 0 \\ -l & b & 0 \\ -l & -b & 0 \end{bmatrix} \tag{14}$$

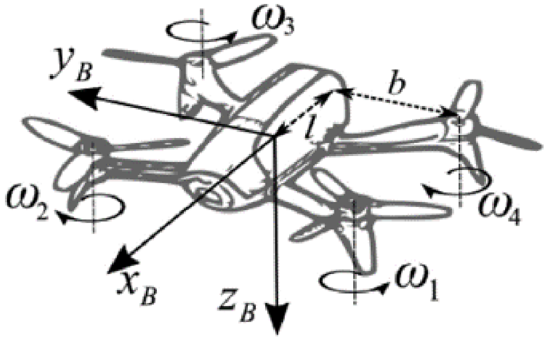


Fig. 9 Drone geometry [12].

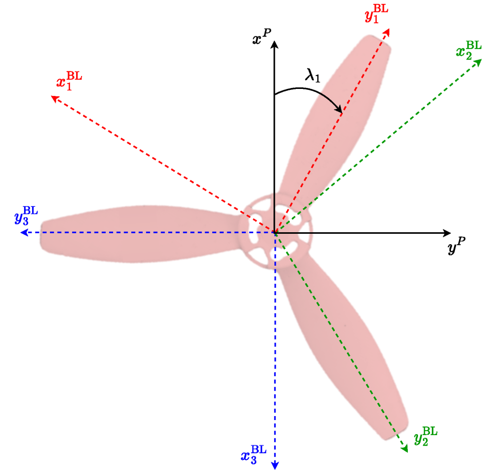


Fig. 10 Blade coordinate frame.

Then, the propeller linear velocity is translated to the blade coordinate frame (BL), which rotates with the respective blade, as can be seen in Fig. 10. The angle of the blade with the propeller coordinate frame x-axis is λ_j and it is used in Eq. (15) for the coordinate frame transformation. As can be observed, a minus sign precedes the transformation matrix because the airspeed vector is opposite to the displacement direction. It is assumed that this value of airspeed, which is a function of the vehicle linear and angular velocities, does not depend on the position along the blade.

$$\overrightarrow{V}^{BL}_{A_{ijk_1}} = \overrightarrow{V}^{BL}_{A_{ij}} = -\begin{bmatrix} \sin\lambda_j & -\cos\lambda_j & 0 \\ \cos\lambda_j & \sin\lambda_j & 0 \\ 0 & 0 & 1 \end{bmatrix} \overrightarrow{V}^{P}_{i} \tag{15}$$

Second, the component of the velocity due to the rotation of the propeller is the product of the distance of the blade section centroid to the centre of rotation (r_{ijk}) and the rotational velocity of the propeller (ω_i), as can be seen in Eq. (16). The main benefit of the chosen blade coordinate frame is that this velocity component only exists along the x-axis. ζ_i is a variable which acquires a value of 1 if the i^{th} propeller is rotating clockwise and -1 if it is rotating counter-clockwise.

$$\overrightarrow{V}^{BL}_{A_{ijk_2}}(r_{ijk}) = \begin{bmatrix} \zeta_i \omega_i r_{ijk} \\ 0 \\ 0 \end{bmatrix} \tag{16}$$

Third, there exist multiple approaches in literature for computing the induced velocity field across the rotor disk, most of the them based on estimates and empirical tests. The work of Gill et al. [21] assumes ideal propeller geometry, considering a constant uniform induced velocity along the propeller, which is mostly not the case in forward flight. The approach followed for the present research is the same one used by Niemiec et al. [22] and that is thoroughly explained by Leishman et al. [23], which combines an initial uniform inflow estimation for the complete propeller with local (blade section) linear inflow model corrections.

For the computation of the uniform induced velocity (v_0[†]), the Glauert formula presented in Eq. (19) is derived from the combination of the mass flow and the propeller thrust equations, shown in Eq. (17) and Eq. (18), respectively. According to the principles of momentum and energy conservation, the far wake velocity equals the airspeed before the rotor plus two times the induced velocity [23], leading to a change in velocity across the rotor of $\Delta V = 2v_0$.

$$\dot{m} = \rho \pi R^2 V_R \tag{17}$$

$$T = \dot{m}\Delta V = \dot{m}(V_{A_i} + 2v_0 - V_{A_i}) = 2\dot{m}v_0 \tag{18}$$

$$v_0 = \frac{T}{2\rho\pi R^2 V_R} \tag{19}$$

[†] Since it will be constantly referred to the same single propeller, the subscript i to denote a specific propeller is dropped for the rest of the paper for readability purposes.

Given that the airspeed at the rotor (V_R) equals the propeller airspeed plus the induced velocity along the z-axis direction, it is expressed in Eq. (20) using the translational velocity found in Eq. (13). As the final Glauert equation lacks a closed form, the induced velocity has to be computed using an iterative optimisation technique, such as Nelder-Mead. To that end, a tailored and efficient gradient-descent (1D first-order derivative) algorithm is detailed in the Appendix.

$$V_R = \sqrt{V_x^P V_x^P + V_y^P V_y^P + (-V_z^P + v_0)^2} \tag{20}$$

After determining the uniform inflow velocity, it can be used as the basis for the computation of the linear inflow model. Several estimation models are available, such as those proposed by Howlett [24], Pitt & Peters [25] and White & Blake [26]. However, for the present research, the Drees model [27] is employed since it is one of the best representations when compared to empirical data [23]. Equation (21) expresses the induced velocity using the uniform inflow as a basis and adjusting it with the k_x and k_y weighting factors, calculated in Eq. (22) and Eq. (23), respectively.

$$v_i(\cdot) = v_0(1 + k_x r \cos\psi + k_y r \sin\psi) \tag{21}$$

$$k_x = \frac{4}{3}\frac{(1 - \cos\chi - 1.8\mu_x^2)}{\sin\chi} \tag{22}$$

$$k_y = -2\mu_x \tag{23}$$

χ is the wake skew angle or the angle between the wake and the propeller's z-axis (Fig. 11), computed using the propeller airspeed and the uniform induced velocity according to Eq. (24). μ_x is the tip speed ratio or advanced ratio defined as the airspeed projected on the x-y plane in the propeller coordinate frame normalised by the blade length and propeller rotational velocity (Eq. (25)). The resulting inflow, illustrated in Fig. 12, depicts the variation of induced velocity across blade sections with respect to their distance from the centre of rotation and azimuth angle.

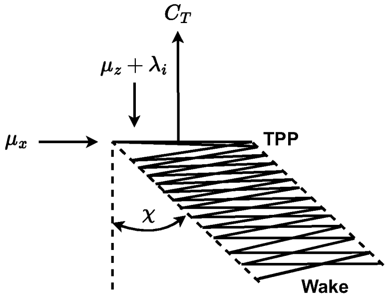


Fig. 11 Illustration of the wake skew angle [23].

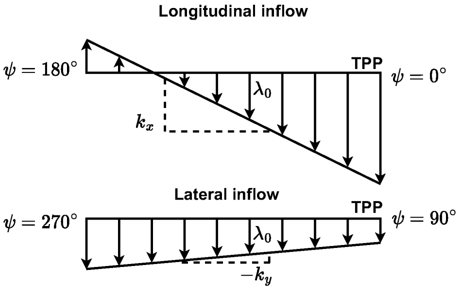


Fig. 12 Linear inflow model [23].

$$\tan\chi = \frac{\sqrt{V_x^P V_x^P + V_y^P V_y^P}}{v_0 - V_z^P} \tag{24}$$

$$\mu_x = \frac{\sqrt{V_x^P V_x^P + V_y^P V_y^P}}{\omega_i R} \tag{25}$$

After computing the three components of the blade section airspeed, they are summed in Eq. (26).

$$\vec{V}^{BL}_{A_{jk}}(\cdot) = \vec{V}^{BL}_{A_{jk_1}} + \vec{V}^{BL}_{A_{jk_2}}(r_{jk}) + \begin{bmatrix} 0 & 0 & v_{i_{jk}}(\cdot) \end{bmatrix}^T \tag{26}$$

Given the airspeed, the angle of attack seen by the blade can be obtained from Eq. (27). Here, θ_{jk} is the average twist of the blade section and it is a linear function of the distance from the centre of rotation (Eq. (28), where θ_{tw} is the blade twist rate per radius of the rotor and θ_0 is the blade twist at the blade root). The closer to the root, the higher the twist in order to compensate for the lower tangential velocity from the propeller rotation.

$$\alpha_{jk}(\cdot) = \theta_{jk}(r_{jk}) - \arctan\frac{V^{BL}_{A_{jk_z}}(\cdot)}{V^{BL}_{A_{jk_x}}(\cdot)} \tag{27}$$

$$\theta_{jk}(r_{jk}) = \theta_0 - r_{jk}\theta_{tw} \tag{28}$$

Finally, from the blade section lift and drag contributions, it is possible to determine the generated thrust (ΔT) and torque (ΔQ) by each blade section with Eq. (29) and Eq. (30), respectively[‡]. Here, ϕ_{jk} is the blade section flow angle

---

[‡] Since every term of these equations is meant for a particular blade section, the subscripts jk have been left out to enhance readability. For the same reason, the variables each of the terms are a function of have also been removed; they are all a function of (ψ_{jk}, r_{jk}).

defined as the difference between the local twist angle θ_{jk} and the local angle of attack α_{jk}, as can be seen in Eq. (31).

$$\Delta T = \Delta L \cos\phi - \Delta D \sin\phi \tag{29}$$

$$\Delta Q = r\Delta L \sin\phi + r\Delta D \cos\phi \tag{30}$$

$$\phi_{jk}(\cdot) = \theta_{jk}(r_{jk}) - \alpha_{jk}(\cdot) \tag{31}$$

B. Aerodynamic forces and moments

Once the percentage of blade damage to be simulated is known, the corresponding "lost" blade sections are identified. For instance, if a 30% blade damage is considered in a blade composed of 100 sections, the 30 sections closest to the blade tip are the ones lost. Then, their forces and moments are added to later be subtracted from those of the nominal operating conditions obtained from the healthy blade model output.

Equation (29) can be used for the computation of the thrust of a single blade section. The added thrust value of all the lost blade sections leads to $F^P_{a1_z}$, as given by Eq. (32). ξ_{jk} is a boolean which has a value of 1 when the blade section is damaged and 0 when it is in its nominal state. The moments emerging about the propeller x- and y-axes from the change in the centre of pressure location are computed by decomposing the moment generated by the blade section thrust around the centre of rotation with Eq. (33) and Eq. (34).

$$F^P_{a1_z} = \sum_{j=1}^{n_b} \sum_{k=1}^{n_{bs}} \xi_{jk} \Delta T_{jk} \tag{32}$$

$$M^P_{a2_x} = \sum_{j=1}^{n_b} \sum_{k=1}^{n_{bs}} -\xi_{jk} \Delta T_{jk} r_{jk} \sin\lambda_j \tag{33}$$

$$M^P_{a2_y} = \sum_{j=1}^{n_b} \sum_{k=1}^{n_{bs}} \xi_{jk} \Delta T_{jk} r_{jk} \cos\lambda_j \tag{34}$$

Furthermore, Eq. (30) can be used for the computation of the torque of each blade section. The integral torque of the lost blade sections leads to the moment about the z-axis, as shown in Eq. (35). The blade section force in the rotor plane can be obtained by dividing the torque by the magnitude of the moment arm. This force is then decomposed in the x- and y-directions, as shown in Eq. (36) and Eq. (37), in order to obtain the last aerodynamic forces.

$$M^P_{a1_z} = \sum_{j=1}^{n_b} \sum_{k=1}^{n_{bs}} \xi_{jk} \Delta Q_{jk} \tag{35}$$

$$F^P_{a2_x} = \sum_{j=1}^{n_b} \sum_{k=1}^{n_{bs}} \xi_{jk} \frac{\Delta Q_{jk}}{r_{jk}} \cos\left(\lambda_j - \zeta\frac{\pi}{2}\right) \tag{36}$$

$$F^P_{a2_y} = \sum_{j=1}^{n_b} \sum_{k=1}^{n_{bs}} \xi_{jk} \frac{\Delta Q_{jk}}{r_{jk}} \sin\left(\lambda_j - \zeta\frac{\pi}{2}\right) \tag{37}$$

These force and moment vectors are subtracted from the nominal physics model output to account for the aerodynamic effects. In Fig. 13, a flowchart visually illustrates their discussed computation.

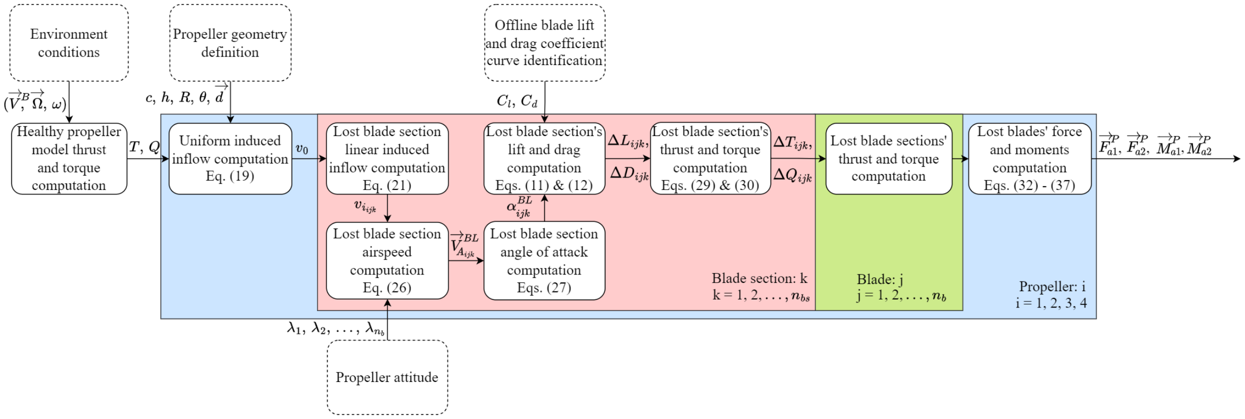


Fig. 13 Flowchart of the computation of the damaged propeller aerodynamic forces and moments at one time step during simulation.

V. Airfoil lift and drag coefficients identification

For most commercial drones, a propeller blade's aerodynamic model is absent. Prior work [20] employs Hybrid Blade Element Momentum Theory, equating the thrust derived from BET (blade geometry) and the thrust derived from Momentum Theory (interacting flow characteristics), for the computation of the induced velocity and aerodynamic coefficients. Its two-step method, estimating induced velocity and then the aerodynamic coefficients, has two drawbacks. Firstly, it lacks clarity on the induced velocity computation without the required prior aerodynamic parameter calculation. Secondly, the optimisation method takes 2 to 10 hours for 20 data points, making simulations of blade damage scenarios in which the induced velocity must be computed for every time step, unfeasible. This section details an alternative optimisation approach for identifying the lift and drag coefficient curves with much shorter computation times.

