

# Low-fidelity design optimization and development of a VTOL swarm UAV with an open-source framework

Nikhil Sethi <sup>a,\*</sup>, Saurav Ahlawat <sup>b</sup>

<sup>a</sup> Department of Mathematics and Computer Science, South Asian University, New Delhi, India

<sup>b</sup> Department of Aerospace Engineering, University of Illinois, Urbana-Champaign, IL, USA

## ARTICLE INFO

### Keywords:

Drones  
CFD  
MDO  
Aircraft design  
Open source  
PSO

## ABSTRACT

Hybrid configurations in aircraft design are highly favorable as they can achieve the appropriate trade-offs required to develop a generalized unmanned aerial system (UAS). Rapid prototyping of such systems at the student level is challenging because commercial software is expensive and difficult to interlink with other tools for creating a multi-disciplinary design. We address this challenge by conceptualizing an aircraft design framework made entirely of open-source software, libraries, and in-house code. We then use this framework to design an all-electric unmanned aerial system with transitioning Vertical Take-off and Landing (VTOL) and Fixed-Wing (FW) modes. The UAV is capable of long-range surveillance up to 100 Kilometers and carrying a maximum relief payload of 1 kg while operating in an ad-hoc wi-fi network with a swarm of similar UAVs. A low fidelity particle swarm optimization algorithm (PSO) and a comprehensive propulsion architecture is also incorporated and validated against commercial software. To validate the design, a prototype is fabricated from glass-fiber and XPS foam, integrated with appropriate sensors and tuned using ArduPilot software. The results show that low-fidelity design is a safe starting point for prototyping under constrained timelines. The study is concluded by discussing the technical challenges of using free software, and some practical considerations while flight testing a UAV with a hybrid configuration.

## 1. Introduction

The development of Unmanned Aerial Vehicles (UAV) is gaining popularity with the advent of more research and robust tools for the same. An even more recent development is the VTOL hybrid concept which combines the advantages of both fixed-wing and vertical thrust configurations [1,2]. They can adapt to the mission scenario while keeping good performance characteristics throughout. The first challenge in developing such vehicles is the added weight and parasite drag of the vertical thrust components as compared to conventional designs [3]. The low operating Reynolds number of Micro Aerial Vehicles (MAVs) makes this an even more difficult problem to tackle.

Secondly, undergraduate students and inexperienced designers face the unavailability of consolidated resources to learn and develop good designs. This is even more pronounced in the case of aircraft design, where the intertwined relationships of multiple disciplines are profound. Commercial tools act as a proponent in such a case as they do not provide the user with the freedom to learn and customize accordingly. Open-source tools and the development of custom libraries/components help in learning from scratch, and the leverage to create and use such tools has never been greater than before.

Finally, a complete end-to-end solution, from conceptual design to autopilot testing, is rarely completed in academic environments due to a lack of funding or constrained timelines for publishing research. A complete document addressing the various challenges while designing, developing, and testing a swarm-capable UAV can benefit the emerging field of unmanned aerial systems and swarm robotics. This paper recognizes the above challenges and attempts to alleviate them by developing a sound methodology spread across various entirely free tools and in-house code. Our research is aimed primarily to help students and upcoming researchers develop feasible designs for student competitions or learning aircraft design. Formally, the rationale of our research is centered around the following outcomes:

(a) A low-fidelity design framework using Particle Swarm Optimization for a VTOL hybrid UAV within the defined aerodynamic and stability constraints.

(b) Inculcation of completely open-source tools and libraries for the entire design and testing routine to move towards “free design” methodologies.

(c) End-to-end fabrication and testing of the UAV equipped with appropriate communication sensors for operating in a homogeneous swarm.

\* Corresponding author.

E-mail address: [nikhilsethi\\_bt2k16@dtu.ac.in](mailto:nikhilsethi_bt2k16@dtu.ac.in) (N. Sethi).

To the best of our knowledge, this is the first work that develops a blended wing tri-copter UAV configuration by incorporating a set of completely open-source tools into a reusable design pipeline. The paper is organized as follows: A background and literature review of the current state of research is carried out in Section 2. This is followed by establishing a custom open-source design pipeline in Section 3. Sections 4 to 8 go through the entire design process from concept to rendering. We also fabricate a prototype of the UAV in Section 9 and explain the challenges faced during the testing of the same in Section 10.1. To conclude, some final comments and discussions on the design, methods, and platforms used are also carried out. Note: This research extensively uses the term “hybrid” which is not to be confused with hybrid-electric propulsion, which relies on gasoline-powered generators to produce electric power. Instead, “hybrid” here would only refer to the aircraft’s operational capabilities as both a multi-rotor and a fixed-wing craft unless stated otherwise.

## 2. Background

A VTOL fixed-wing UAV can take off and land vertically along with sustained cruised flight. These hybrid UAVs combine VTOL capability with the standard forward propulsion of a fixed-wing UAV. Rotary lift propellers are typically incorporated into the aircraft’s wings in many hybrid VTOL UAVs, enabling a transition to forward flight. VTOL fixed-wings have several advantages over typical fixed-wing unmanned aircraft. They require much less space to launch and recover, as they do not need to use a specified area for Take-Off [4]. They are suited to applications where aerial inspection and monitoring are needed, allowing the aircraft to hover over a position for some time.

A study of [5] covers the use of SUAVE, a software used to analyze, optimize, and design a range of small UAVs, which is a flexible aircraft design tool with the unique ability to handle alternative energy systems and unconventional designs. In [6], a configuration similar to our work is explored using standard low fidelity analyses. The nature of the software and tools used is not stated, however. In [7], potential design strategies are presented and compared, identifying various design variables and constraints that take precedence during a particular optimization while taking a novel weight build-up approach. This work simplifies the more extensive study of optimizing the sensitivity of constraints while removing implicitly resolved parameters alongside reducing the computational cost. Inculcation of stable open-source tools to address problems in aircraft design is discussed in [8].

Various tools and resources were used while designing and developing the UAV in this paper. Brief descriptions of each of them and their capabilities are highlighted below

- (a) *FreeCAD*<sup>1</sup>: It is a powerful open-source solid modeling software written in C++ with a highly inter-operable Python API. It has multiple workbenches to deal with different kinds of geometry. Owing to the software’s long tenure, the documentation is sufficient and has a well-maintained forum.
- (b) *OpenFoam*<sup>2</sup> + *ParaView*<sup>3</sup>: An automated numerical solver written in C++ for continuum mechanics problems in computational fluid dynamics [9]. It has a powerful file-based API and pre-processing utilities like Snappy-Hex-Mesh. The library is also distributed with ParaView: a versatile post-processing software to visualize the computed fields. OpenFOAM is well-regarded in the research community and has been linked with several applications to further its development [10,11]