The proposed method is based on equating the thrust and torque obtained from the healthy UAV model (or experimental data) to the sum of the blade sections' moment and forces computed following BET. To that end, given a drone geometry (R, and l and b from Eq. (14)), combinations of vehicle body linear ($\overrightarrow{V}^B$) and angular velocities ($\overrightarrow{\Omega}$), as well as propeller rotational velocities (ω), are fed as input to both models. Then, their outputs are combined in order to solve a constrained minimisation problem that finds the lift and drag coefficient functions. C_l and C_d are modelled as polynomials of m and n degree as a function of the angle of attack, as can be seen in Eq. (38) and Eq. (39), respectively.

$$C_{l_{jk}} = x_0 + x_1\alpha_{jk} + x_2\alpha_{jk}^2 + \cdots + x_m\alpha_{jk}^m \tag{38}$$

$$C_{d_{jk}} = y_0 + y_1\alpha_{jk} + y_2\alpha_{jk}^2 + \cdots + y_n\alpha_{jk}^n \tag{39}$$

With these lift and drag coefficient polynomials, BET thrust and torque are computed with Eq. (40) and Eq. (41).

$$\begin{aligned} T &= \left(\sum_j^{n_b}\sum_k^{n_{bs}} \frac{1}{2}\rho C_{l_{jk}} c_{jk} V_{A_{jk}}^2 \cos\phi_{jk} dr\right) - \left(\sum_j^{n_b}\sum_k^{n_{bs}} \frac{1}{2}\rho C_{d_{jk}} c_{jk} V_{A_{jk}}^2 \sin\phi_{jk} dr\right) \\ &= \frac{1}{2}\rho dr\left[\left(\sum_o^m x_o \sum_j^{n_b}\sum_k^{n_{bs}} \alpha_{jk}^o c_{jk} V_{A_{jk}}^2 \cos\phi_{jk}\right) - \left(\sum_o^n y_o \sum_j^{n_b}\sum_k^{n_{bs}} \alpha_{jk}^o c_{jk} V_{A_{jk}}^2 \sin\phi_{jk}\right)\right] \end{aligned} \tag{40}$$

$$Q = -\zeta\frac{1}{2}\rho dr\left[\left(\sum_o^m x_o \sum_j^{n_b}\sum_k^{n_{bs}} \alpha_{jk}^o r_{jk} c_{jk} V_{A_{jk}}^2 \sin\phi_{jk}\right) + \left(\sum_o^n y_o \sum_j^{n_b}\sum_k^{n_{bs}} \alpha_{jk}^o r_{jk} c_{jk} V_{A_{jk}}^2 \cos\phi_{jk}\right)\right] \tag{41}$$

With the previous definitions of the thrust and the torque, it is possible to create a system of the $\overrightarrow{A}\overrightarrow{x} = \overrightarrow{b}$ form. Each pair of rows of the A matrix and the b vector corresponds to the thrust and torque of a data point from the BET and healthy UAV models, respectively. A data point refers to a set of conditions ($\overrightarrow{V}^B$, $\overrightarrow{\Omega}$, ω) that are provided as input to both models. Equations (42) to (48) show the different components of the system with q data points.

$$l_1^o = \sum_j^{n_b}\sum_k^{n_{bs}} \alpha_{jk}^o c_{jk} V_{A_{jk}}^2 \cos\phi_{jk} \tag{42}$$

$$l_2^o = -\sum_j^{n_b}\sum_k^{n_{bs}} \alpha_{jk}^o c_{jk} V_{A_{jk}}^2 \sin\phi_{jk} \tag{43}$$

$$l_3^o = -\zeta\sum_j^{n_b}\sum_k^{n_{bs}} \alpha_{jk}^o r_{jk} c_{jk} V_{A_{jk}}^2 \sin\phi_{jk} \tag{44}$$

$$l_4^o = -\zeta\sum_j^{n_b}\sum_k^{n_{bs}} \alpha_{jk}^o r_{jk} c_{jk} V_{A_{jk}}^2 \cos\phi_{jk} \tag{45}$$

$$\overrightarrow{A} = \frac{1}{2}\rho dr\begin{bmatrix} (l_1^0)_1 & (l_1^1)_1 & \cdots & (l_1^m)_1 & (l_2^0)_1 & (l_2^1)_1 & \cdots & (l_2^n)_1 \\ (l_3^0)_1 & (l_3^1)_1 & \cdots & (l_3^m)_1 & (l_4^0)_1 & (l_4^1)_1 & \cdots & (l_4^n)_1 \\ (l_1^0)_2 & (l_1^1)_2 & \cdots & (l_1^m)_2 & (l_2^0)_2 & (l_2^1)_2 & \cdots & (l_2^n)_2 \\ (l_3^0)_2 & (l_3^1)_2 & \cdots & (l_3^m)_2 & (l_4^0)_2 & (l_4^1)_2 & \cdots & (l_4^n)_2 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ (l_1^0)_q & (l_1^1)_q & \cdots & (l_1^m)_q & (l_2^0)_q & (l_2^1)_q & \cdots & (l_2^n)_q \\ (l_3^0)_q & (l_3^1)_q & \cdots & (l_3^m)_q & (l_4^0)_q & (l_4^1)_q & \cdots & (l_4^n)_q \end{bmatrix} \tag{46}$$

$$\overrightarrow{x} = \begin{bmatrix} x_0 & x_1 & \cdots & x_m & y_0 & y_1 & \cdots & y_n \end{bmatrix}^T \tag{47}$$

$$\overrightarrow{b} = \begin{bmatrix} (T)_1 & (Q)_1 & (T)_2 & (Q)_2 & \cdots & (T)_q & (Q)_q \end{bmatrix}^T \tag{48}$$

The output of the BET model, namely the $\overrightarrow{A}$ matrix, has to be averaged over a rotation of the propeller, integrating over the azimuth angle [20]. Equation (49) shows how this is done for l_1^0; the same procedure can be applied to l_2^0, l_3^0 and l_4^0. In practice, instead of integrating, a specific discrete number of rotation angles are chosen, and their contributions to the $\overrightarrow{A}$ matrix are computed and averaged.

$$l_1^o = \frac{1}{2\pi} \int_{\psi=0}^{2\pi} \sum_{j}^{n_b} \sum_{k}^{n_{bs}} \alpha_{jk}^o(\cdot) c_{jk}(r_{jk}) V_{A_{jk}}^2(\cdot) \cos \phi_{jk}(\cdot)\, d\psi \tag{49}$$

In order to guarantee that the lift and drag curves with respect to the angle of attack have their characteristic shape, a constrained optimisation problem is posed in order to include airfoil aerodynamic knowledge in the solver. Given that there is a large difference in scale between the thrust and torque values, the Normalised Root Mean Squared Error (NRMSE) is used as objective function (Eq. (50) and Eq. (51)), where the standard deviation (σ_b) of the aerodynamic healthy model output ($\overrightarrow{b}$) is the normalisation factor. Thrust and torque errors, along with their respective observations' standard deviations, are computed independently. Hence, the figure that the optimisation function targets is the averaged NRMSE of thrust and torque, as can be seen in Eq. (52), and the objective function for the aerodynamic parameter computation $\overrightarrow{x}^*$ is defined in Eq. (53).

$$\overrightarrow{\epsilon} = \overrightarrow{b} - \overrightarrow{A}\overrightarrow{x} \tag{50}$$

$$\text{NRMSE} = \frac{\sqrt{\left(\overrightarrow{\epsilon}^T \cdot \overrightarrow{\epsilon}\right)/q}}{\sigma_b} \tag{51}$$

$$\text{NRMSE}_{\text{total}} = \frac{\text{NRMSE}_T + \text{NRMSE}_Q}{2} \tag{52}$$

$$\overrightarrow{x}^* = \arg\min_{\overrightarrow{x}} \text{NRMSE}_{\text{total}} \tag{53}$$

Furthermore, the following constraints are used to achieve the lift and drag coefficient curves characteristic shapes:

1) The maximum lift coefficient can not be higher than 5 within the angle of attack range of -30 to 30 degrees: $C_l(\alpha) < 5, \quad \forall\alpha \in [-30°, 30°]$.
2) The lift coefficient curve should have a negative slope within the angle of attack range of 25 to 30 degrees: $dC_l(\alpha)/d\alpha < 0, \quad \forall\alpha \in [25°, 30°]$.
3) The lift coefficient curve should have a positive slope within the angle of attack range of 0 to 7 degrees: $dC_l(\alpha)/d\alpha > 0, \quad \forall\alpha \in [0°, 7°]$.
4) The lift coefficient curve should intersect the angle of attack axis within the angle of attack range of -10 to 10 degrees: $\min C_l(\alpha) < 0, \quad \forall\alpha \in [-10°, 10°]$.
5) The drag coefficient curve can not be negative within the angle of attack range of -30 to 30 degrees: $C_d(\alpha) > 0, \quad \forall\alpha \in [-30°, 30°]$.

For the declaration of these constraints, matrix $\overrightarrow{C}(\overrightarrow{\alpha}) \in R^{s\times(mn)}$ is created with the range of angles of attack mentioned in each constraint definition. The matrix has the same number of rows as integer angles within the constraint range, namely $s = \alpha_{\max} - \alpha_{\min} + 1$ for $\alpha \in [\alpha_{\min}, \alpha_{\max}]$; each row corresponds to an angle of attack. The number of columns equals the length of the parameter vector $\overrightarrow{x}$. Its input is an angle of attack vector which is a function of the $\alpha_{\min}$ and $\alpha_{\max}$, as can be deduced from its definition in Eq. (54).

$$\overrightarrow{\alpha}(\alpha_{\min}, \alpha_{\max}) = \Big[\alpha_{\min} + \eta\Big]_{\eta=0}^{s-1} = \begin{bmatrix}\alpha_0 & \alpha_1 & \cdots & \alpha_{s-1}\end{bmatrix}^T \tag{54}$$

Since a constraint regarding the lift coefficient curve does not require information about the drag coefficient parameters, the last n columns will be full of zeros (Eq. (55) and Eq. (57)) for constraints 1 to 4. In the case of the drag constraint (constraint 5), it would be the first m columns that would be full of zeros, as shown in Eq. (56). Constraints 2 and 3 impose a limit in the gradient of the curve, so the derivative of $\overrightarrow{C}_{C_l}$ with respect to the angle of attack is taken in Eq. (57).

$$\overrightarrow{C}_{C_l}(\overrightarrow{\alpha}) = \left[\begin{array}{ccccc|c} 1 & \alpha_0 & \alpha_0^2 & \cdots & \alpha_0^m & \\ 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^m & \overrightarrow{0}_{s\times n} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \\ 1 & \alpha_{s-1} & \alpha_{s-1}^2 & \cdots & \alpha_{s-1}^m & \end{array}\right] \tag{55}$$

$$\overrightarrow{C}_{C_d}(\overrightarrow{\alpha}) = \left[\begin{array}{c|ccccc} & 1 & \alpha_0 & \alpha_0^2 & \cdots & \alpha_0^n \\ \overrightarrow{0}_{s\times m} & 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^n \\ & \vdots & \vdots & \vdots & \cdots & \vdots \\ & 1 & \alpha_{s-1} & \alpha_{s-1}^2 & \cdots & \alpha_{s-1}^n \end{array}\right] \tag{56}$$

$$\frac{d\overrightarrow{C}_{C_l}(\overrightarrow{\alpha})}{d\alpha} = \left[\begin{array}{ccccc|c} 0 & 1 & 2\alpha_0 & \cdots & m\alpha_0^{m-1} & \\ 0 & 1 & 2\alpha_1 & \cdots & m\alpha_1^{m-1} & \overrightarrow{0}_{s\times n} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \\ 0 & 1 & 2\alpha_{s-1} & \cdots & m\alpha_{s-1}^{m-1} & \end{array}\right] \tag{57}$$

The $\overrightarrow{C}$ matrix or its derivative is multiplied with the parameter vector $\overrightarrow{x}$ and the maximum or minimum value from the output is taken for the definition of the inequality constraints. As a result, the following constrained optimisation problem is posed:

$$\begin{aligned} \min_{\overrightarrow{x}} \quad & \text{NRMSE}_{\text{total}} \\ \text{s.t.} \quad & \max\left(\overrightarrow{C}_{C_l}(\overrightarrow{\alpha})\overrightarrow{x}\right) - 5 < 0,\ (\alpha_{\min}, \alpha_{\max}) = (-30, 30) \\ & \max\left(\frac{d\overrightarrow{C}_{C_l}(\overrightarrow{\alpha})}{d\alpha}\overrightarrow{x}\right) < 0,\ (\alpha_{\min}, \alpha_{\max}) = (25, 30) \\ & \min\left(\frac{d\overrightarrow{C}_{C_l}(\overrightarrow{\alpha})}{d\alpha}\overrightarrow{x}\right) > 0,\ (\alpha_{\min}, \alpha_{\max}) = (0, 7) \\ & \min\left(\overrightarrow{C}_{C_l}(\overrightarrow{\alpha})\overrightarrow{x}\right) < 0,\ (\alpha_{\min}, \alpha_{\max}) = (-10, 10) \\ & \min\left(\overrightarrow{C}_{C_d}(\overrightarrow{\alpha})\overrightarrow{x}\right) > 0,\ (\alpha_{\min}, \alpha_{\max}) = (-30, 30) \end{aligned} \tag{58}$$

Figure 14 visually illustrates the lift and drag coefficient identification with a flowchart.

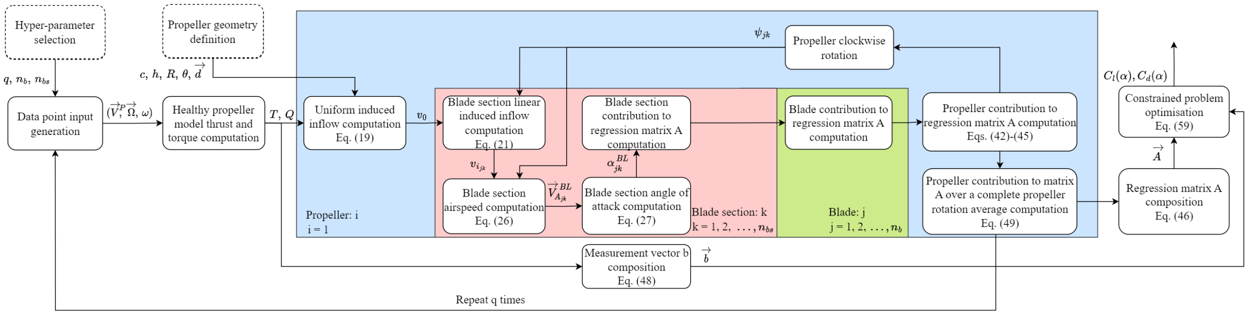


Fig. 14 Flowchart of the offline lift and drag coefficient curves identification.

VI. Results

The department of Control & Simulation at Delft University of Technology developed in 2019 a grey-box aerodynamic model of the Bebop 2 drone based on wind tunnel experiments [28]. To demonstrate the presented methodology, it will be used as the healthy UAV identified model, turning the Bebop 2 into the platform of choice.

A. Bebop 2 lift and drag coefficients identification

Table 1 summarises the geometry of the Bebop 2 drone and propeller. In section V it was mentioned that the aerodynamic grey-box and BET models require as input for each data point a set of conditions $(\overrightarrow{V}^B, \overrightarrow{\Omega}, \omega)$ beyond the drone geometry. The range of those input conditions is also explained next:

1) The drone linear velocity is constrained in the x-z plane (V_y^B=0). Its value in the z-direction (V_z^B) is sampled from a uniform distribution over the closed interval [-2, -0.5] m/s, avoiding positive velocities in the z-axis that could cause nonlinear behaviour, e.g. Vortex Ring State, that was not accounted for by the aerodynamic grey-box model.

Table 1 Geometrical properties of the Bebop 2 drone and propeller.

Drone geometry		Propeller geometry							
b	l	c_r	c_c	c_t	h_1 & h_2	R	n_b	θ_0	θ_{tw}
(mm)	(mm)	(mm)	(mm)	(mm)	(mm)	(mm)	(-)	(°)	(°/mm)
115	87.5	13	20	8	32	75	3	27	0.29

Furthermore, the value of the drone linear velocity in the x-direction (V_x^B) is also sampled from a uniform distribution, in this case over the closed interval [-3, 3] m/s. As a result, for the identification of the lift and drag curves only scenarios in which the drone is ascending, flying forward or backwards are considered.