- (c) *LuxCoreRender*<sup>4</sup>: This rendering software provides designers with a powerful tool to create realistic and accurate images. It is very flexible and does not compromise on quality and physical correctness by making ‘unbiased’ designs using equations that model light flow. There are many standard features like configuration file systems and acceleration with a CUDA backend.
- (d) *OpenVSP*<sup>5</sup>: It is a parametric aircraft geometry tool where the user is allowed to create a 3D model of an aircraft defined by standard engineering parameters. This model can be processed into formats suitable for engineering analysis. This software was released as an open-source project under NASA’s open-source Agreement (NOSA) version 1.3. giving the advanced tools of aircraft design for use to the general public. OpenVSP has gained recognition through the years and is popular among researchers for extending it to a complete MDO software [12,13].
- (e) *XFLR5*<sup>6</sup>: It is an analysis software for airfoils, wings, and planes operating at low Reynolds Numbers, which includes XFoil’s Direct and Inverse analysis capabilities with wing design and analysis capabilities based on the Lifting Line Theory, on the Vortex Lattice Method, and a 3D Panel Method. It is prevalent among students and novice researchers as it has a good learning curve and an interactive GUI.
- (f) *ArduPilot SITL*<sup>7</sup> + *Mission Planner*<sup>8</sup>: The SITL (software in the loop) simulator allows us to run Plane and Copter simulations. It is a build of the autopilot code using the original C++ compiler, giving us a native executable that allows one to test the code’s behavior without implementation on the hardware itself. The simulator uses the sensor data from various flight dynamics models in a flight simulator. This allows testing of ArduPilot SITL on a number of vehicle types, essential for testing a hybrid system such as ours.

## 3. Design methodology

Incorporating completely open-source tools for the conceptual preliminary and detailed design is a challenge because of three factors:

1. The accuracy of results.
2. Lack of documentation.
3. Lack of experienced users/populated forums.

Considering the above challenges and the fidelity required for the design, the tools and the design process are shown in Fig. 1.

The high degree of interrelation between disciplines in aircraft design makes it difficult to dodge locally optimal designs developed by intuition. Multidisciplinary Design Optimization (MDO) has become a necessary part of the toolbox of aerospace designers. The approach used in the following sections is not a full-scale MDO solution by any means but a novel, low-fidelity attempt at design exploration. A human is kept in the loop at all stages to guide the design and perform validation checks. Further methods pertaining to the same are explained in the following sections.

## 4. Conceptual design

We start by using a systems engineering approach to establish high level design requirements and timelines for the project. This was completed using V-models and system design timelines, which are not covered here for compactness but are included in supplementary material (S1). For the purpose of this research, we model our aircraft

<sup>4</sup> <https://github.com/LuxCoreRender/LuxCore>.

<sup>5</sup> <https://github.com/OpenVSP/OpenVSP>.

<sup>6</sup> <https://sourceforge.net/projects/xflr5/>.

<sup>7</sup> <https://github.com/ArduPilot/ardupilot>.

<sup>8</sup> <https://ardupilot.org/planner/>.

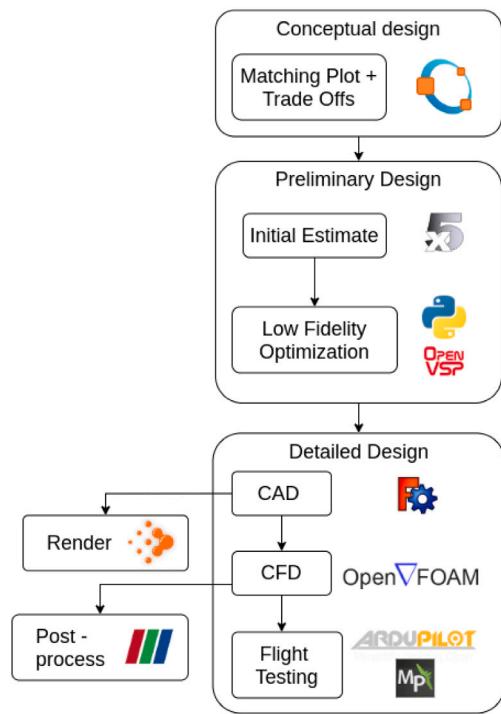


Fig. 1. High level design process.

to maximize its flight range up to 100 Kilometers (Kms), cruising at 25 m/s without sacrificing its VTOL functionality. The aircraft will keep a standard mission profile with vertical take-off, cruised flight, surveillance and target detection, payload delivery, and vertical landing. It is also kept in mind that multiple similar UAVs may be used in a collaborative swarm which is advantageous in many applications. A concept of operations for the entire mission profile was also envisioned and a video of the same has been included in the supplementary material (S2).

#### 4.1. Configuration selection

There are many configuration parameters to consider, and respective trade-offs for each are carried out using a Pugh matrix analysis. These matrices (given in supplementary material (S3)) are filled with weighted scores for each configuration, and the scores are summed up to compare and evaluate. Since the design is conceptual at this stage, inherent safety factors and design experience play a crucial role in selection.

A hybrid of a blended flying wing configuration supplemented with a tri-copter configuration with two forward tilting motors is arrived at. The front motors tilt forward for transition to cruise flight after a vertical take-off and tilt back to multi-rotor while landing. These motors can be vectored along the aircraft's pitch axis to stabilize the yawing moment. The rear motor is operational for the multi-rotor segment of the flight. The following equation gives the total drag component of the UAV and can be used to determine the areas which can be altered to lower the drag.

$$(C_{Do})_{total} = (C_{Do})_w + (C_{Do})_{ht} \frac{S_{ht}}{S} + (C_{Do})_{vt} \frac{S_{vt}}{S} + \dots + (C_{Do})_{vtol} \frac{S_{vtol}}{S}$$

Here  $(C_{Do})_w$  is the total zero-lift drag of the wing,  $(C_{Do})_{ht}$  and  $(C_{Do})_{vt}$  are the horizontal and vertical tail zero-lift drags, and  $(C_{Do})_{vtol}$  accounts for the drag in extra mechanisms (motors and structural components necessary for transition operations).  $S$  represents the wetted areas of

the various components (for eg.  $S_{vtol}$  = wetted area of VTOL motor, tilt-rotor mechanisms, aluminum struts etc.) and their ratios with respect to the wetted wing area. A blended flying wing will prove to be a better candidate for long-endurance missions with significantly reduced  $(C_{Do})_{ht}$  component, better performance of winglets to reduce vortex induced drag, and an overall lower  $(C_{Do})_{vt}$  component (Supplementary material S2b).

Most commercially available products use a quad-plane configuration, which is essentially a quad-copter attached to the frame of an independent fixed-wing aircraft. This solution, save for its simplicity, added greater structural weight and reinforcement, which resulted in more  $(C_{Do})_{vtol}$  components. The extra deadweight while cruising would not be an optimal choice for a high endurance mission.

A tri-plane with two forward tilt motors is a highly favorable choice in terms of weight and lower  $(C_{Do})_{vtol}$ . Even though planes with greater number of rotors provide better stability and airworthiness, their cost and weight diminish the advantages in financially constrained environments. Therefore, a tri-copter configuration was chosen due to its lower current draw, less dead weight, and lower cost (Supplementary material S2a). Note that such trade-offs, while feasible at a conceptual stage, do not provide the entire picture, and actual flight tests were done on a prototype frame to confirm/refute the assumptions. The challenges regarding the same are discussed thoroughly in Section 10.1.

#### 5. Propulsion

The propulsion selection is one of the most important trade-offs to obtain the most out of the aircraft's endurance and reach the desired range. During cruise flight, all the load will be on the forward motors whereas, during hover, all the three motors would share the load. This introduces the issue of transition where the forward motors have to be chosen in such a way to provide sufficient thrust during hover and have optimal RPM and pitch speed for a sustained cruised flight in the mission.

As trends dictates, we must have a minimum thrust to weight ratio (T/W) of 1.5 for a hovering craft for sufficient thrust during maneuvers. Further, existing aircraft design trends show a T/W of 0.75 to be sufficient during cruise. This allows us to arrive at an initial thrust margin for all the motors and a ratio of 1 : 1 : 2 between the left-front, right-front, and rear motors, respectively. This distribution was necessary to ensure that the forward motors operate at their peak optimum during cruise. The rear motor, only operational during hover segments, is optimized for the same, and its thrust is finalized only after calculating the safe thrust for the front motors during cruise. This is primarily done because the cruise segment dominates the mission profile.

#### 6. Weight estimation

The initial estimation is done by analyzing historical data, collecting the necessary features, and arriving at a hypothesis akin to the methods in [14]. Since the payload is defined within the constraints, the payload fraction was used as the most important feature at this stage, and a linear regression curve was fitted to the data (supplementary material S4a). This trend line, and the quadratic equation generated are solved analytically (Supplementary material S4b) to give a suitable prediction. It is interesting to note how the value of  $b$  is larger than typically obtained for fixed-wings ( $\approx 0.7$ ). This is in line with expectations and primarily attributed to the extra weight used to carry VTOL systems at the expense of better aerodynamic efficiency.

$$\frac{W_e}{W_o} = a \cdot W_o + b \quad (1)$$

$$W_o = W_e + W_p \quad (2)$$

where,

$$a = -0.00468; b = 0.887 \quad (3)$$

The above regression analysis yields an empty weight ( $W_e$ ) of 5.84 kg and, therefore, an estimated total mass ( $W_t$ ) of 6.84 kg to carry 1 kg of payload ( $W_p$ ). This estimate is further qualified by including a safety margin of 160 g based on prior experience, possible dead-weights, and accumulation of weight after repairs because of potential crashes.

$$W_p = 1 \text{ kg} \quad (4)$$

$$W_o = 6.84 + 0.16 \approx 7 \text{ kg} \quad (5)$$

## 7. Preliminary design

The preliminary design stage makes use of extensive mathematical optimization. The objective of the optimization process is to design a geometry and propulsion system to maximize the range — which implies minimizing power and maximizing cruise efficiency. In our case, we have divided the optimization and design process into three stages.

### 7.1. Initial sizing

The first stage gives a good estimate of the two driving factors for any design: wing loading and power loading. Appropriate relations for the transition, hover, and climb phases for the tri-copter sizing [15] are also included to account for disk loading using an approach similar to the one in [14]. Since a rough but mathematically sound estimate is desired at this stage, a design space is created using Linear programming based on the formulae in [14,16][17]. As a result, we only include brief explanations of the equations here.

Eq. (6) gives the wing loading at which the plane stalls. This is usually the constraint that sets the lower bound on the wing area. In cases where the stall speed is high and the cruise ceiling is also high enough, Eqs. (7) might set this lower bound. Eqs. (8), (9) and (10) give the optimal wing loadings for maximizing endurance, climbing at a certain rate ( $ROC$ ), and moving with  $V_{max}$  velocity at sea level ( $SL$ ) altitude respectively. They are derived in [14,17]. Here  $\eta_p = 0.8$  is the propeller efficiency,  $AR = 8$  the aspect ratio,  $(L/D)_{max} = 11$ , and  $e = 0.8$  is the Ostwald's efficiency factor. Note that these values, especially for  $AR$  and  $(L/D)_{max}$  are just educated guesses and must be iterated and changed depending on the final results.

$$\left(\frac{W}{S}\right)_{stall} = 0.5\rho(V_{stall})^2 C_{L_{max}} \quad (6)$$

$$\left(\frac{W}{S}\right)_{maxrange} = q\sqrt{\pi(AR)eC_{D_0}} \quad (7)$$

$$\left(\frac{W}{S}\right)_{maxloiter} = q\sqrt{3\pi(AR)eC_{D_0}} \quad (8)$$

$$\left(\frac{W}{P}\right)_{ROC} = \frac{1}{\frac{ROC}{\eta_p} + \sqrt{\frac{2}{\rho\sqrt{\frac{3C_{D_0}}{K}}} \left(\frac{W}{S}\right) \left(\frac{1.155}{\left(\frac{L}{D}\right)_{max} \eta_p}\right)}} \quad (9)$$

$$\left(\frac{W}{P_{SL}}\right)_{V_{max}} = \frac{\eta_p}{\frac{1}{2}\rho_o V_{max}^3 C_{D_0} \left(\frac{W}{S}\right) + \frac{2K}{\rho \sigma_{air} V_{max}} \left(\frac{W}{S}\right)} \quad (10)$$

The following equations are used to size the propellers' total area that provides vertical thrust. Similar to the wing loading, a disk loading ( $DL$ ) characterizes this area. Eq. (13) is used to size the propeller area during transition. This relation aims to keep a constant velocity during the shift from hover to cruise mode and is therefore dependent on the tilting angle ( $\theta_{tilt}$ ). The angle here is taken as constant. However, one might also explore a varying angle with a constant rate which closely matches with how real systems transition. These equations are derived in [16]

$$\left(\frac{W}{P}\right)_{hover} = FOM \left( \frac{\sqrt{2\rho}}{DL} \right) \quad (11)$$

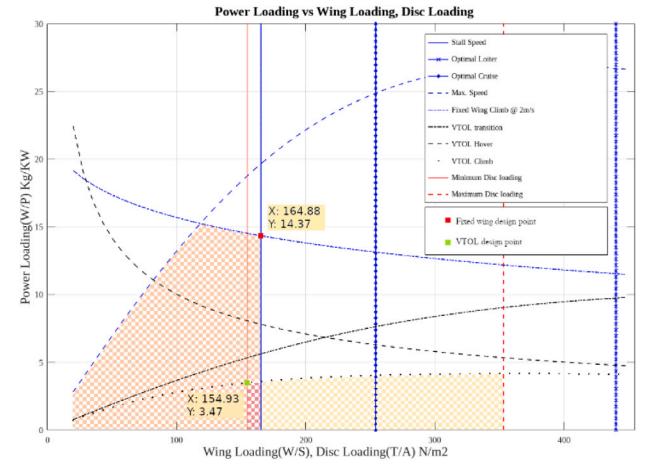


Fig. 2. Linear programming model.

$$\left(\frac{W}{P}\right)_{climb} = \frac{1}{V_y - \frac{k_1 V_y}{2} + \frac{k_1}{2} \sqrt{V_y^2 + \frac{2(DL)}{\rho_o} + \dots} + \frac{\rho_o V_{tip}^3}{(DL)} \left(\frac{\sigma C_d}{8}\right)} \quad (12)$$

$$\left(\frac{W}{P}\right)_{trans} = \frac{1}{d_1 + d_2 + d_3} \quad (13)$$

where,

$$d_1 = \frac{k_1}{\sin(\theta_{tilt})} \sqrt{\frac{-V_\infty^2}{2} + \sqrt{\left(\frac{-V_\infty^2}{2}\right)^2 + \dots} + \left(\frac{DL}{2\rho \sin(\theta_{tilt})}\right)^2} \quad (14)$$

$$d_2 = \frac{\rho V_{tip}^3}{DL} \left(\frac{\sigma C_d}{8} (1 + 4.6\mu^2)\right) \quad (15)$$

$$d_3 = \left( \frac{1}{2} \rho_o V_\infty^3 C_{D_0} \frac{1}{\left(\frac{W}{S}\right)} + \frac{2K}{\rho_o V_\infty} \left(\frac{W}{S}\right) \right) \quad (16)$$