2) The angular velocity of the drone is always zero ($\overrightarrow{\Omega} = \overrightarrow{0}$).
3) The rotation velocity of the propeller (ω) is sampled from a uniform distribution over the closed interval [300, 1256] rad/s.

Furthermore, the drag and lift curves were approximated with second degree polynomials (m and n equal 2) and the integral with respect to the azimuth angle presented in Eq. (49) was approximated with 10 discrete equally spaced azimuth angles starting at 0°, namely [0°, 36°, 72°, ..., 288°, 324°]. The constrained optimisation method is posed using Python's Scipy package with the trust-region interior point method ("trust-constr") [29] solver. Additionally, the number of blade sections (n_{bs}) and the number of data points used for the identification (q) were 100 and 16,000, respectively.

The data gathering and optimisation took 10,707 seconds (2 hours and 58 minutes) using the aforementioned chosen hyper-parameters in a consumer laptop with an Intel Core i7-9750H CPU running Python 3.7. Equations (59) and (60) show the identified polynomials that define the airfoil lift and drag curves with respect to the angle of attack. Plots of these polynomials in Fig. 15 and Fig. 16 show the characteristic shape expected from those aerodynamic curves. Only the 5th constraint in section V is limiting in the solution, namely that the drag coefficient cannot be negative.

$$C_l = 0.24 + 5.15\alpha - 12.25\alpha^2 \qquad (59) \qquad\qquad C_d = 0.0092 - 0.79\alpha + 15.13\alpha^2 \qquad (60)$$

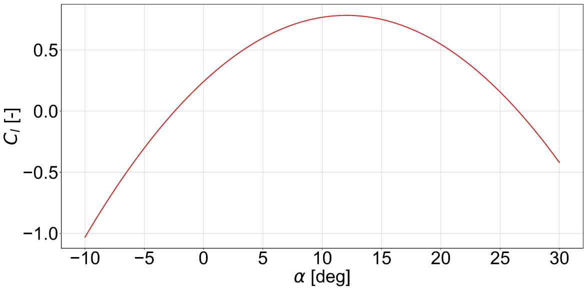


Fig. 15 Identified airfoil lift coefficient curve with respect to the angle of attack from the aerodynamic grey-box model [12] data. Illustration of Eq. (59).

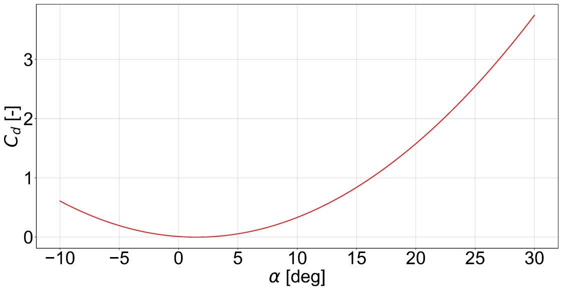


Fig. 16 Identified airfoil drag coefficient curve with respect to the angle of attack from the aerodynamic grey-box model [12] data. Illustration of Eq. (60).

The identified aerodynamic model was positively validated by verifying that the model residuals on a validation dataset of 4,000 data points (25% of the training dataset) approximate zero mean white noise.

B. Bebop 2 mass and aerodynamic forces and moments

To observe the magnitude of the forces and moments caused by the change in mass, the front left Bebop 2 propeller is simulated as damaged with a single blade loss of 20% of its length. It rotates at 600 rad/s counterclockwise from a top-down view (ζ=-1) for 0.25 seconds. The attitude of the drone is such that the z-axis direction of the propeller and inertial coordinate frame coincide. Additionally, the drone is moving with a body linear velocity $\overrightarrow{V}^B = \begin{bmatrix} 3 & 0 & -1 \end{bmatrix}^T$ [m/s] and body angular velocity $\overrightarrow{\Omega} = \overrightarrow{0}$ rad/s. The Bebop 2 propeller has a total mass of 5.07 [g] and a mass per blade (without the central hub) of 1.11 [g].

Figures 17 and 18 show the forces and moments caused by the change of mass for the aforementioned scenario. As can be observed, the signals in the x and y directions have a oscillatory behaviour due to the propeller rotations. Since the drone's propeller z-axis is aligned with its counterpart in the inertial frame, the gravity force coincides with the z-axis in the propeller coordinate frame, leading to a constant force in the z-direction and a zero moment about the z-axis.

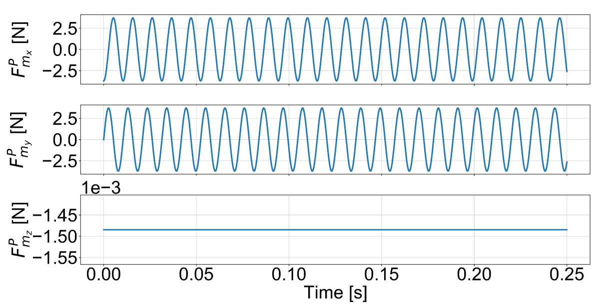


Fig. 17 BET-simulated evolution of forces caused due to mass change upon 20% Bebop 2 blade damage for 0.25 s rotating at $\omega_0 = 600$ rad/s.

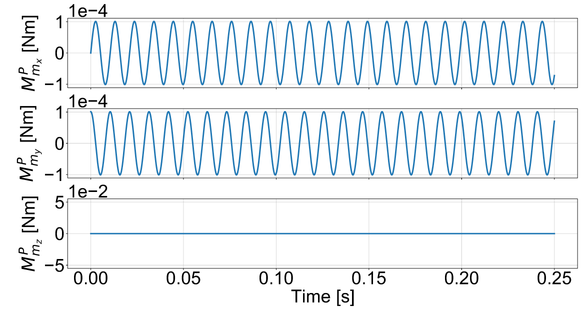


Fig. 18 BET-simulated evolution of moments caused due to mass change upon 20% Bebop 2 blade damage for 0.25 s rotating at $\omega_0 = 600$ rad/s.

Next, the aerodynamic forces and moments are also analysed for the same 0.25 s Bebop 2 propeller scenario, leading to the results illustrated in Fig. 19 and Fig. 20. As can be observed, the signals around the y-axis are centred around the 0 datum, whereas the wrenches in the x- and z-direction are biased. The wrench in the x-direction is negatively biased because the lift and drag forces are the highest when the damaged blade is advancing and not retreating. Since the propeller is rotating counterclockwise, the thrust produced when the blade is advancing creates a negative moment around the x-axis and the force creating the torque points towards the negative x-direction. The oscillatory motion in the z-direction is also due to the difference in incoming flow between when the damaged blade advances and retreats. If the drone would be hovering, then the aerodynamic wrench in this direction would be constant and will have a value approximately equal to the observed bias.

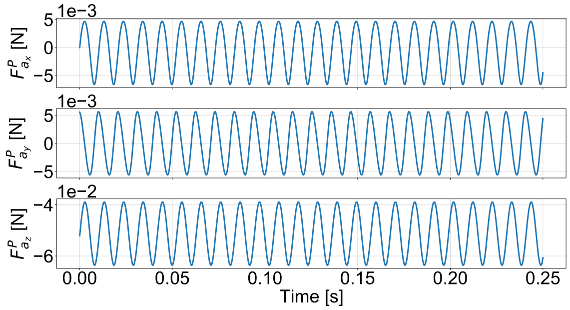


Fig. 19 BET-simulated evolution of aerodynamic forces generated by lost blade sections upon 20% Bebop 2 blade damage for 0.25 s rotating at $\omega_0 = 600$ rad/s.

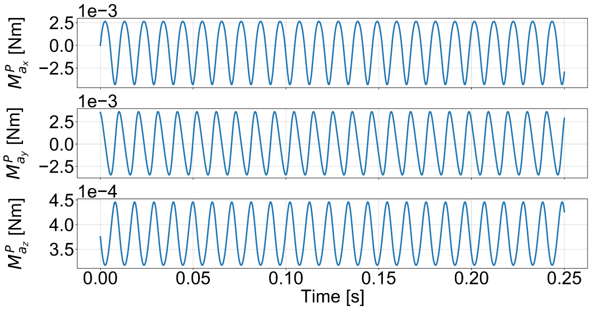


Fig. 20 BET-simulated evolution of aerodynamic moments generated by lost blade sections upon 20% Bebop 2 blade damage for 0.25 s rotating at $\omega_0 = 600$ rad/s.

Furthermore, Fig. 21 and Fig. 22 show the mass and aerodynamic caused forces and moments super-imposed. The mass change effects are predominant in the x- and y-components of the force, whereas the aerodynamic effects are predominant in the force z-direction and in all moment directions.

Finally, the mass and aerodynamic effects around the propeller's centre of rotation are combined as shown by Eq. (61) and Eq. (62); quantities that will be used in the validation phase. The combined effect for the discussed 0.25 s simulation leads to Fig. 23 and Fig. 24. Previous literature is correct in focusing on the mass related centrifugal forces, since they are one to three orders of magnitude greater than the rest. However, they ignore the effects outside the x-y plane in the propeller coordinate frame, especially the force in the z-direction. Additionally, although subtle, the oscillations in the moment signals could help in the identification of the blade damage.

$$\Delta\vec{F}^P = \vec{F}^P_{m1} + \vec{F}^P_{m2} - \vec{F}^P_{a1} - \vec{F}^P_{a2} \tag{61}$$

$$\Delta\vec{M}^P = \vec{M}^P_{m} - \vec{M}^P_{a1} - \vec{M}^P_{a2} \tag{62}$$

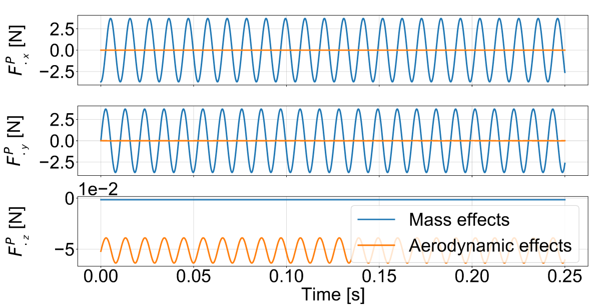


Fig. 21 BET-simulated evolution of mass and aerodynamic forces generated by lost blade sections upon 20% Bebop 2 blade damage for 0.25 s rotating at ω_0 = 600 rad/s.

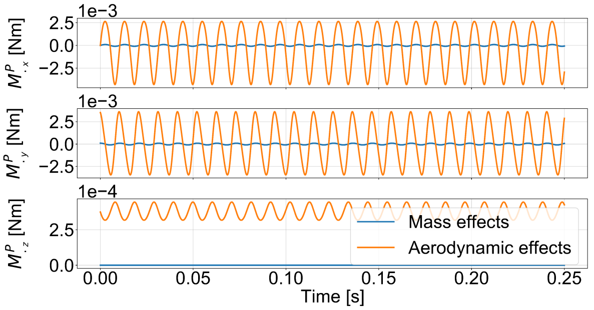


Fig. 22 BET-simulated evolution of mass and aerodynamic moments generated by lost blade sections upon 20% Bebop 2 blade damage for 0.25 s rotating at ω_0 = 600 rad/s.

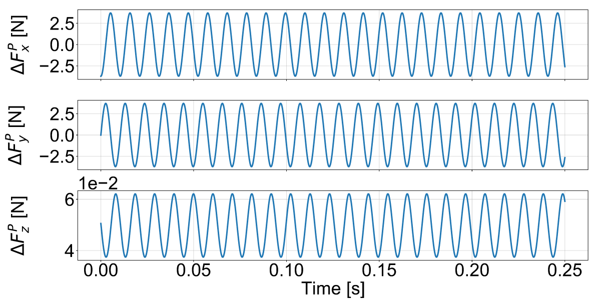


Fig. 23 BET-simulated evolution of forces upon 20% Bebop 2 blade damage for 0.25 s rotating at ω_0 = 600 rad/s.

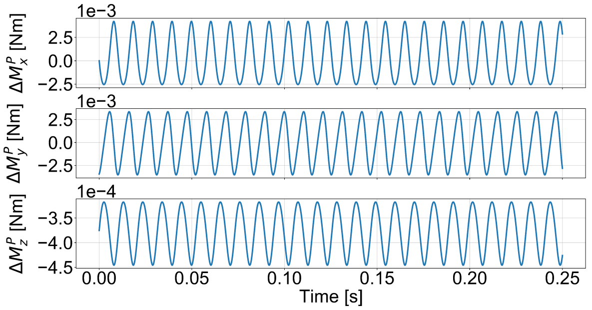


Fig. 24 BET-simulated evolution of moments upon 20% Bebop 2 blade damage for 0.25 s rotating at ω_0 = 600 rad/s.

Even though there is a different order of magnitude among the forces and the moments, all the signals are oscillatory. Figures 25 and 26 show the upper and lower limit of these wrench oscillations for different degrees of blade damage, namely from 0% (intact blade) to 100% (complete blade loss). All forces and moments have their upper and lower limits symmetric with respect to the 0 datum, except the force and moment in the z-direction. ΔF^P_z moves in the positive direction with increasing blade damage — the higher the blade damage, the larger the thrust loss (positive thrust points down in the propeller coordinate system). ΔM^P_z moves in the negative direction with increasing blade damage because the analysis is done on a counter-clockwise rotating propeller.

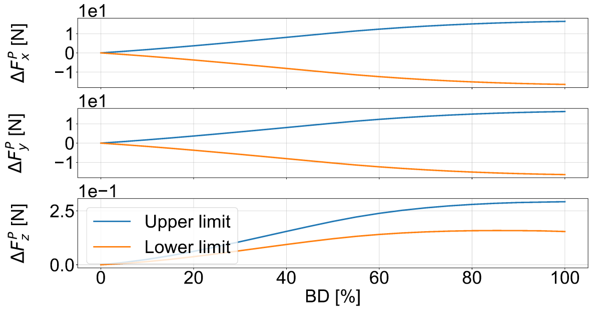


Fig. 25 Upper and lower limits of the forces' oscillations for different degrees of BET-simulated blade damage.

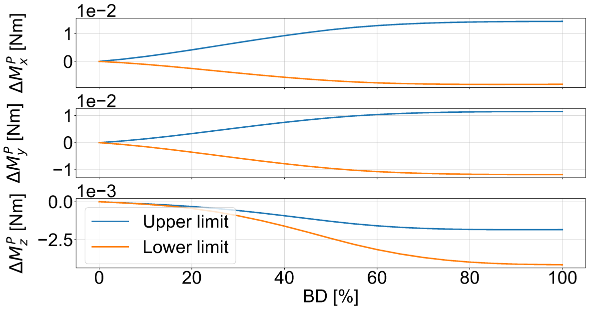


Fig. 26 Upper and lower limits of the moments' oscillations for different degrees of BET-simulated blade damage.

The higher the blade damage, the higher the oscillations. However, depending on whether the dominating effect is mass or aerodynamic, the behaviour of those limits is different. The gradient of the upper and lower limits of ΔF_x^P can be observed in Fig. 27; the gradient first increases and around 50% starts decaying. This is caused by the modelled double trapezoid Bebop 2 blade shape. As the blade is progressively damaged from the tip (0% damage) to the location of the central chord (50% damage) the removed blade sections are progressively growing in size, causing constantly greater shifts of the centre of gravity and, hence, greater increments in the centrifugal force. When the damage reaches the maximum chord, the removed blade sections start to decrease in size, leading to smaller centre of gravity shifts and more slowly growing centrifugal forces.

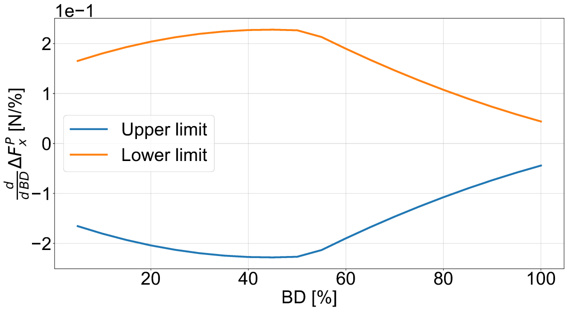


Fig. 27 Gradient of the upper and lower limits of the ΔF_x^P oscillations with respect to different degrees of BET-simulated blade damage.

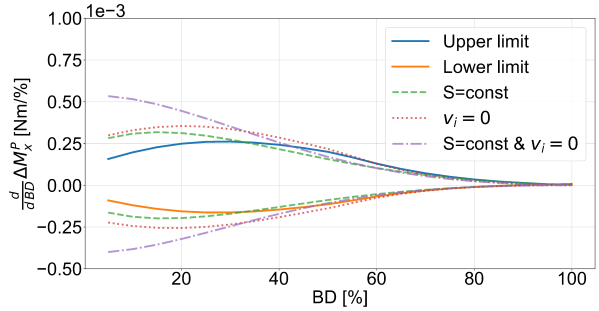


Fig. 28 Gradient of the upper and lower limits of the ΔM_x^P oscillations with respect to different degrees of BET-simulated blade damage. The non solid lines represent scenarios in which the blade section area is constant and/or there is no induced velocity.

Figure 28 depicts the gradients of the upper and lower oscillation limits of ΔM_x^P, an aerodynamic-dominated wrench component. As can be observed, the moment gradient increases in magnitude until around 30%, primarily due to the combined effect of the expanding blade section area from the tip to the central maximum chord and the induced velocity impacting the angle of attack. If the blade section area is held constant and the induced velocity is removed, oscillations only decay in magnitude. However, with at least one of these factors active, an initial gradient increment occurs.

Upon reaching 30% blade damaged, the most influential factor in the further gradient decay is the reduced distance from the blade section to the centre of rotation (r_{jk}). ΔM_x^P is proportional to r_{jk}^3 considering that at high propeller rotational rates, airspeed is dominated by $V_{A_{jk_2}}$ (Eq. (16)), a function of r_{ijk}. The airspeed is squared in the computation of ΔL_k and ΔD_k (Eq. (11) and Eq. (12)). These lift and drag components are then used for the computation of ΔT_{jk}, which is further multiplied by r_{jk} to obtain the moment.

C. Importance of induced velocity

The significance of including the induced velocity in the aerodynamic calculations outlined in subsection IV.A is evident when comparing Fig. 29 and Fig. 30; incorporating the linear induced velocity model reduces the amplitude and bias magnitude of the oscillations. This reduction is attributed to the decrease in the angle of attack resulting from the introduction of v_i, as shown in Fig. 31 and Fig. 32. These figures present box plots of the angles of attack seen by each of the 100 blade sections in the 16,000 scenarios used for identification of the lift and drag curves (subsection VI.A) with and without the linear induced velocity model, respectively. The following additional observations can be made:

1) Besides the general reduction in the angle of attack, the introduction of the induced velocity model shifts the linear part of the angle of attack with respect to the blade section box plot towards the blade tip. In contrast, when $v_i = 0$, the box plot is mostly linear. It is crucial to note that the linearity of the v_i model applies to the x and y directions in the propeller reference frame, as illustrated in Fig. 12, rather than along the blade radial direction.
2) Even though the twist of the blade goes from 25° to 5° from the root to the tip, the centre line of Fig. 31 has a lower value than the twist — especially close to the root — because the distribution of the linear body velocity in the z-direction is biased towards negative values (the drone is flying upwards). In subsection VI.A it was established that V_z^B has a value in the closed interval [-2,-0.5] m/s.
3) For both scenarios, the range of angles of attack is larger at the root due to the higher sensitivity to the vehicle's velocity; the blade section tangential velocity due to the propeller's rotation is lower at the root than at the tip.

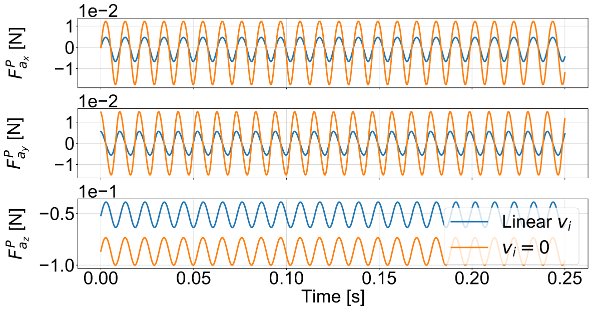


Fig. 29 BET-simulated evolution of aerodynamic forces generated by lost blade sections upon 20% Bebop 2 blade damage for 0.25 [s] rotating at ω_0 = 600 [rad/s] with and without linear inflow model.

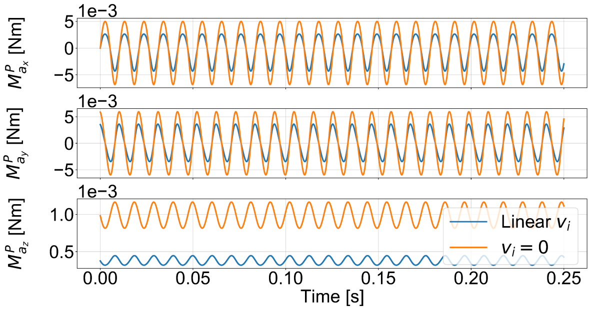


Fig. 30 BET-simulated evolution of aerodynamic moments generated by lost blade sections upon 20% Bebop 2 blade damage for 0.25 [s] rotating at ω_0 = 600 [rad/s] with and without linear inflow model.

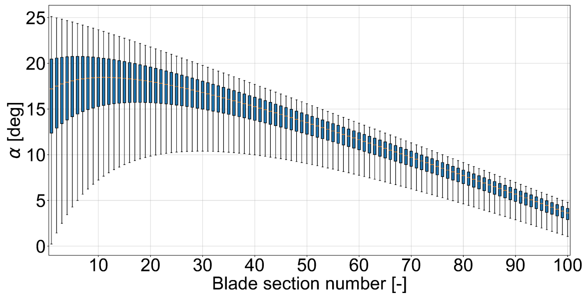


Fig. 31 Box plot with the angles of attack seen by each BET-simulated blade section during 16,000 data point optimisation without induced velocity model. The inputs that shape each data point are taken from uniform distributions with the following value ranges: $\overrightarrow{V}^B_x$ =[-3, 3] m/s, $\overrightarrow{V}^B_y$ =0 m/s, $\overrightarrow{V}^B_z$ =[-2.5, -0.5] m/s, $\overrightarrow{\Omega} = \overrightarrow{0}$ rad/s, ω =[300, 1256] rad/s.

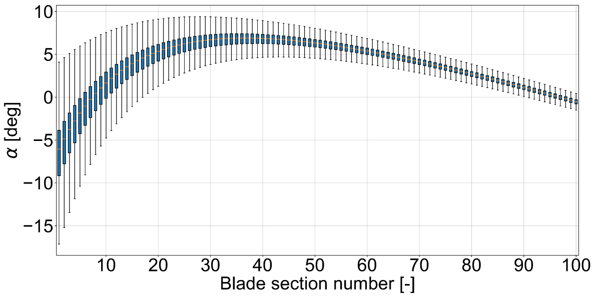
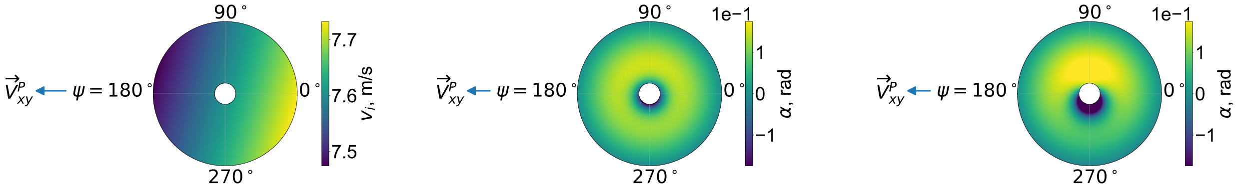


Fig. 32 Box plot with the angles of attack seen by each BET-simulated blade section during 16,000 data point optimisation with linear inflow velocity model. The inputs that shape each data point are taken from uniform distributions with the following value ranges: $\overrightarrow{V}^B_x$ =[-3, 3] m/s, $\overrightarrow{V}^B_y$ =0 m/s, $\overrightarrow{V}^B_z$ =[-2.5, -0.5] m/s, $\overrightarrow{\Omega} = \overrightarrow{0}$ rad/s, ω =[300, 1256] rad/s.

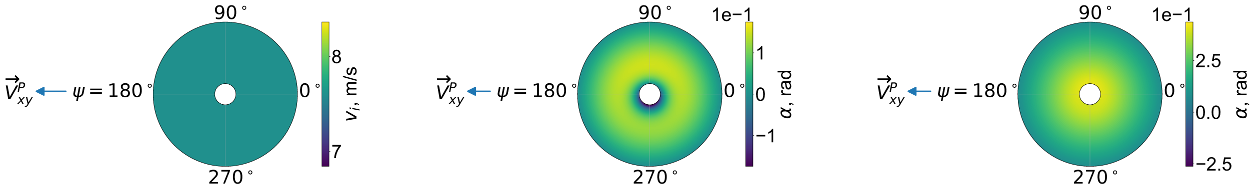
The value of the induced velocity and its effect on the angle of attack can also be visualised in the propeller plane, as illustrated in Fig. 33a and Fig. 33b for the Bebop 2 propeller located on the front left of the vehicle. The propeller is moving towards the left with 3 m/s, out of the plane with 1 m/s and it is rotating counter-clockwise. The empty internal concentric circle represents the propeller hub, which is not an aerodynamic surface, resulting in an annulus heat map. Figure 33b reflects the same behaviour as in Fig. 32 but in 2D, namely that the angle of attack rapidly increases close to the root until about 35% of the blade before it starts decaying more slowly towards the tip. The low angle of attack values close to the root in the direction of ψ =270° correspond to the retreating blade sections. When the rotational speed is lowered to 300 rad/s, the stalled retreating blade sections become more apparent, as shown in Fig. 33c.



(a) ω **=1256 rad/s** **(b)** ω **=1256 rad/s** **(c)** ω **=300 rad/s**

Fig. 33 Heat map of the linear induced model velocity and angle of attack for the BET-simulated front left Bebop 2 propeller moving to the left with 3 m/s, out of the plane with 1 m/s and it is rotating counter-clockwise.

Finally, Fig. 34a and Fig. 34b show the same induced velocity and angle of attack plots when a uniform induced flow is considered. As can be observed, the variations in both variables brought by the linear model corrections are very small, practically unnoticeable in the angle of attack when comparing the results with Fig. 33b. Whereas the uniform inflow model creates an induced velocity of 7.6 m/s across the complete blade, the linear inflow model creates an induced velocity that varies from 7.48 to 7.72 m/s. Even though literature has proven empirically that the linear inflow model is more accurate than the uniform counterpart [23], the difference can be considered negligible for propellers of small radius, as it is the case for most commercial drones such as the Bebop 2; a dependency on the blade radius which can be deduced from Eq. (21). The effect of the linear model can be observed in the main rotor system of helicopters which have blades longer than 1.5 metres.



(a) Uniform induced model **(b) Uniform induced model** **(c) No induced model**

Fig. 34 Heat map of the induced model velocity and angle of attack for the BET-simulated front left Bebop 2 propeller rotating at 1256 rad/s and moving to the left with 3 m/s, out of the plane with 1 m/s and it is rotating counter-clockwise.

Hence, the additional computations required for the uniform model corrections can be disregarded in simulation. However, including the uniform induced model is crucial, as it can modify the vertical airspeed seen by the blade sections from 1 to 8.6 m/s at full propeller rotational speed (1256 rad/s) for the Bebop 2 example scenario (Fig. 34a). This effect can be visualised when comparing the angle of attack heat map when there is a uniform induced velocity, as it is the case in Fig. 34b, to the scenario when $v_i = 0$ shown in Fig. 34c, the latter being a 2D representation of Fig. 31. Notably, the importance of the uniform inflow model is pronounced at the retreating blade sections near the root.

D. Assumptions and recommendations

For the development of the blade damage forces and moments model, as well as the lift and drag coefficient curves identification, the following assumptions were made:

1) The mass along the blade is homogeneous, meaning that the centroid equals the location of the centre of gravity.
2) The Bebop 2 blades are simplified as two trapezoids with parallel sides connected by the long parallel side.
3) The twist decreases linearly from the root to the tip.
4) The airfoil is constant throughout the blade.
5) The cross flow along the span of the blade is ignored.
6) Aeroelasticity effects are ignored.
7) The blade root and tip lift losses are ignored.
8) The induced velocity is computed with the simplified linear induced inflow.
9) The nonlinear aerodynamic effects between (damaged) blades are not considered.
10) The nonlinear aerodynamic effects between propellers are not considered.
11) The nonlinear aerodynamic effects between the propellers and the body frame are not considered.
12) The data used for the identification of the lift and drag coefficient curves were obtained from the aerodynamic grey-box model [12] that provides the propeller thrust and torque. Hence, the present work adopts the assumptions taken for the development of this model.
13) The blade is cut parallel to the edge of the propeller, perpendicular to its span, such that the remaining polygon is still a trapezoid. Hence, slanted or irregular cuts are not considered.

Further work in the simulation of propeller damage could be oriented towards the refinement of the model developed in this chapter in order to remove one or multiple of the aforementioned assumptions; contributing to its generalisation and application to different propeller types. For instance, the geometrical assumptions 1)-4) could be eliminated by creating a 3D model (digital twin) of the propeller using scanning technologies that probe the propeller through physical touch (with contact), such as Coordinate Measuring Machines [30], or scanning technologies that exploit acoustic, optical or magnetic approaches (without contact), such as laser scanning, structured light or photogrammetry [31, 32] (e.g. structure from motion). If translated to a CAD model, this would allow the computation of the twist, chord and volume of each blade section, the latter being used for the computation of the centre of gravity when the density of the material is known. Additionally, such model would contribute to the potential discovery of multiple airfoils present in the blade. If that would be the case, the parameter vector of Eq. (47) would be expanded with the polynomial coefficients used to identify the lift and drag coefficient curves of those additional airfoils.

Assumptions 5)-11) are related to the degree of aerodynamic complexity introduced in the model. In particular, assumption 6) points out that aeroelastic effects have been ignored. Most literature in this regard is oriented towards the modelling of helicopter aeroelastic and blade flapping behaviour [33]. Unfortunately, this knowledge is not directly applicable to drones given that helicopters have a horizontal hinge, also known as flapping hinge, which allows the blade to be displaced up and down to compensate for the rotor lift dissymmetry [23]. Instead, commercial drone rotors lack an articulated head, causing their material to bend and the rotor to tilt with the possibility of flapping [34]. As an alternative, the field of wind energy could be explored since Blade Element Momentum Theory approaches have been used as the aerodynamic component of wind turbine aeroelastic models [35]. However, given the circular dependency between the blade deformations (aeroelastic effects), the induced velocity, and the generated moments and forces, the authors consider such implementation to be challenging for real-time simulations.

Assumption 7) mentions that the blade root and tip losses were ignored. At those blade locations, the circulation must be equal to zero and at the tip there is an additional reduction of lift due to the appearance of tip vortices — airflow around the tip due to the pressure difference between the pressure side (high pressure) and the suction side (low pressure). In the field of wind energy [19], these effects are taken into account by multiplying the induced velocity with a correction factor that is a function of the distance to the centre of rotation (r). This factor would acquire a value of 1 in the centre of the blade and a value of 0 at the edges, allowing the induced velocity to fall to zero at the blade edges. Alternatively, previous literature [19, 23, 36] has also proposed the Prandtl tip-loss factor approximation (B=0.95-0.98) to compute the effective blade radius (R_e = BR) and account for the loss of blade lift. As a result, the outer portion of the blade ($R-R_e$) is considered to be incapable of carrying lift. Given that in helicopter aerodynamics the introduction of the tip loss factor can cause rotor thrust reductions between 6-10% [23], the study of its implementation in drone propellers is recommended for the further improvement of the BET thrust and torque predictions.