The LPP outputs a comfortable design (Fig. 2) space to choose points from depending on the desired performance. To make the analysis more constrained, minimum and maximum values of disk loading are also used depending on the motors and their rated propeller sizes. The VTOL and fixed-wing design points are far apart, both constrained by their climb limits. The maximum power points are chosen for each configuration to incorporate a safety factor.

$$S_{wing} = \frac{W_o}{(W/S)_{opt}} \quad (17)$$

$$P_{cruise} = \frac{W_o}{(W/P)_{FW}} \quad (18)$$

$$P_{VTOL} = \frac{W_o}{(W/P)_{VTOL}} \quad (19)$$

### 7.2. Geometry optimization

#### 7.2.1. Initial estimate

The aircraft performance is first estimated on XFLR5 to obtain initial but educated estimates of static stability parameters. The analysis starts with the selection of a reflexed airfoil and rudimentary optimization at multiple angles of attacks by observing peak  $C_l/C_d$  values (Fig. 4(a)) and stall qualities (Fig. 4(b)) at cruising altitude and Reynold's number.

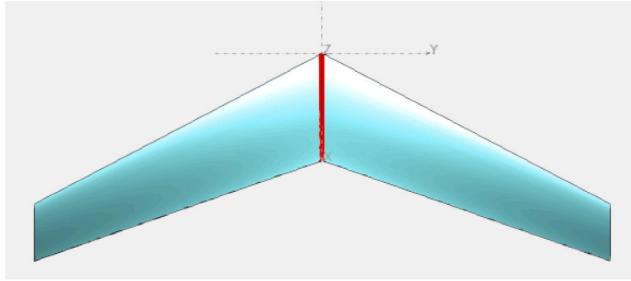


Fig. 3. Wing top view.

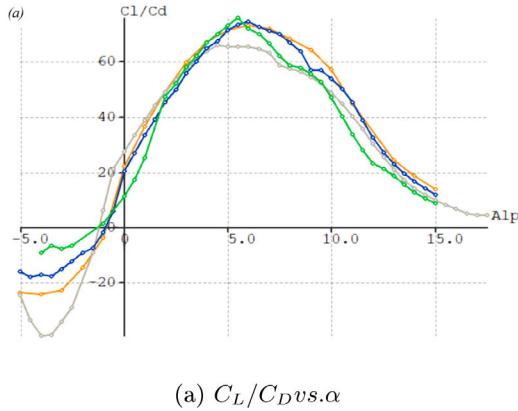


Fig. 4. Xfoil curves.

The methods for the same are taken from [18] and a trade-off matrix is given in supplementary material (S2d).

Fixed lift analysis is then conducted on the operating GTOW of the flying wing (Fig. 3) to introduce static stability into the aircraft. Iterations were performed on the design variables to optimize the cruise  $C_L/C_D$  of the whole wing while balancing out the Center of Gravity (CoG) placement to ensure close to null  $C_m$  (pitching moment). The resultant twist in the wing subsequently reduced the  $C_L$  of the whole wing, making the wing area more than initial calculations.

A rough and conceptual estimate of the intended design (Fig. 5) is also analyzed to ensure consistency in performance estimates.

#### 7.2.2. Problem setup

The second stage is the optimization of geometry. While being decoupled from propulsion selection is not optimal, it makes the development of the interface and debugging easier. Owing to the small number of variables, ease of writing code, and pre-existing libraries [19], a particle swarm optimizer is used. The parameter settings for the same are given in Table 1. The inertial and cognitive weights were

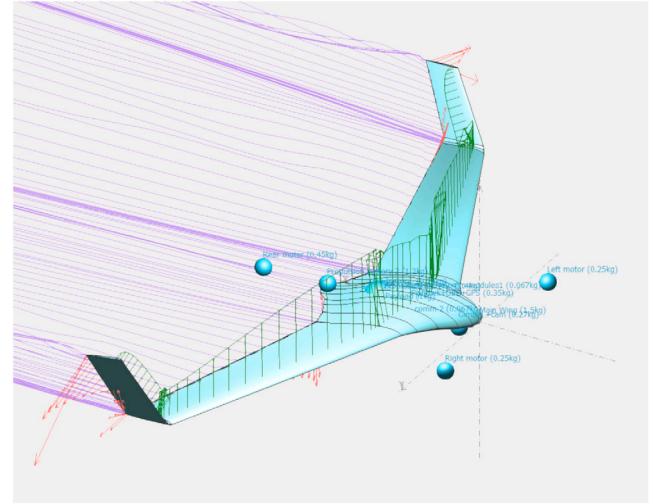


Fig. 5. Preliminary design.

**Table 1**  
PSO parameter settings.

Parameter	Initial	Final	Method
$c_1$ (Cognitive weight)	2.5	0.5	Linear decay
$c_2$ (Social weight)	2.5	2.5	Constant
$w$ (Inertia weight)	0.9	0.4	Linear decay
$N$ (generations)	0	100	Linear

decreased over generations to exploit high fitness solutions [20]. The objective function used for the problem is just the weighted sum of each individual objectives (Eq. (20)). For simplicity all the weights ( $w$ ) are taken as equal and only have opposite signs based on the minimization/maximization of each of the objectives. Four variables namely the aspect ratio ( $AR$ ), root-tip twist ( $\Theta_{rel}$ ), taper-ratio ( $\lambda$ ) and leading edge sweep ( $\Lambda_{LE}$ ) are used to parameterize the problem. The combined objective function accounts for the following three factors:

1. Drag: the analysis is done at constant lift coefficient (Eq. (23)) which is equal to the airfoil lift coefficient at cruise condition taken with a margin to account for a 2D cross-section vs 3D wing [14].
2. Lift distribution: The actual lift distribution ( $C_L^a$ ) is a function of twist, taper ratio and sweep. A least squares cost function is used to match the target curve( $C_L^t$ ) of the load distribution. A separate python script calculates this.
3. Stability: Since optimizing for stability often requires compromise on performance, the damping ratios of the short period ( $\tau_{SP}$ ), Dutch roll ( $\tau_{DR}$ ) and Phugoid ( $\tau_{PH}$ ) mode are incorporated within the objective. For static stability however, two constraints to the allowable static margin (S.M. Eq. (24)) at zero moment at the design lift coefficient (Eq. (25)) are enough. The stability coefficients and “BaseAero” values are sought from the .stab file output by OpenVSP and converted to a data frame for further processing.