Regarding the induced velocity model used (assumption 8)), a comprehensive benchmark study of the different induced inflow models applied to drone propellers is missing in the current literature and it could be considered a line of further work. It is recommended that future studies investigate the suitability and accuracy of the inflow models of Mangler and Squire [37, 38] and Ormiston [39, 40]. The former associates the pressure field across the rotor disk to the inflow with the incompressible, linearised Euler equations. This method originally requires to solve for the rotor loading (Δp) using BET, an approach which is computationally expensive when compared to the linear induced inflow model that optimises the induced velocity before the first BET iteration.

Other interesting approaches to consider include linear inflow models, such as those from Payne [41] and Pitt & Peters [25], as well as Pitt-Peters [42] and Peters-He [43, 44] dynamic inflow models. The last two approaches have been consolidated and broadly used in the field of rotorcraft dynamics because they exploit unsteady actuator disc theory for hover and forward flight. Instead of ignoring wind-speed fluctuations by averaging the wind field (frozen wake model) or assuming that the instantaneous wind velocity corresponds to that of steady-flow conditions (equilibrium wake model), these dynamic inflow models accurately describe the wake behaviour by assuming the existence of a delay before the induced inflow reacts to modifications in the wind field (unsteady-flow) [19]. Additionally, they are both represented in state-space form, which could be implemented and solved in real-time simulations, and there exist augmentations to their original formulations which include wake distortion effects during manoeuvring flight [45]. Although vortex methods offer higher accuracy, their computational cost is prohibitive for online blade damage simulations [23, 46].

To address non-linear inter-propeller, inter-blade or body-blade interactions (assumptions 9)-11)), the creation of a data-driven model that provides the highly nonlinear lift and drag contributions of each blade section that are not encapsulated in the BET model, is recommended. Similar work that could serve as inspiration is carried out within the field of aerodynamics, discipline in which turbulence is modelled for Reynolds-Averaged Navier-Stokes computations using artificial intelligence [47, 48] (data assimilation for CFD closure). In this approach, physics is exploited for simulating large scale flow behaviours, whereas machine learning, a mostly black-box approach, is used for modelling the highly nonlinear lower scale turbulence using experimental data. This method is valued for its low computational cost when compared to higher fidelity but more expensive simulations, such as Direct Numerical Simulations.

Finally, the grey-box aerodynamic model [28] is a data-driven identification approach with physical and semi-physical parameters. Its parametric model structure, namely a piecewise polynomial, is variable since components have been added and removed according to a stepwise selection scheme depending on their contribution to the model accuracy.

Beyond its structure, the main model assumption derives from the identification of its parameters with wind tunnel data obtained in quasi-steady flow conditions; there is no rate of change of velocity with time at a single point in the test section volume but the vehicle states, such as the angle of attack, constantly change due to its circular flight motion. It does not enter the unsteady-aerodynamic flow regime because effects caused by the changing circulation and wake on the aerodynamic surfaces are not considered. Future research that would aim to use the developed fault detection and identification framework "in the wild" under the presence of wind field changes, drastic manoeuvres, gusts and turbulence would require the revision of this assumption.

VII. Model validation

The model was validated by comparing the thrust and torque signals measured in an experimental set-up to those predicted by the BET model given the same input conditions ($\overrightarrow{V}^B$, $\overrightarrow{\Omega}$, ω). For that purpose, an experimental campaign was carried out in the Open Jet Facility wind tunnel at the Faculty of Aerospace Engineering at TU Delft.

A. Test set-up and data collection

The OJF is a wind tunnel with an octagonal open test section of 2.85 metres in width and height through which the air flows into a room with a width of 13 metres and a height of 8 metres. The maximum wind speed that can be reached is 35 m/s. For the measurement of thrust and torque, the Series 1580 test stand from Tyto robotics was used. This is a dynamometer for drone propulsion systems capable of measuring up to 5 kgf of thrust and 2 Nm of torque, as well as voltage, current, power, motor rotational speed and vibrations. Figures 35 and 36 show the test stand from the side and top, highlighting its most important components.

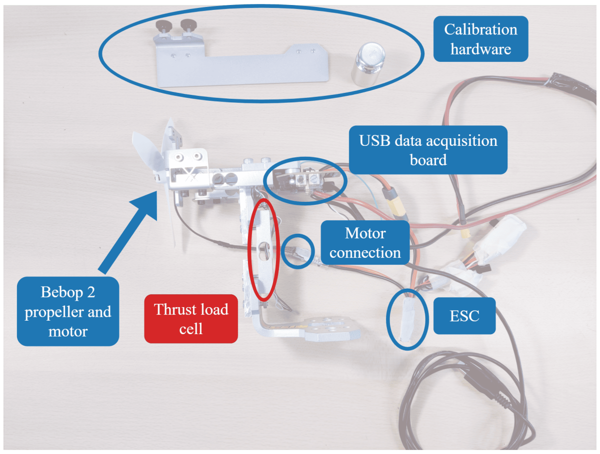


Fig. 35 Tyto stand: side view with calibration hardware.

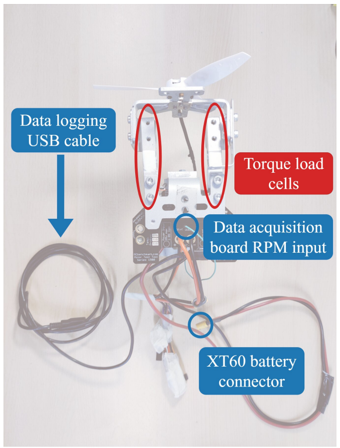


Fig. 36 Tyto stand: top view.

Table 2 displays the modified parameters and their value ranges, each potential combination forming a scenario. For the experimental campaign, a single blade was cut at a time with damage represented by BD percentage, capped at 25% due to excessive vibrations on the load cell at higher values. The damage involved a perpendicular cut to the blade's span, illustrated in Fig. 37 and Fig. 38 for 10% and 25%, respectively. Furthermore, as explained in subsection VI.A, the BET model utilises drone linear velocity $\overrightarrow{V}^B$, angular velocity $\overrightarrow{\Omega}$ and propeller speed ω to predict thrust and torque. $\overrightarrow{V}^B$ is simulated as the negative wind speed vector and, since the test stand remains stationary during measurements, $\overrightarrow{\Omega}$ is set to zero. The wind speed vector is decomposed into magnitude V_∞ and the propeller incidence angle i_p (Fig. 39).

Figures 40 and 41 depict a beam assembly used to minimise the local influence of the test platform on the free-stream flow. This setup enables positioning the test stand within the wind tunnel airflow, thereby mitigating wind tunnel wall effects on the flow field.

Thrust and torque data were recorded at 7Hz for 20 seconds per scenario, totaling 630 scenarios and a 3.5-hour data collection phase. During that time, the following challenges were encountered that may have impacted the reliability of the results and conclusions derived from the collected data:

Table 2 Experimental campaign testing parameters and values.

Parameter	Unit	Values
BD	%	0, 10, 25
i_p	°	0, 15, 30, 45, 60, 75, 90
V_∞	m/s	0, 2, 4, 6, 9, 12
ω	rad/s	300, 500, 700, 900, 1100

Fig. 37 Damaged propeller with BD=10%.

Fig. 38 Damaged propeller with BD=25%.

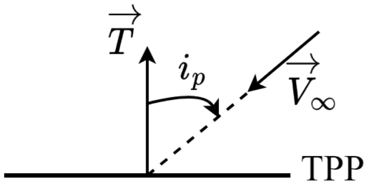


Fig. 39 Propeller incidence angle.

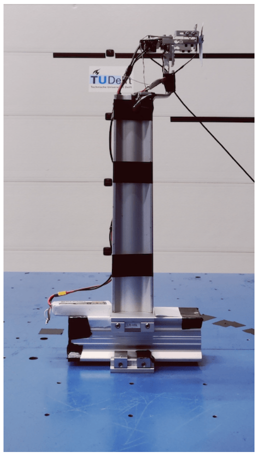


Fig. 40 Test set-up in the wind tunnel.

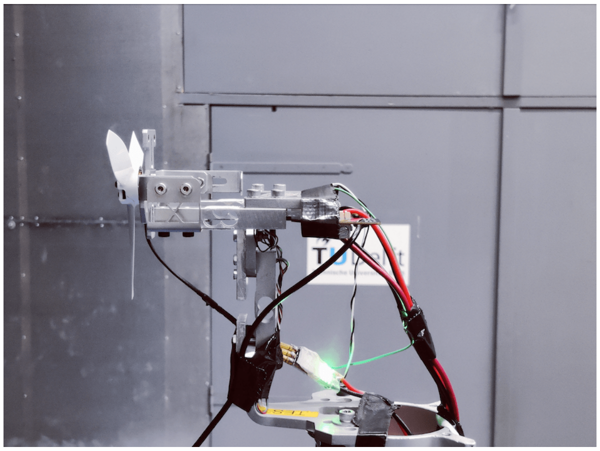


Fig. 41 Tyto test stand in the wind tunnel.

1) Even though the test stands measures accelerometer and propeller rotation values at around 100Hz, thrust and torque are measured at 6-7Hz. This is insufficient for signal reconstruction, specially in cases of blade damage where the observed sinusoids in section VI have a frequency of 50Hz-175Hz, depending on the propeller rotational speed. These oscillations far exceed the test stand wrench sampling Nyquist frequency of 3.5Hz. Even though larger test stands for measuring wrenches exist, designed for larger propellers, their sensitivity or accuracy for small drone propellers may be insufficient, especially when anticipating the observation of small vibrations on the order of 10^{-4}.
2) During the execution of the experiments it was encountered that the test stand resonated with the vertical beam and/or the platform at certain rotational frequencies. This resulted in the observation of peaks in the rpm and wrench measurement signals.
3) The 5kg thrust load cell was not able to withstand prolonged vibrations at 25% blade damage or survive a higher degree of blade damage for more than a few seconds.

B. Experimental data pre-processing

The data pre-processing phase has 2 steps. First, the data were corrected by adding the effect of the wind on the test stand. The thrust and torque are measured without a propeller installed at different wind speeds and angles with respect to the incoming flow and the mean measured wrenches are then added to each data point accounting for their V_∞ and i_p.

Second, in the case of blade damage it was observed that the forces and moments were increasing with time even though the parameters in Table 2 were kept constant. This is considered sensor error and the signals are detrended.

C. Experimental results

First, subsubsection VII.C.1 will present the results without blade damage followed by subsubsection VII.C.2 which will examine those in the presence of 10% and 25% blade damage.

1. Without blade damage

This section analyses the impact of the last three input parameters from Table 2 on the BET model's performance without blade damage (BD=0%). In contrast to scenarios with blade damage, a comparison between BET's performance and the grey-box (healthy) aerodynamic model is feasible. First, the effect of the incidence angle on the results is studied by maintaining constant the wind and the propeller rotational speeds while varying i_p from 0° to 90°. Figures 42 and 43 illustrate the results for thrust and torque at V_∞=2 m/s and ω=700 rad/s, respectively.

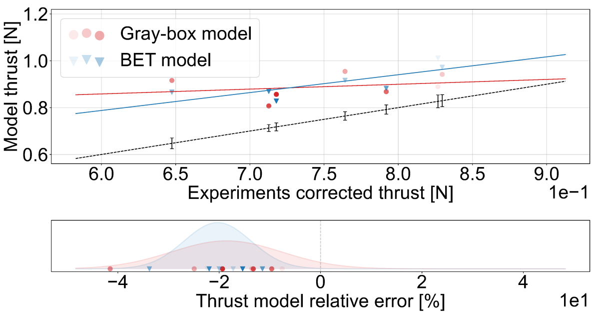


Fig. 42 Experimental and model thrust measurements and their relative error for: BD=0%, V_∞=2 m/s and ω=700 rad/s. The black dashed line represents the ideal scenario in which the model and experimental thrust would match.

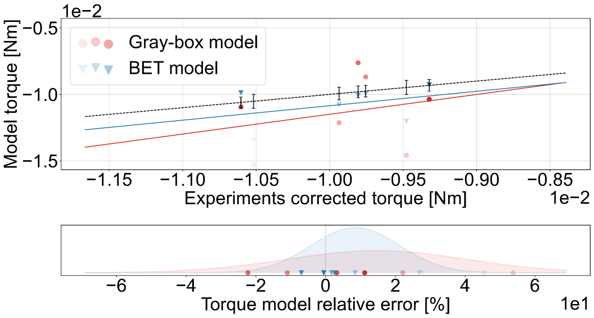


Fig. 43 Experimental and model torque measurements and their relative error for: BD=0%, V_∞=2 m/s and ω=700 rad/s. The black dashed line represents the ideal scenario in which the model and experimental torque would match.

The upper window of each plot presents the values obtained in the experimental campaign (x-axis) with respect to the values obtained by each of the models (y-axis) given the same conditions in terms of $\overrightarrow{V}^B$, $\overrightarrow{\Omega}$ and ω. In an ideal scenario, both models' data points would fall on the dashed black line, meaning that both experimental and simulated results are equal. Unfortunately, that is mostly not the case and the solid red and blue lines represent the linear fit of each model's data. Besides that, the data points are plotted with different degrees of transparency. The degree of transparency varies linearly from the brightest or most opaque markers representing those data points measured at i_p=0° to the most transparent ones representing those data points measured at i_p=90°. Finally, the whiskers represent the range of values in which 95% of the experimental wrench data samples can be found ($\approx 2\sigma$). For the computation of these value ranges, the standard deviation of the forces and moments exerted by the wind on the test stand (subsection VII.B) were included. The lower window of each plot shows the wrench relative error of each model with respect to the experimental measurements, as well as a fitted Gaussian curve to those relative error data points.

As expected, the measured thrust decreases with decreasing i_p because the blade element angle of attack is decreased due to a higher wind speed perpendicular to the plane of rotation. Furthermore, it can be observed that the thrust is always positively biased and the torque is negatively biased for both models. This indicates that there exist some unmodelled physical effects that have not been taken into consideration, among which might be those outlined in subsection VI.D. Additionally, these plots show that the performance of the BET and grey-box models is very similar, which supports the hypothesis that the BET model has been well identified and that the errors are due to those in the grey-box model whose data were used for identification.

After having seen the effect of varying i_p, the next step would be the analysis of the influence of the propeller rotational speed. All the results for different values of ω can be synthesised in Fig. 44 by plotting the mean of the relative error for each propeller rotational speed with whiskers representing 1.96 times its standard deviation. Both the mean and the standard deviation were obtained from the Gaussian curves in the lower window of plots identical to Fig. 42 and Fig. 43 for different values of ω. Three conclusions can be derived for the BET and grey-box aerodynamic models comparison. First, the BET model is more accurate in torque but less in thrust. Second, the BET is more (over)confident of its predictions because of its smaller confidence intervals. The latter observation is also visible in Fig. 42 and Fig. 43 due to the taller and narrower Gaussian fitted curves for the BET model. Third, model accuracy increases with ω for both models.