The values in constraint Eqs. (28) to (31) are sensible values which closely follow trends in supplementary material (S4(a)).

$$\text{Min}_x \sum_{i=1}^{#J} w_i J_i \quad (20)$$

$$J = \left\{ C_D, \frac{1}{n} \sum_{y=0}^{b/2} (C_L^t(y) - C_L^a(y))^2, \tau_{SP}, \tau_{DR}, \tau_{PH} \right\} \quad (21)$$

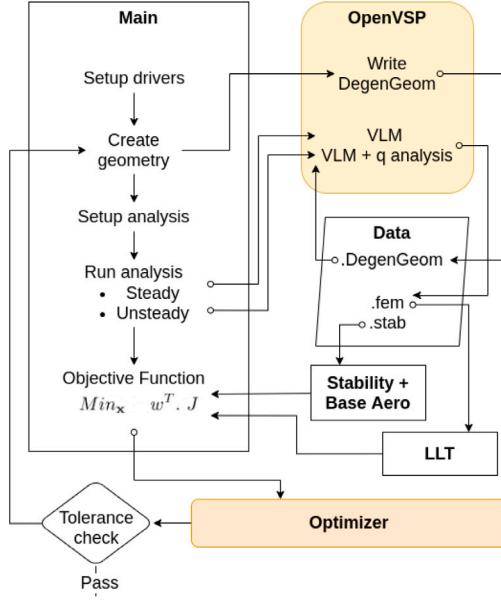


Fig. 6. Optimization architecture.

$$x = [AR, \Theta_{rel}, \lambda, \Lambda_{LE}] \quad (22)$$

$$C_L = 0.9C_l^c \quad (23)$$

$$S.M. = 4\% \quad (24)$$

$$C_M \approx 0 \quad (25)$$

$$C_L^t(y) = \text{Target lift} \quad (26)$$

$$C_L^a(y) = \text{Actual lift} \quad (27)$$

$$5 \leq AR \leq 10 \quad (28)$$

$$-1 \leq \Theta_{rel} \leq -5 \quad (29)$$

$$0.4 \leq \lambda \leq 0.9 \quad (30)$$

$$20 \leq \Lambda_{LE} \leq 30 \quad (31)$$

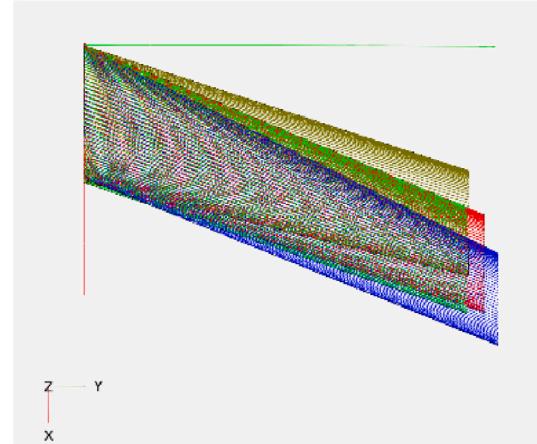
### 7.2.3. Architecture

Python was chosen as the scripting language to minimize the cost function with appropriate weights using a Particle Swarm Optimization algorithm. The libraries/software used for the design have a well exposed API that helps create custom loops and carry out efficient numerical computation using multiple processes. A detailed analysis for such an interface selection is done in [13]. The geometry is arrived upon by interfacing a python script with *OpenVSP* and conducting a Vortex Lattice method analysis. The process loop for the same is shown in Fig. 6.

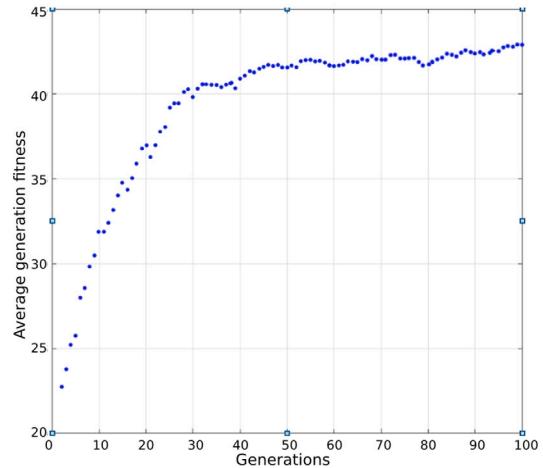
### 7.2.4. Results

The particle swarm optimizer is run for a total of 100 generations, with the fitness increasing and then saturating over time 7(b). Some design trade-offs in *OpenVSP* are shown in Fig. 7(a). Since the code does not address the problem of structures and aeroelasticity, an appropriate and feasible design is chosen with respect to design experience. The geometry for the chosen design is given in Table 2. A factor of safety during weight estimation is also considered for this. Some comments regarding the tools are written to help the reader understand the available sophistication level and possible caveats.

(a) Vortex lattice method works well for lift analysis, but it is not an accurate method for determining the overall drag. Therefore the values are further verified by CFD analysis in 8.2.



(a) Optimization Trade-offs



(b) Average fitness vs. Generations

**Table 2**  
Aircraft parameters.

Parameter	Values
Wing loading	164.88 N/m <sup>2</sup>
Power loading	3.47 kg/kW
Wing area	0.418 m <sup>2</sup>
Aspect ratio	7.35
Taper ratio	0.689
Wing incidence	5°
Relative twist	-2°
Leading edge sweep	26.5°

(b) The focus of the geometry optimization is primarily for the cruise phase (therefore conducted at the cruising angle of attack), as this will have the most significant effect on performance.

(c) Since this is not a full-scale MDO solution, a human in the loop is necessary to guide and remove suboptimal/non-convex geometries.

### 7.3. Propulsion

A propulsion architecture is conceived to drive the aircraft at its optimal performance with the maximum range and/or endurance.

#### 7.3.1. Forward motors

The forward motors are essential for the cruising segment of the flight. The right combination of propeller size, pitch and the motor

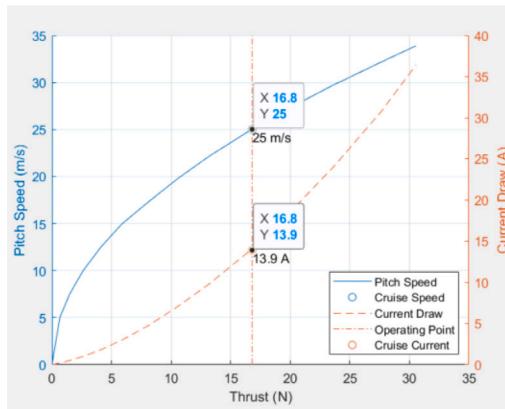


Fig. 8. Forward motor optimal thrust.

combination was combed through in Eqs. (35)–(39) to achieve the required performance of the motors while cruising at 25 m/s. This was matched with the pitch speed of the motor operating on the thrust needed to overcome drag at cruise (Eq. (36)) and hence maximizing endurance. The optimal thrust curve in Fig. 8 explains this method graphically.

The initial design of our fixed-wing yielded a cruise drag  $C_d^c$ , which would have to be overcome by the forward cruise motors. We keep achievable ranges of propeller pitches ( $x_{pitch}$  Eq. (38)) that were available in local markets to match our requirements. The cruise RPM ( $\Omega_{cruise}$  (32)) is taken to calculate the motor KV.

A scaling factor  $\Phi_c$  is introduced in Eq. (34) keeping in mind that the required  $\Omega_{cruise}$  (Eq. (32)) increases quadratically with the required thrust for a constant pitch propeller powered by a Brush-Less Direct Current (BLDC) motor (37). This factor scales the motor RPM at cruise to what its RPM would likely be at  $T_{cruise}$  and  $T_{max}$ . Note that the calculations and ratios in this section include the combined thrust of both forward motors.

Eq. (39) defines safety margins of required thrust for cruise motor selection, out of which design changes would be necessary.

$$\Omega_{cruise} = \frac{60 v_{pitch}}{0.0254 x_{pitch}} \quad (32)$$

$$KV = \left[ \frac{\Omega_{cruise}}{V} \Phi_c \right] \quad (33)$$

$$\Phi_c = \sqrt{\frac{T_{max}}{T_{cruise}}} \quad (34)$$

where,

$$v_{pitch} = 25 \text{ m/s} \quad (35)$$

$$T_{cruise} = \frac{1}{2} \rho v^2 S C_d^c \quad (36)$$

$$V = 22.2 \text{ V}(6\text{S} \text{Battery}) \quad (37)$$

$$10.16 \text{ cm} \leq x_{pitch} \leq 20.32 \text{ cm} \quad (38)$$

$$0.6 \leq \left( \frac{T}{W} \right)_{max} \leq 0.8 \quad (39)$$

BLDC motors operating at 22.2 V (6S) battery packs were searched, which could give the required thrust of 17 N at an altitude of 500 meters and a pitch speed of 25 m/s. EMAX MT3515 650 KV motor coupled with a 25.4 cm (10 in.)  $\times$  15.24 cm (6 in.) propeller was finalized after iterations of motor's various KV (RPM/voltage) ratings. The maximum load that the selected motor can handle is 26 N of thrust — comfortable for flying at a max T/W of 0.74. This distribution can easily climb with a thrust margin of 1.48:1 and therefore with a ratio of 1:1:2, which aligns with our assumptions in Section 5.

**Table 3**  
Battery data.

Parameter	Values
Form factor	18 650
Cell voltage (nominal)	22.2 V
Cell capacity	3300 mAh
Cell discharge	3C (9.9 A)
Cell discharge (Peak)	4C (13.2 A)
Weight (cell)	48 g

**Table 4**  
Final motors selection.

Forward motor (x2)	Value
EMAX MT3515	
Type	BLDC 22.2 V
KV rated	650 RPM/V
Maximum power	630 W
Propeller	25.4 cm $\times$ 15.24 cm
Max thrust	2660 g
Weight	131 g
Rear motor (x1)	
T-MOTOR MN501-S	
Type	BLDC 22.2 V
KV rated	360 RPM/V
Maximum power	1000 W
Propeller	50.8 cm $\times$ 15.24 cm
Max thrust	5372 g
Weight	171 g

### 7.3.2. Battery selection

Observing modern trends in electric propulsion, Lithium-Ion battery packs prove to be the best energy density as they can hold more amperage than their Lithium Polymer counterparts. Lithium-Ion battery packs do have their drawbacks of limited current discharge. To confirm before selecting, the current draw was evaluated for all flight segments and was found low enough to use Li-ion batteries.

A theoretical flight range ( $R_{th}$ ) of 107 km and cruise endurance ( $E_{cruise}$ ) of 71 min were subsequently calculated by estimating the battery usage in cruised flight. Battery weight is estimated to be 2880 g at a 6S 10P configuration. Here  $C_{cell}$  is battery capacity of one cell given in Table 3.

$$\eta_{total} = \eta_{propeller} \eta_{motor} \eta_{battery} \quad (40)$$

$$E_{cruise} = \frac{C_{cell} n_{parallel}}{I_{cruise} n_{motors}} \eta_{total} \quad (41)$$

$$R_{th} = v_{cruise} E_{cruise} \quad (42)$$

### 7.3.3. Rear motor

With the forward motors handling half of the thrust required during hover segments, the rear motor handles the remaining. Its only goal is to draw the least amount of current to be as efficient as possible in hover. Motors with larger propeller size, lower pitch, and lower RPM (KV rating) draw lesser current compared to their short as the induced pitch speed is lesser than those with smaller length, higher pitch, and higher RPM [21]. The rear propeller was finalized by limiting the propeller disk diameter such that it does not intersect with the fuselage and has enough disk loading to keep the thrust as close to the theoretical value. The size of 50.8 cm(20 in.) for the propeller hence limited the motor KV range between 330 to 390 KV, and the appropriate motor was sourced from local markets (Table 4).

The collective propulsion results are also validated on *eCalc*<sup>9</sup>: a paid propulsion selection tool. The results are in close agreement with the developed formulae.

<sup>9</sup> <https://www.ecalc.ch/>.

#### 7.4. Tilt mechanism

The tilt mechanism needs to be mechanically robust to rotate and transfer the thrust and torque from drive system aiding in transition between different flight modes. It is required to rotate 90 degrees during cruise and further 10 odd degrees during vectored yaw in tri-copter mode. Initially, a gearing mechanism was chosen over a push rod tilting mechanism due to its high efficiency during rotation. The pressure angle of spur gears was chosen 20 degree, a standard value because of its stronger tooth quality, higher load carrying capacity and its greater length of contact [22]. The gear ratio of gears was taken as 1:1. The minimum number of teeth ( $Z_{min}$ ) were calculated using Eq. (43) to be 17 so as to avoid interference.

$$Z_{min} = \frac{2}{\sin^2 \phi} \quad (43)$$

where,

$\phi$  = Pressure angle

#### 8. Detailed design

The preliminary design outputs a flyable aircraft to satisfy the theoretical performance objectives. More sophisticated software for computational fluid dynamics and aircraft control simulation engines are used at this point to validate the design.

##### 8.1. CAD and pre-processing

The solid model with all internal features is designed on *FreeCAD* (Fig. 9(a)). Extensive use of the part design and draft workbenches allows the creation of complex models and geometries. The commercial tool used for comparison is *SolidWorks*. As shown below, *FreeCAD* can capture most of the geometry, but the tail is off-design due to the lack of sophisticated lofting features with guide curves as compared to *Solidworks* (Fig. 9(b)).

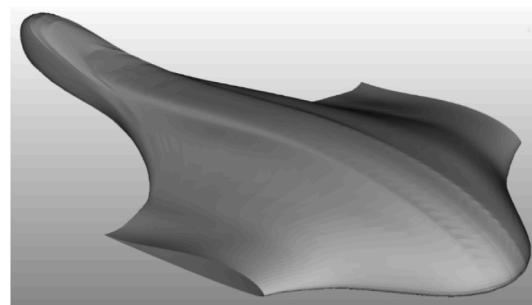
Triangulation for the CFD model is done using the draft, part, and mesh design workbench. Each part and face are sequentially upgraded using the draft upgrade, fused using the 'boolean fuse' option, and finally, the entire fusion is converted to a mesh. Owing to the tedious processing time of a Netgen mesh, the standard mesh was used, and further mesh optimizations and quality improvements were completed in the CFD tool.