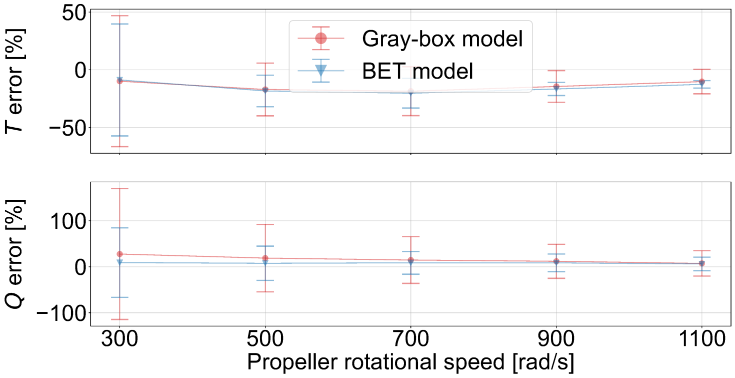


Fig. 44 BET and grey-box aerodynamic model thrust and torque relative error for: BD=0% and V_∞=2 m/s.

Finally, in order to observe the effect of the increasing wind speed, Fig. 44 is repeated for each wind speed listed in Table 2 within the same plot. This is shown in Fig. 45 and Fig. 46 for the grey-box aerodynamic and BET models, respectively. Note that the whiskers representing the confidence intervals have been removed for clarity. From these plots, four observations can be made:

1) The performance of both thrust and torque degrades with increasing wind speed for both models.
2) The relative thrust error of the BET model has a sudden increase when the wind speed is 6 m/s or higher when compared to the grey-box aerodynamic model.
3) The BET model performs better than the grey-box aerodynamic model in terms of torque except at 12 m/s.
4) The performance of the grey-box aerodynamic model at a wind speed of 12 m/s for thrust and higher than 4 m/s for torque is very low (sometimes with relative error values above 1000% for torque).

The reason behind the first three differences in performance between models originates from a design choice in subsection VI.A, namely that the BET model airfoil lift and drag coefficients were identified with wind speeds up to 3.6 m/s. Hence, the BET model has never seen data collected at wind speeds higher than 4 m/s. The last observation is unexpected as the grey-box aerodynamic model was identified with data gathered at wind speeds up to 14 m/s.

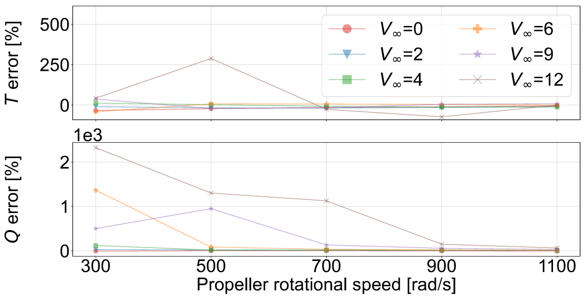


Fig. 45 Grey-box aerodynamic model thrust and torque relative error for BD=0%.

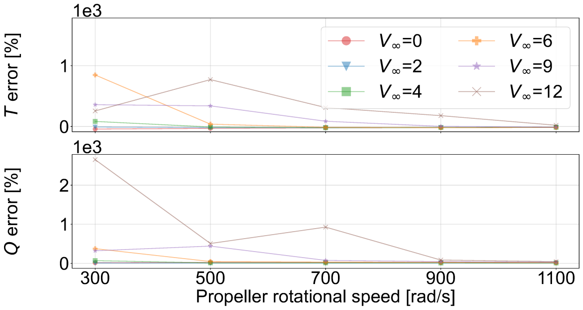


Fig. 46 BET model thrust and torque relative error for BD=0%.

One general conclusion that can be derived from these observations is that the BET model architecture has a stronger physical foundation for torque than for thrust. Both were identified with data collected at wind speeds lower than 3.6 m/s and the torque is able to perform better at those speeds that the model had not seen before during identification, namely 4, 6, 9 m/s, when compared to the thrust. In most cases, it even performs better than the grey-box aerodynamic model that was used for the identification data generation. This highlights that the unmodelled aerodynamic effects have a stronger impact on the thrust than on the torque.

2. *With blade damage*

In the presence of blade damage, two signal features need to be validated, namely the bias of the signal and the amplitude of the damage induced oscillations. Similar to subsubsection VII.C.1, a sensitivity analysis will be conducted, examining the impact of parameters in Table 2 with 10% and 25% blade damage. Additionally, the BET model's performance with blade damage will also be compared to that presented in subsubsection VII.C.1 without blade damage. The latter serves as baseline since its error can be associated to the identification dataset, and not to the BET model's architecture (the object of validation of the present section). Finally, there is no control signal from the grey-box (healthy) aerodynamic model in the presence of damage.

To assess the impact of varying i_p, Fig. 47 to 50 depict thrust and torque plots for both blade damages at V_∞=2 m/s and ω=700 rad/s. In contrast to the scenario without blade damage, at BD=25%, thrust and torque are not consistently positively and negatively biased. This outlying behaviour — attributed to the resonance between the test stand, vertical beam and platform, as discussed in subsection VII.A — is evident only in the thrust plots when ω=500 and 700 rad/s.

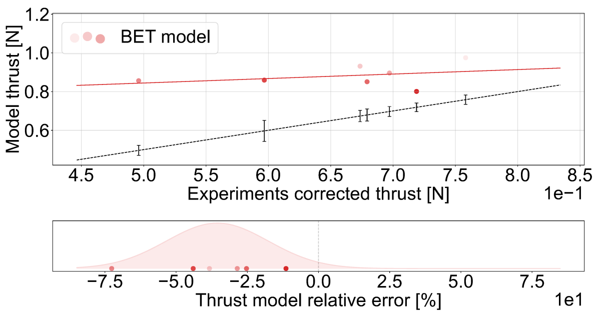


Fig. 47 Experimental and model thrust measurements and their relative error for: BD=10%, V_∞=2 m/s and ω=700 rad/s. The black dashed line represents the ideal scenario in which the model and experimental thrust would match.

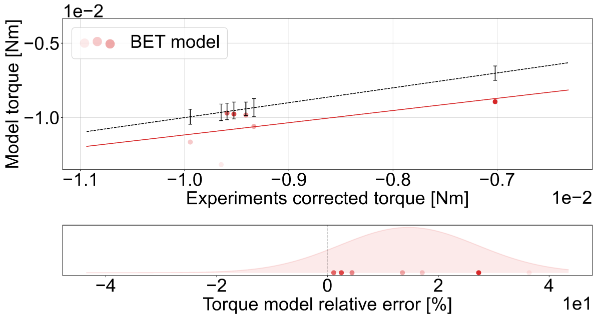


Fig. 48 Experimental and model torque measurements and their relative error for: BD=10%, V_∞=2 m/s and ω=700 rad/s. The black dashed line represents the ideal scenario in which the model and experimental torque would match.

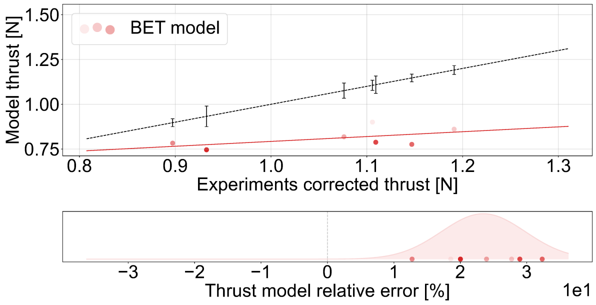


Fig. 49 Experimental and model thrust measurements and their relative error for: BD=25%, V_∞=2 m/s and ω=700 rad/s. The black dashed line represents the ideal scenario in which the model and experimental thrust would match.

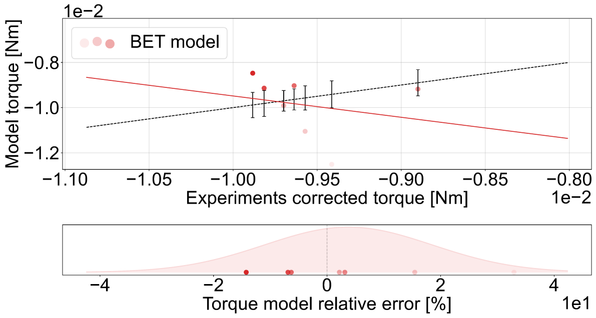


Fig. 50 Experimental and model torque measurements and their relative error for: BD=25%, V_∞=2 m/s and ω=700 rad/s. The black dashed line represents the ideal scenario in which the model and experimental torque would match.

To assess the impact of ω on the results, Fig. 51 and Fig. 52 show the relative error for each value of ω for both degrees of blade damage. With BD=10%, the relative error follows a pattern similar to the undamaged scenario: mean and standard deviation decrease with increasing ω. However, at BD=25%, that pattern is disrupted, with the mean even crossing the x-axis for thrust between 500 and 700 rad/s. Again, this anomalous behaviour is attributed to the strong vibrations observed during the experimental campaign at those rotational speeds and blade damage values.

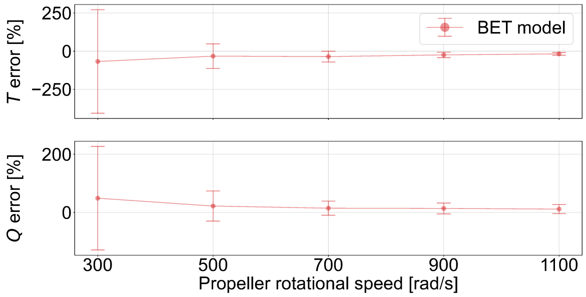


Fig. 51 BET model thrust and torque relative error for: BD=10% and V_∞=2 m/s.

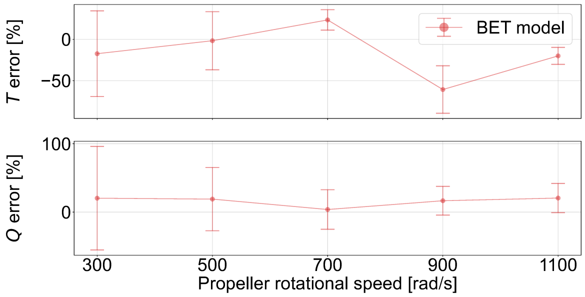


Fig. 52 BET model thrust and torque relative error for: BD=25% and V_∞=2 m/s.

To better assess the BET model architecture error in scenarios with blade damage, Fig. 53 shows the same curves for BD=10% and 25% with the error when BD=0% — identification data related error — subtracted (Δerror). Two observations emerge. First, performance for BD=10% is worse at low ω values up to 700 rad/s than BD=25%. Second, BD=10% performance improves with ω, whereas BD=25% performance does not show a clear pattern. Table 3 details the Δerror range for both blade damages in thrust and torque.

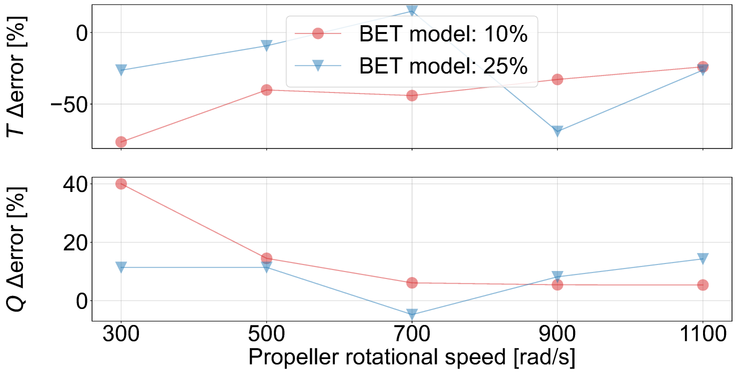


Fig. 53 BET model thrust and torque relative error for BD=[10, 25]% and V_∞=2 m/s, after subtracting the relative error when BD=0%.

Table 3 Δerror ranges for thrust and torque for 10% and 25% blade damage at 2 m/s wind speeds.

	***BD*=10%**	***BD*=25%**
T Δerror [%]	[-76.9, -23.6]	[-68.5, 15.8]
Q Δerror [%]	[5.1, 39.8]	[-4.9, 14.5]

After varying i_p and ω, the last parameter adjusted is V_∞, as shown in Fig. 54 and Fig. 55 for 10% and 25% blade damage. Again, a sudden performance decline is observed beyond 4 m/s wind speeds in both scenarios, phenomenon attributed to the BET model identification data limited to 3.6 m/s.

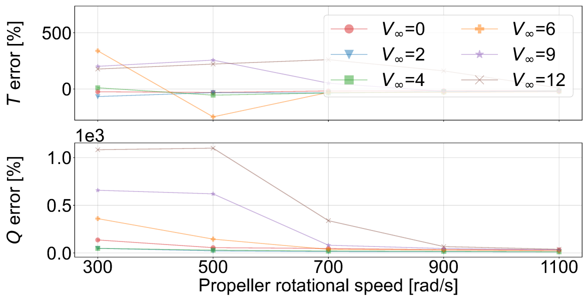


Fig. 54 BET model thrust and torque relative error for BD=10%.

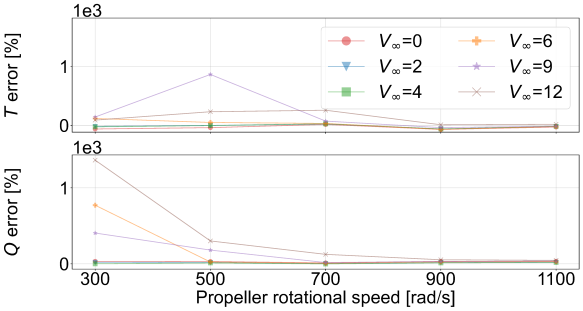


Fig. 55 BET model thrust and torque relative error for BD=25%.

Finally, validating damage-induced oscillation amplitudes is challenging due to the low sampling frequency for reliable signal reconstruction. Two attempted approaches, an evolutionary algorithmic method with Particle Swarm Optimisation and a statistical approach using the Lomb-Scargle periodogram [49–51], aimed to fit a sinusoid matching the propeller's oscillation frequency to the experimental data, but both were deemed unfit for the challenging task.

VIII. Conclusion

This paper proposes a white-box model for blade damage simulation which combines the effects caused by the shift of the centres of gravity and pressure. Mass effects were modelled by discretising the propeller into trapezoids whereas the aerodynamic effects were derived from first principles exploiting Blade Element Theory (BET). Additionally, a BET-based methodology for identifying 2D aerodynamic properties of UAV propellers, specifically airfoil lift and drag curves as functions of the angle of attack, was presented. Such information is currently unavailable for most commercial off-the-shelf UAVs.

The presented model has three key advantages. First, it does not require additional costly experimental wind tunnel campaigns for the blade damage modelling. Second, it enables the simulation of any degree of blade damage instead of being limited to a discrete number of failure scenarios within a safe flight regime. Third, it is complementary to existing healthy UAV models and can be used as a plug-in to extend its range of operations to damaged cases.

The methodology was applied to the Bebop 2 drone, leveraging on the available grey-box aerodynamic model of the chosen platform [12] to build the BET model. From the results it was concluded that previous literature was correct in claiming that the centrifugal forces due to the shift in the centre of gravity were dominant. However, they ignore the loss of weight and the aerodynamic effects, which are not negligible especially at high degrees of blade damage and propeller rotational speeds. The main concern are the neglected forces which, depending on the drone's geometry, could lead to large moments around the UAV's CG. Those oscillations could help in the identification of blade damage and ignoring them could render fault-tolerant control approaches unsuccessful when deployed in the real world. Furthermore, the results showed that the induced velocity is an essential parameter in the model. However, due to the small propeller size of most UAVs, the corrections brought by the linear inflow model over the simpler uniform baseline are negligible and can be ignored in simulation.