##### 8.2. CFD and post-processing

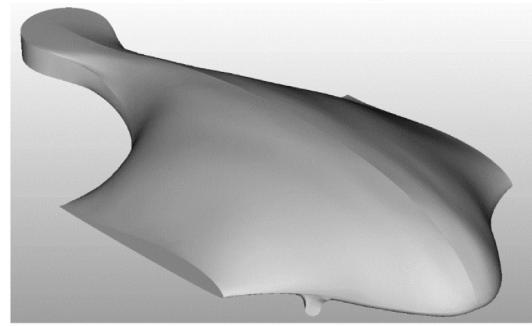
A computational fluid dynamics analysis was set up on OpenFOAM to calculate the lift and drag more accurately than the low fidelity analysis.

For pre-processing, *blockMesh* and *snappyHexMesh* were used with the parameters given in Table 5. Some refinement regions were added in particular areas like the leading edge and winglet-wing blend (Fig. 10). The edge mesh was extracted and refined on *Blender*<sup>10</sup> to remove double points and line intersections. Some parameters were interactively tuned in the mesh quality dictionary. Table 5 shows only the edited values.

The basic steady-state *simpleFoam* solver was used to obtain the coefficients at the cruise state. The solver was run for 500 iterations but resulted in satisfactory residual convergence in 200 iterations (Fig. 11(c)). The comparison of results between a low and high fidelity analysis is summarized in Table 6. There is a marked difference in the glide ratio, which is attributed to the lack of sophisticated viscous drag calculation with algorithms like Vortex Lattice Method (VLM) and Lifting Line Theory (LLT). The CFD analysis outputs a value of 10.6 for  $C_L/C_D$  which is very close to our initial assumption of 11.



(a) FreeCAD Fuselage



(b) Solidworks Fuselage

Fig. 9. Fuselage comparison.

Table 5

Mesh parameters.

Parameter	Value
addLayers	False
Leading edge refinement levels	(1E15 8)
Winglet refinement levels	(1E15 7)
Front motor refinement levels	(1E15 7)
Local refinement box levels	(1E15 5)
implicitFeatureSnap	False
mergeTolerance	1e-6
maxNonOrtho	60
minTwist	-1e+30
minVol	-1e+30
minDeterminant	-1
minFaceWeight	-1
minTriangleTwist	-1

Table 6

Coefficients comparison.

Coefficient	XFLR5	OpenFOAM
CD	0.018	0.038
CL	0.477	0.403

##### 8.3. Render

While rendering is not essential to the design process, it is required in some commercial sectors. Conventionally, it is done using expensive tools and software. Contrarily, we have used *LuxCoreRenderer* as the open-source tool to do the same and included it in the design loop.

The ray-tracing module of *FreeCAD* is used to output a *.lxs* configuration file, which is modified accordingly. Note that the default *.lxs* file template has preset sizes and transformations of area light sources, and one can either change the configuration file or the CAD itself. Our option was to choose the former, highlighting the freedom offered by open-source software. The changes to the default configuration are given. The entire render with a final noise pass error of about 0.001 took about 15 min to complete on an *Intel i7* 7th gen laptop CPU. The configuration used for the *LuxCoreRender* file is given in Fig. 12.

<sup>10</sup> <https://www.blender.org/>.

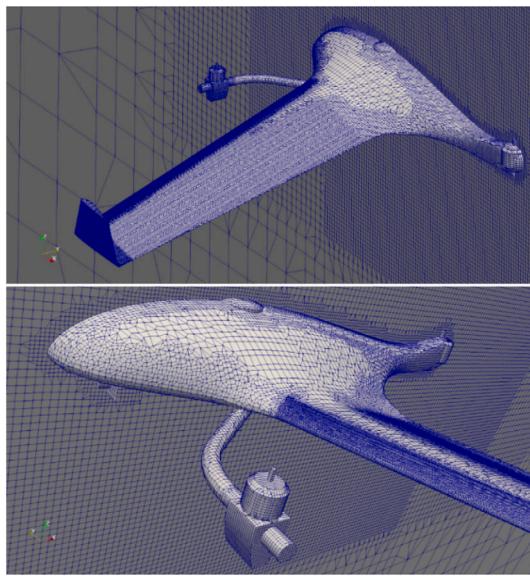
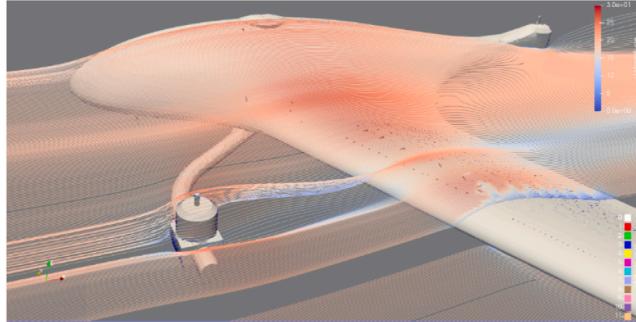


Fig. 10. Snappy Hex Mesh results.

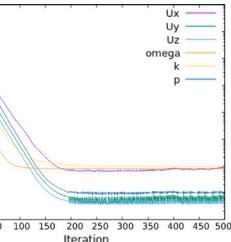
```

1. Camera
2. LookAt
42.9652 27.7823 39.252
-7.02438 -22.438 -0.599087
-0.336994 -0.35615 0.871547
3. Flex image
4. "integer xresolution" [1080]
5. "integer yresolution" [1360]
6. Groundplane
7. Transform
[4.000000000000000 0.000000000000000
0.000000000000000 0.000000000000000
0.000000000000000 4.000000000000000
0.000000000000000 0.000000000000000
0.000000000000000 0.000000000000000
4.000000000000000 0.000000000000000
0.000000000000000 0.000000000000000
0.000000000000000 4.000000000000000]
8. Area Lightsource : sun
9. Transform
[15.000000000000000 0.000000000000000
0.000000000000000 0.000000000000000
0.000000000000000 15.000000000000000
0.000000000000000 0.000000000000000
0.000000000000000 0.000000000000000
15.000000000000000 0.000000000000000
0.000000000000000 0.000000000000000
40.000000000000000 1.000000000000000]
10. "float power" [200.000000000000000]
11. "color L" [0.5000 0.9536 0.7063]
```

Fig. 12. LuxCore render configuration.



(a) Streamlines coloured by velocity



(b) Pressure contours

(c) Residuals

Fig. 11. CFD results.

A comparison between *SolidWorks + PhotoView 360* and *FreeCAD + LuxCore* is shown in Figs. 13(a) and 13(b). As shown, both the software output remarkable results with marginal differences.

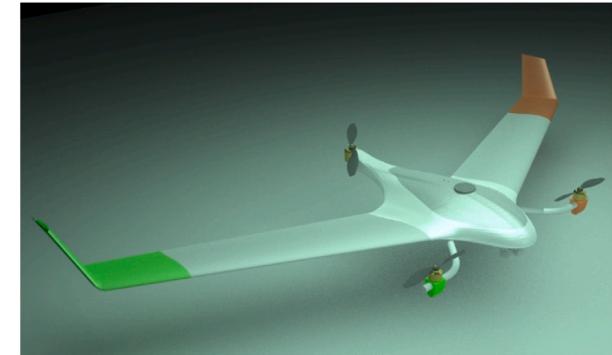
## 9. Development and integration

### 9.1. UAV fabrication

A prototype body is fabricated to validate some performance and stability characteristics of the aircraft. The body is made of Extruded polystyrene foam (XPS), and glass fiber fabric through a vacuum-assisted wet layup method. Some aspects of the configuration were altered depending on the resources available. The fuselage is made by hot-wiring the foam along each orthographic projection plane. This



(a) Solidworks + PhotoView Render



(b) FreeCAD + LuxCore Render

Fig. 13. Render comparison.

method was highly efficient in terms of time and ensured minimum sanding of the foam to get the final shape (Fig. 15(a)). Aluminum pipes were used to define the structure of the tri-copter assembly, and the foam blended wing body was integrated along with it (Fig. 15(d)). An in-depth analysis and comparison of the materials used and their rationale is beyond the scope of this paper but has been completed in [23].