To validate the proposed approach, its predicted forces and moments were compared to those obtained from wind tunnel experiments. They were conducted at the Open Jet Facility at Delft University of Technology with a Bebop 2 propeller mounted on a static test stand while four parameters were varied, namely the degree of blade damage (BD), the propeller incidence angle (i_p), the wind speed (V_∞) and the propeller rotational speed (ω). In the scenarios without blade damage, it was possible to compare the BET model performance with that of the grey-box aerodynamic model. The performance of both is very similar, indicating that the BET model has been well identified and its validation errors are attributed to the identification data gathered from the grey-box aerodynamic model, instead of the BET model architecture. Besides that, the thrust is positively biased and the torque is negatively biased for both models. This remark points to the existence of unmodelled physical effects, among which might be those outlined in subsection VI.D. Additionally, the performance of both models decreases with lower values of ω and higher values of V_∞. This shows that both approaches struggle to correctly model blade sections under a negative angle of attack; a phenomenon mostly emergent in those ω-V_∞ conditions for blade sections close to the propeller hub.

Despite their similarities, three differences emerged between the models. First, the BET model demonstrates higher confidence in its predictions than the grey-box aerodynamic model. Second, BET exhibits a faster decline in thrust performance for wind speeds exceeding 4 m/s. This can be attributed to the design choice of airfoil lift and drag coefficient identification using data with wind speeds up to 3.6 m/s, preventing exposure to data beyond 4 m/s. Third, while BET is slightly more accurate in torque, it lags in thrust accuracy, emphasising the greater impact of unmodelled aerodynamic effects on thrust. Besides these differences, the grey-box aerodynamic model's unexpectedly low performance at high speeds is noteworthy, given its identification with wind tunnel data gathered up to 14 m/s [12].

Validation of damage-induced wrench oscillations in scenarios with blade damage requires analysing the bias and amplitude. A comparison between experimental and BET model signal biases indicates that BD=10% behaves similarly to that of BD=0%. In contrast, when BD=25% the relative error does not decrease with ω as it would be expected. This outlying behaviour, especially noticeable when ω=500 and 700 rad/s, is attributed to the resonance of the test set-up. Overall, in the presence of the tested blade damage, BET's performance at high propeller rotational speeds is similar to that without blade damage, with relative errors fluctuating between 5% and 24%. However, errors at low speeds can be more than 3 times higher, ranging from 15% to 75%. For the validation of the oscillations' amplitude, two approaches were implemented for signal reconstruction, namely Particle Swarm Optimisation (metaheuristic evolutionary optimisation algorithm) and the Lomb-Scargle periodogram (statistical algorithm). Unfortunately, reliable assessment was hindered by load cell noise and challenges in reconstruction attributed to test stand low sampling rates.

Besides the recommendations derived from the model assumptions and outlined in subsection VI.D, the authors recommend designing an in-house test stand for measuring partially damaged propeller's wrenches. The proposed stand would require a sampling frequency exceeding 100 Hz for measurements at the minimum rotational speed of 300 rad/s (or 350 Hz for ω=1100 rad/s), a dampening system to prevent resonance with the testing platform, and especially designed load cells capable of withstanding at least 6g of sustained vibrations. In addition, characterising the complete test set-up to identify its dynamics could contribute to the removal of the resonance present in the measurements. Knowledge about the complete system behaviour enables a deeper understanding of the measured signals and the separation of the set-up dynamics from the "pure" thrust and torque oscillations caused by the damaged propeller.

To conclude, the authors hope that the outlined lessons will serve as basis in the design of a future experimental campaign with more specialised hardware. The developed "plug-in" BET model with its future work aspires to become an indispensable cost-effective tool for researchers when designing and testing their work to build more resilient UAVs against blade damage in a wide range of fields, from fault detection and diagnosis to fault-tolerant control.

Appendix

As outlined in section IV, v_0 computation requires a numerical approach, vital in every simulation time step following propeller damage. Consequently, a tailored computationally efficient gradient-descent approach is here developed.

In order to define the optimisation problem objective function, the alternative V_R definition in Eq. (63) is used in conjunction with the Glauert formula in Eq. (19). In contrast with Eq. (20), the new definition of V_R translates the 3 components of the linear velocity of the propeller assembly ($\overrightarrow{V}^P$) into 2 components, namely its magnitude (V) and the angle of attack of the rotor disk relative to the oncoming flow (α_d). The latter is illustrated in Fig. 56.

$$V_R = \sqrt{(V\cos\alpha_d)^2 + (V\sin\alpha_d + v_0)^2} \tag{63}$$

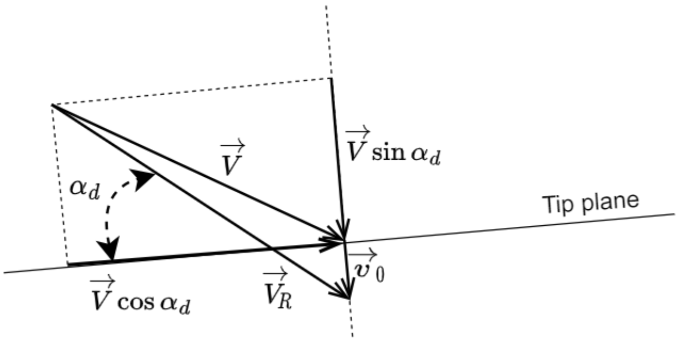


Fig. 56 Angle of attack of the rotor relative to the oncoming flow.

The optimisation problem objective function can be defined as described in Eq. (64), which is the same as finding the location where the function $f(v_0)$ intersects the x-axis.

$$\min_{v_0} \quad |f(v_0)| = \left| T - 2\rho\pi R^2 v_0 \sqrt{(V\cos\alpha_d)^2 + (V\sin\alpha_d + v_0)^2} \right| \tag{64}$$

Gradient-descent methods are used in optimisation for finding the local minimum of a differentiable function by traversing the solution space in the opposite direction of the function gradient, also known as the direction of steepest descend. In the case of the present objective function, local minima will be found where the derivative of $f(v_0)$ with respect to v_0 is zero and where $f(v_0) = 0$. In the case that it can be proven that the function $f(v_0)$ is strictly monotonic, meaning that it only increases or decreases, then $f(v_0)$ will not have local minima and it will be zero at a single value of v_0. Then, there exists a single (global) minimum in the objective function and a gradient-descent approach could be used to find it. Given the definition of the Glauert formula (Eq. (19)), the uniform induced velocity can only have a positive value. Hence, it is only required to prove the strict monotonocity for v_0 values in the half-open interval $[0, \infty)$.

Equation (65) shows the derivative of $f(v_0)$ with respect to v_0 and Eq. (66) shows the uniform induced velocity values that make it zero. As can be seen, $f(v_0)$ has one or two optima when $9\sin^2\alpha_d - 8 \geq 0$. Since the uniform induced velocity can only be positive, the only solutions that disprove $f(v_0)$ monotonicity comes from negative α_d angles, ergo when the condition in Eq. (67) is met. When the angle of attack of the rotating disk is higher than $\arcsin -2\sqrt{2}/3$, $f(v_0)$ is strictly monotone and gradient-descent would be able to find the global minimum.

$$\frac{df(v_0)}{dv_0} = -2\rho\pi R^2 \left(\sqrt{(V\cos\alpha_d)^2 + (V\sin\alpha_d + v_0)^2} + v_0 \frac{V\sin\alpha_d + v_0}{\sqrt{(V\cos\alpha_d)^2 + (V\sin\alpha_d + v_0)^2}} \right) \tag{65}$$

$$v_0 = \frac{V}{4}\left(-3\sin\alpha_d \pm \sqrt{9\sin^2\alpha_d - 8} \right) \tag{66}$$

$$\sin\alpha_d \leq -\frac{2\sqrt{2}}{3} \tag{67}$$

During nominal flight, the drone will experience a positive α_d when in cruise. However, in the case of failure, when the induced velocity has to be computed, the drone could pitch or roll excessively causing the air flow to impact the propeller from below. Hence, it is important to consider the presence of the discovered local minima. Even though the function can contain local minima, they could be avoided by a proper selection of hyper-parameters and initialisation of the optimisation; tuning the gradient-descent to the particular (known) $f(v_0)$ function.

Considering extreme values of v_0, it can be observed in Eq. (64) that the second term of $f(v_0)$ is dominant. As a result, $f(-\infty)$ leads to a positive value and $f(+\infty)$ to a negative one, meaning that the function is decreasing in value independently of the value of α_d. In the case that there is a local minimum (Eq. (67) is fulfilled) and it takes place at a lower uniform induced velocity than when $f(v_0) = 0$, as illustrated in Fig. 57, the gradient-descent could be initialised with a high value of v_0 to guarantee that the optimisation will encounter the global optimum before the local minimum. Since the function is decreasing, this approach would not work if $f(v_0) < 0$ at $df(v_0)/dv_0 = 0$, as can be seen in Fig. 58. In order to check whether this latter scenario exists, Eq. (66) is inserted in $f(v_0)$, leading to Eq. (68).

$$T - \frac{\rho\pi R^2 V^2}{2}\left(-3\sin\alpha_d \pm \sqrt{9\sin^2\alpha_d - 8} \right) \sqrt{\cos^2\alpha_d + \frac{1}{16}\left(\sin\alpha_d \pm \sqrt{9\sin^2\alpha_d - 8} \right)^2} \leq 0 \tag{68}$$

Since the local minimum can only be found when $\sin\alpha_d \in [-1, -2\sqrt{2}/3]$ (Eq. (67) is met), the two limits of this range are inserted in Eq. (68), resulting in the two conditions presented in Eq. (69) and Eq. (70). Observing both conditions, the second one is automatically met when the first one is fulfilled. Hence only when Eq. (67) and Eq. (69) are met, there is a local minimum which takes place with a higher uniform induced velocity than when $f(v_0) = 0$. In that case, initialising the optimisation with a high value of v_0 would most likely not converge to the undesired local minimum.

$$\sin\alpha_d = -\frac{2\sqrt{2}}{3}, \quad T < \frac{\sqrt{3}}{3}\rho\pi R^2 V^2 \tag{69}$$

$$\sin\alpha_d = -1, \quad T < \frac{1}{2}\rho\pi R^2 V^2 \tag{70}$$

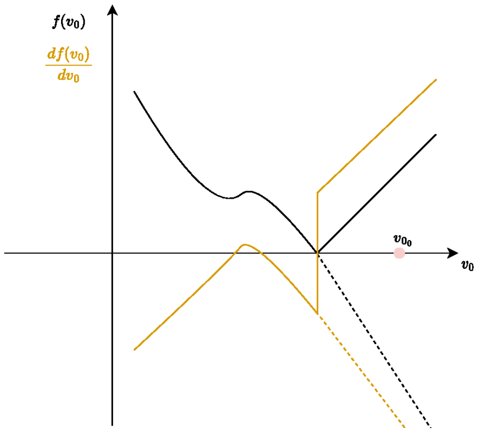


Fig. 57 Desired scenario: local minima at lower induced velocity than global minima, so gradient descend will reach global minima first. The dotted line corresponds to $f(v_0)$ and the bold line to $|f(v_0)|$.

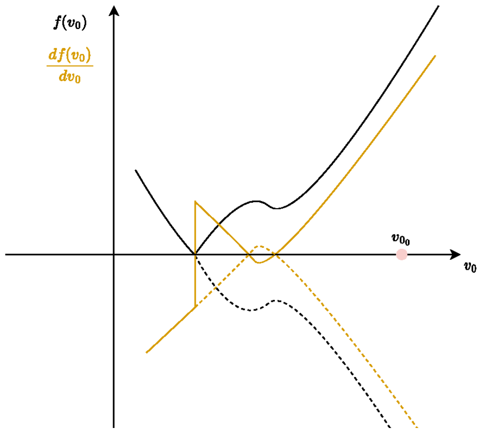


Fig. 58 Undesired scenario: local minima at higher induced velocity than global minima, so gradient descend will reach local minima first. The dotted line corresponds to $f(v_0)$ and the bold line to $|f(v_0)|$.

To assess whether the blade damage simulation with the Bebop 2 model would encounter scenarios in which both conditions (Eq. (67) and Eq. (69)) are met, 100,000 scenarios were run under the specified conditions: $\overrightarrow{V}_x^B \in [-3, 3]$, $\overrightarrow{V}_y^B = 0$, $\overrightarrow{V}_z^B \in [-3, 3]$, $\overrightarrow{\Omega} = \overrightarrow{0}$ and $\omega \in [300, 1256]$. The body velocity in the z-direction can now be positive with an increased absolute value to consider failure cases. Figure 59 displays the V-T plot for all scenarios. Notably, there is no overlap between points satisfying the first condition in Eq. (67) (pink points within the convex hull) and those meeting the second condition in Eq. (69) (green points under the dashed line). Consequently, simulations of the present research will only observe the desired scenario illustrated in Fig. 57. Thus, initiating gradient descent with a high positive v_0 guarantees encountering the global minimum initially.

Now that it has been proven to be beneficial to initialise the gradient-descent with a high positive value of uniform induced velocity, the question is what the exact initialisation value should be. Observing Fig. 57, it is enough to initialise the gradient-descent with a uniform induced velocity value higher than the maximum v_0 that the local minimum could have. If initialised between the local and global minimum, the gradient-descent will move the solution towards the global optimum, to the right along the v_0 axis. If initialised to the right of the global minimum, the gradient-descent will move the solution towards the global optimum, to the left along the v_0 axis.

Given Eq. (66), the maximum v_0 at the local minimum occurs when the square root output is positive and α_d has a value of -90°; resulting in uniform induced flow equalling the incoming velocity ($v_0 = V$). From the 100,000 scenarios presented in Fig. 59, the maximum velocity observed is 4.24 m/s. Therefore, initialising gradient-descent with $v_{0_0} = 4.5$ m/s guarantees the initial function evaluation to the right of the local optima.

Furthermore, the gradient-descent optimisation requires the selection of the learning rate (γ). This hyper-parameter needs to be carefully chosen in order to avoid overshooting the global optimum and landing in the local minimum. Given the update law of the gradient-descent provided by Eq. (71), the algorithm can overshoot the global optimum by a value equal to $\gamma \frac{d|f(v_0)|}{dv_0}$. Hence, the smaller this update step, the lower the probability that the optimisation overshoots the global optimum and lands to the left of the local minimum.

$$v_{0_{i+1}} = v_{0_i} - \gamma \frac{d|f(v_{0_i})|}{dv_0} = v_{0_i} - \gamma \frac{df(v_{0_i})}{dv_0} \frac{f(v_{0_i})}{|f(v_{0_i})|} \tag{71}$$

In this study, two learning rates, 0.5 and 0.1, are assessed and the value of the learning rate is decreased by a factor of 0.5 every time $\frac{d|f(v_0)|}{dv_0}$ changes sign. The performance results using the same 100,000 simulation scenarios of the Bebop 2 drone in Fig. 59 are shown in Table 4. An optimisation is defined as successful when it yields $f(v_0) < 10^{-5}$. As can be seen, the gradient-descent approach has a 100% success rate for both learning rates, in contrast with the slight worse performance of 98.56% for Nelder-Mead. Additionally, the gradient-descent optimisation shows an approximately 20% and 55% computational time reduction with respect to the Nelder-Mead alternative for the 0.1 and 0.5 learning rates, respectively. Hence, the chosen learning rate for the present research is 0.5.

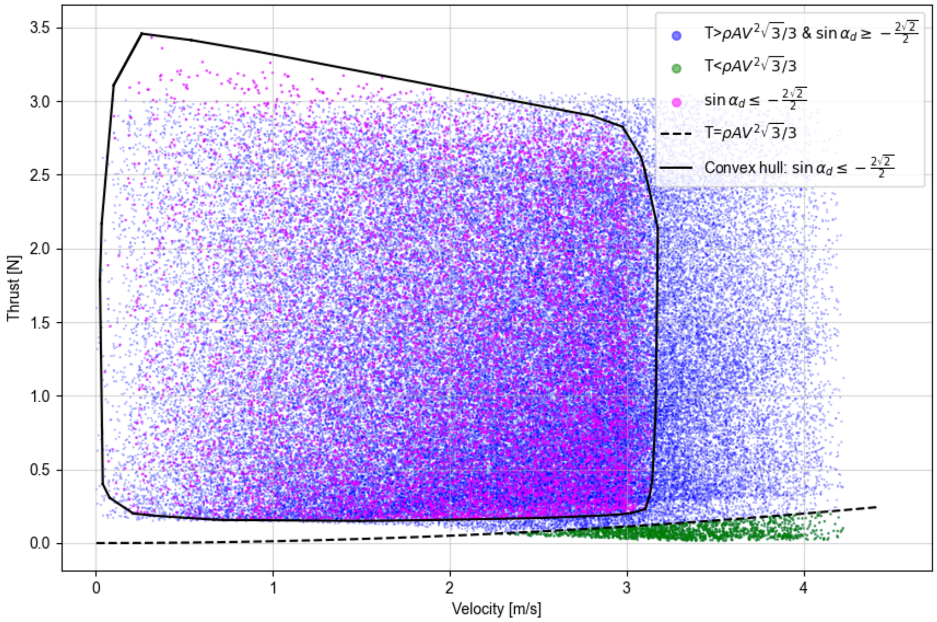


Fig. 59 V-T graph for 100,000 simulation scenarios of the Bebop 2 drone. The convex hull encapsulates all the pink points that meet the (first) condition in Eq. (67)**. The green points under the dashed line meet the (second) condition outlined in Eq.** (69)**. The magenta points are those scenarios in which neither of the conditions are met. From the figure, there is no overlap between both conditions sets.**

Table 4 Performance results of Nelder-Mead and Gradient-Descent with learning rate values of 0.1 and 0.5.

	Success rate [%]	Average time per scenario [ms]
Nelder-Mead	98.56	5.51
Gradient-descent (γ=0.1)	100	4.39
Gradient-descent (γ=0.5)	**100**	**2.45**

Finally, the optimisation concludes when one of the following 3 conditions is met:

1) The maximum number of iterations is reached (i_{max}=10,000).
2) The change in the solution is lower than a threshold ($m < 0.01$) for a certain number of iterations ($c_{max} = 20$).
3) The denominator of the gradient update ($|f(v_0)|$) is less than a very small value ($\epsilon = 10^{-10}$) because then the solution has been found.

Acknowledgments

The work presented in this paper was conducted when José Ignacio de Alvear Cárdenas was a graduate student at the Faculty of Aerospace Engineering at Delft University of Technology within the Control and Simulation Division. Hence, the authors would like to thank this academic institution for providing the funding and the resources that enabled the completion of this research. The authors also thank Charlie Bogaerts and Burhanuddin Saify for their insights and help in the execution of the experimental wind tunnel campaign.

References

[1] Isermann, R., “Fault Diagnosis Systems An Introduction from Fault Detection to Fault Tolerance,” *Fault-Diagnosis Systems*, 2006. https://doi.org/10.1007/3-540-30368-5.

[2] Tidriri, K., Chatti, N., Verron, S., and Tiplica, T., “Bridging Data-Driven and Model-Based Approaches for Process Fault Diagnosis and Health Monitoring: A Review of Researches and Future Challenges,” *Annual Reviews in Control*, Vol. 42, No. C, 2016, pp. 63–81. https://doi.org/10.1016/j.arcontrol.2016.09.008.

[3] Heredia, G., and Ollero, A., “Detection of Sensor Faults in Small Helicopter UAVs Using Observer/Kalman Filter Identification,” *Mathematical Problems in Engineering*, Vol. 2011, 2011. https://doi.org/10.1155/2011/174618.

[4] Aboutalebi, P., Abbaspour, A., Forouzannezhad, P., and Sargolzaei, A., “A Novel Sensor Fault Detection in an Unmanned Quadrotor Based on Adaptive Neural Observer,” *Journal of Intelligent & Robotic Systems*, 2017. https://doi.org/10.1007/s10846-017-0690-7.

[5] Ducard, G., *Actuator Fault Detection in UAVs*, Springer Netherlands, Dordrecht, 2015, pp. 1071–1122. https://doi.org/10.1007/978-90-481-9707-1_43.

[6] Heng, L., Meier, L., Tanskanen, P., Fraundorfer, F., and Pollefeys, M., “Autonomous Obstacle Avoidance and Manoeuvring on a Vision-Guided MAV using On-Board Processing,” *2011 IEEE International Conference on Robotics and Automation*, Shanghai, 2011, pp. 2472–2477. https://doi.org/10.1109/icra.2011.5980095.

[7] Sun, S., Baert, M., Schijndel, B., and De Visser, C., “Upset Recovery Control for Quadrotors Subjected to a Complete Rotor Failure from Large Initial Disturbances,” *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, pp. 4273–4279. https://doi.org/10.1109/icra40945.2020.9197239.

[8] Nguyen, D.-T., Saussié, D., and Saydy, L., “Fault-Tolerant Control of a Hexacopter UAV based on Self-Scheduled Control Allocation,” *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 385–393. https://doi.org/10.1109/icuas.2018.8453440.

[9] Xue, Y., Zhen, Z., Yang, L., and Wen, L., “Adaptive Fault-Tolerant Control for Carrier-Based UAV with Actuator Failures,” *Aerospace Science and Technology*, Vol. 107, 2020, p. 106227. https://doi.org/10.1016/j.ast.2020.106227.

[10] Wang, B., Shen, Y., and Zhang, Y., “Active Fault-Tolerant Control for a Quadrotor Helicopter Against Actuator Faults and Model Uncertainties,” *Aerospace Science and Technology*, Vol. 99, 2020, p. 105745. https://doi.org/10.1016/j.ast.2020.105745.

[11] Jiang, Y., Zhiyao, Z., Haoxiang, L., and Quan, Q., “Fault Detection and Identification for Quadrotor Based on Airframe Vibration Signals: A Data-Driven Method,” *2015 34th Chinese Control Conference (CCC)*, 2015, pp. 6356–6361. https://doi.org/10.1109/chicc.2015.7260639.

[12] Sun, S., and de Visser, C., “Aerodynamic Model Identification of a Quadrotor Subjected to Rotor Failures in the High-Speed Flight Regime,” *IEEE Robotics and Automation Letters*, Vol. 4, No. 4, 2019, pp. 3868–3875. https://doi.org/10.1109/lra.2019.2928758.

[13] Avram, R. C., Zhang, X., and Khalili, M., “Quadrotor Actuator Fault Diagnosis with Real-Time Experimental Results,” *Annual Conference of the PHM Society*, Vol. 8, 2016. https://doi.org/10.36001/phmconf.2016.v8i1.2504.

[14] Ghalamchi, B., Jia, Z., and Mueller, M. W., “Real-Time Vibration-Based Propeller Fault Diagnosis for Multicopters,” *IEEE/ASME Transactions on Mechatronics*, Vol. 25, No. 1, 2020, pp. 395–405. https://doi.org/10.1109/tmech.2019.2947250.

[15] Newman, S., “Principles of Helicopter Aerodynamics,” *The Aeronautical Journal*, Vol. 111, 2007, pp. 825–826. https://doi.org/10.1017/S0001924000087352.

[16] Selig, M., *Modeling Full-Envelope Aerodynamics of Small UAVs in Realtime*, Toronto, Ontario, Canada, 2010. https://doi.org/10.2514/6.2010-7635.

[17] Orsag, M., and Bogdan, S., *Influence of Forward and Descent Flight on Quadrotor Dynamics*, IntechOpen, London, United Kingdom, 2012, Chap. 7, pp. 141–156. https://doi.org/10.5772/37438.

[18] Khan, W., and Nahon, M., “Toward an Accurate Physics-Based UAV Thruster Model,” *IEEE/ASME Transactions on Mechatronics*, Vol. 18, No. 4, 2013, pp. 1269–1279. https://doi.org/10.1109/tmech.2013.2264105.

[19] Burton, T., Sharpe, D., Henkins, N., and Bossanyi, E., *Wind Energy Handbook*, 2$^{\text{nd}}$ ed., Wiley, Chichester, West Sussex, 2011.

[20] Gill, R., and D’Andrea, R., “Propeller Thrust and Drag in Forward Flight,” *2017 IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 73–79. https://doi.org/10.1109/ccta.2017.8062443.

[21] Gill, R., and D'Andrea, R., "Computationally Efficient Force and Moment Models for Propellers in UAV Forward Flight Applications," *Drones*, Vol. 3, No. 4, 2019. https://doi.org/10.3390/drones3040077.

[22] Niemiec, R., and Gandhi, F., "Effects of Inflow Model on Simulated Aeromechanics of a Quadrotor Helicopter," *Proceedings of the 2016 72nd American Helicopter Society (AHS) International Annual Forum*, 2016.

[23] Leishman, J. G., *Principles of Helicopter Aerodynamics*, Cambridge University Press, Cambridge, 2006.

[24] Howlett, J. J., *UH-60A Black Hawk Engineering Simulation Program*, Vol. 1, NTIS, Springfield, Va., 1981.

[25] Pitt, D., and Peters, D., "Theoretical Prediction of Dynamic Inflow Derivatives," *Vertica*, Vol. 5, 1981.

[26] White, F., and Blake, B. B., "Improved Method Of Predicting Helicopter Control Response And Gust Sensitivity," *Proceedings of the 1979 35th Annual Forum of American Helicopter Society*, 1979.

[27] Drees, J., "A Theory of Airflow Through Rotors and its Application to some Helicopter Problems," *Journal of the Helicopter Association of Great Britain, 3*, Vol. 2, 1949, pp. 79–104.

[28] Sun, S., de Visser, C. C., and Chu, Q., "Quadrotor Gray-Box Model Identification from High-Speed Flight Data," *Journal of Aircraft*, Vol. 56, No. 2, 2019, pp. 645–661. https://doi.org/10.2514/1.c035135.

[29] Byrd, R. H., Hribar, M. E., and Nocedal, J., "An Interior Point Algorithm for Large-Scale Nonlinear Programming," *SIAM Journal on Optimization*, Vol. 9, No. 4, 1999, pp. 877–900. https://doi.org/10.1137/s1052623497325107.

[30] Mostafa, A.-B., and Ebrahim, M., "3D Laser Scanners' Techniques Overview," *International Journal of Science and Research (IJSR)*, Vol. 4, 2015, pp. 5–611. https://doi.org/10.1007/s41062-021-00550-9.

[31] Perez-Cortes, J.-C., Perez, A., Saez-Barona, S., Guardiola, J.-L., Salvador Igual, I., and Sáez-Barona, S., "A System for In-Line 3D Inspection without Hidden Surfaces," *Sensors*, Vol. 18, 2018, p. 2993. https://doi.org/10.3390/s18092993.

[32] Bhatti, A. Q., Wahab, A., and Sindi, W., "An Overview of 3D Laser Scanning Techniques and Application on Digitization of Historical Structures," *Innovative Infrastructure Solutions*, Vol. 6, No. 4, 2021, p. 186. https://doi.org/10.1007/s41062-021-00550-9.

[33] Leza, D. V., "Development of a Blade Element Method for CFD Simulations of Helicopter Rotors using the Actuator Disk Approach," Master's thesis, Delft University of Technology, 2018.

[34] Omari, S., Hua, M.-D., Ducard, G., and Hamel, T., "Nonlinear Control of VTOL UAVs Incorporating Flapping Dynamics," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 2419–2425. https://doi.org/10.1109/iros.2013.6696696.

[35] Zhang, P., and Huang, S., "Review of Aeroelasticity for Wind Turbine: Current Status, Research Focus and Future Perspectives," *Frontiers in Energy*, Vol. 5, No. 4, 2011, p. 419–434. https://doi.org/10.1007/s11708-011-0166-6.

[36] Gessow, A., and Myers, G. C., *Aerodynamics of the Helicopter*, F. Ungar Pub. Co., New York, 1967.

[37] Mangler, K. W., and Squire, H. B., *The Induced Velocity Field of a Rotor*, Ministry of Supply, Aeronautical Research Council, London, 1950.

[38] Stepniewski, W. Z., *Rotary-Wing Aerodynamics*, Dover Publications, New York, 1984.

[39] Ormiston, R. A., *An Actuator Disk Theory for Rotor Wake Induced Velocities*, Advisory Group for Aerospace Research and Development (AGARD-CP-111), Aerodynamics of Rotary Wings, 1972. https://doi.org/10.1007/978-3-030-05455-7_2-2.

[40] Ormiston, R. A., "Induced Power of the Helicopter Rotor," *60th Annual Forum of the American Helicopter Society International*, 2004, pp. 33–53.

[41] Payne, P. R., "Helicopter Dynamics and Aerodynamics," Vol. 63, No. 585, 1959. https://doi.org/10.1017/S0368393100071728.

[42] Peters, D. A., and Haquang, N., "Dynamic Inflow for Practical Applications," *Journal of the American Helicopter Society*, Vol. 33, No. 4, 1988, pp. 64–68. https://doi.org/10.4050/jahs.33.64.

[43] Peters, D. A., Boyd, D. D., and He, C., "Finite-State Induced-Flow Model for Rotors in Hover and Forward Flight," *Journal of The American Helicopter Society*, Vol. 34, 1989, pp. 5–17. https://doi.org/10.4050/jahn.34.5.

[44] Peters, D. A., and He, C., “Correlation of Measured Induced Velocities with a Finite-State Wake Model,” *Journal of The American Helicopter Society*, Vol. 36, 1991, pp. 59–70. https://doi.org/10.4050/jahs.36.59.

[45] Zhao, J., Prasad, J., and Peters, D., “Rotor Dynamic Wake Distortion Model for Helicopter Maneuvering Flight,” *Journal of The American Helicopter Society*, Vol. 49, 2004, pp. 414–424. https://doi.org/10.4050/jahs.49.414.

[46] Leishman, J. G., Bhagwat, M. J., and Bagai, A., “Free-Vortex Filament Methods for the Analysis of Helicopter Rotor Wakes,” *Journal of Aircraft*, Vol. 39, No. 5, 2002, pp. 759–775. https://doi.org/10.2514/2.3022.

[47] Singh, A. P., Medida, S., and Duraisamy, K., “Machine-Learning-Augmented Predictive Modeling of Turbulent Separated Flows over Airfoils,” *AIAA Journal*, Vol. 55, No. 7, 2017, pp. 2215–2227. https://doi.org/10.2514/1.j055595.

[48] Volpiani, P. S., Meyer, M., Franceschini, L., Dandois, J., Renac, F., Martin, E., Marquet, O., and Sipp, D., “Machine Learning-Augmented Turbulence Modeling for RANS Simulations of Massively Separated Flows,” *Physical Review Fluids*, Vol. 6, 2021, p. 064607. https://doi.org/10.1103/physrevfluids.6.064607.

[49] Lomb, N. R., “Least-Squares Frequency Analysis of Unequally Spaced Data,” *Astrophysics and Space Science*, Vol. 39, No. 2, 1976, pp. 447–462. https://doi.org/10.1007/bf00648343.

[50] Scargle, J. D., “Studies in Astronomical Time Series Analysis. II. Statistical Aspects of Spectral Analysis of Unevenly Spaced Data,” *Astrophysical Journal*, Vol. 263, 1982, pp. 835–853. https://doi.org/10.1086/160554.

[51] Scargle, J. D., “Studies in Astronomical Time Series Analysis. III. Fourier Transforms, Autocorrelation Functions, and Cross-Correlation Functions of Unevenly Spaced Data,” *Astrophysical Journal*, Vol. 343, 1989, p. 874. https://doi.org/10.1086/167757.