The tilt-rotor system for the UAV was a highly sensitive component to design and fabricate. Initially, a geared system was designed and printed out of ABS plastic (Fig. 14(b)). The calculations of the required

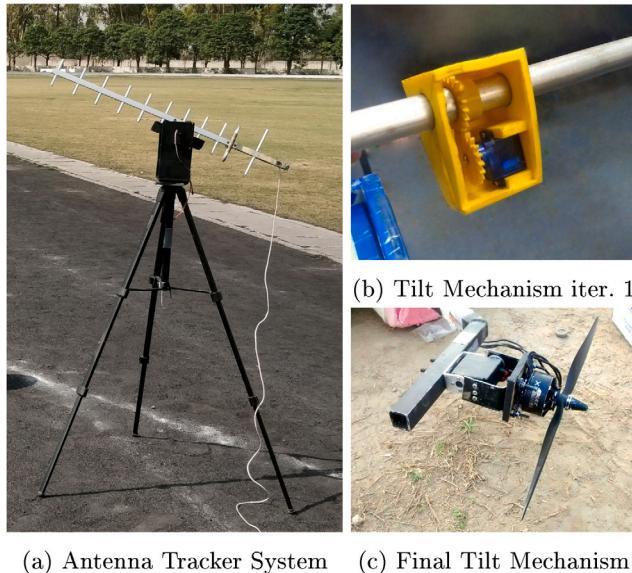


Fig. 14. Mechanical subsystems.

servo torque were satisfied theoretically, and the mechanism was also bench-tested successfully at the motor's burst thrust. However, it was found that on flight testing the UAV, the mechanism kept failing due to the vibrations and persistent servoing by the autopilot to maintain stability. After trying out many iterations, it was found that the gears in the mechanism could not be created by the 3D printer at our disposal with sufficient accuracy so as to resist the dynamic forces. The mechanism had to be replaced by a more expensive but robust pre-fabricated tilt servo, as shown in Fig. 14(c).

To track the UAV autonomously and ensure a proper line of communication from the ground control station, an antenna tracker system (ATS) was also developed (Fig. 14(a)). The ATS has two axes (Yaw and Pitch) made from Medium Density Fiberboard (MDF). The system is actuated by high torque servos and can be controlled using a transmitter.

## 9.2. Subsystems and integration

A well-defined architecture is shown in Fig. 16 highlighting data flow between the subsystems. A brief description of some of the components is given:

- (a) *Pixhawk 2 Cube*: An autopilot is necessary to carry out custom missions and provide additional stability during flight. It also allows the UAV to maintain a level altitude while transitioning into a cruise state.
- (b) *Here GPS*: A GPS is used to localize the UAV in its environment.
- (c) *RFD 900x*: It is used to create a link between the ground systems and the UAV. To plan missions dynamically with Mission Planner, this link is necessary.
- (d) External Sensors: These include everything ranging from the Battery Eliminator Circuits (BEC) and airspeed sensors.
- (e) *GitUp Git2P*: An on-board electro-optical camera selected for its FOV required in the mission.
- (f) *Tarot T4 3D*: A 3-Axis gimbal used to mount and stabilize the imaging sensor during flights.
- (g) *Nvidia Jetson Nano*: The onboard computer's primary purpose is, to carry out the image recognition task among other necessary mission scripts.
- (h) *TP-Link 2.4 GHz Wi-fi Modules*: These Wi-fi modules are responsible for publishing state information to other agents in the ad-hoc mesh network.

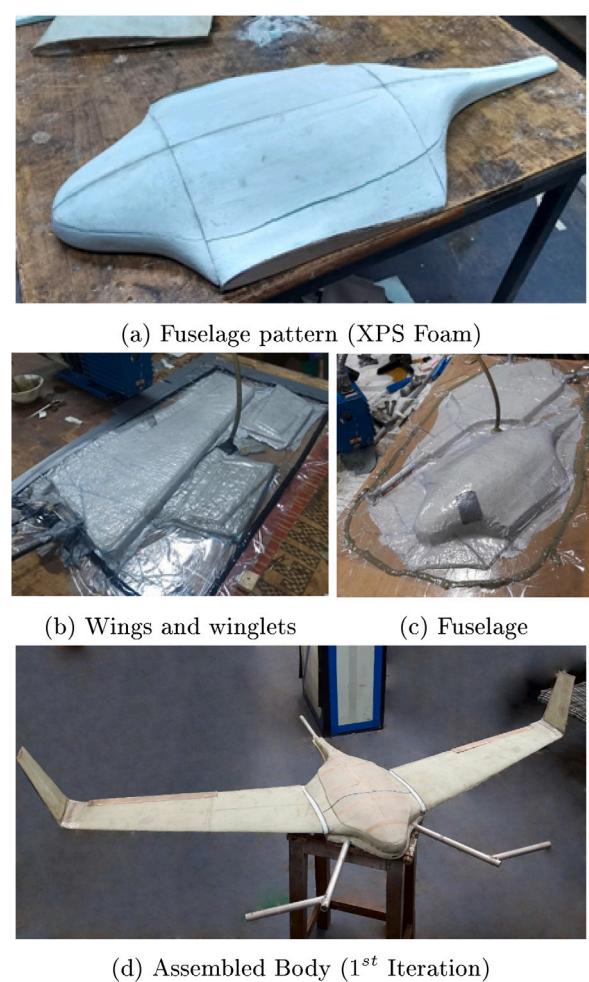


Fig. 15. Vacuum Wet-Layup fabrication.

**Table 7**  
Tuned flight parameters.

Parameter	Original	Changed
ARSPD_FBW_MAX	30	28
ARSPD_FBW_MIN	10	16
PTCH2SRV_D	0.200000003	0.04
PTCH2SRV_P	2.5	1
Q_FRAME_CLASS	1	7
Q_TILT_YAW_ANGLE	20	30
Q_A_RAT_PIT_P	0.25	0.2
Q_A_RAT_YAW_P	0.1800000072	0.25
Q_A_THR_MIX_MAN	0.1000000015	0.5

A key feature is the form of the modular propulsion battery cells. Multiple Lithium-Ion cells can be used to form arrangements so as to fit into the body (Fig. 17). The blended design offers a lot of payload volume, which is exploited by stacking nine parallel sets of six cells, each connected in series. Some battery packs are strategically placed very close to the CoG, allowing the user to swap the pack for additional payloads or sensors. The integration process is shown in Fig. 18.

## 10. Results and discussions

### 10.1. Flight testing

The final prototype (Fig. 19) is tested physically over a few flight tests to evaluate flight dynamics and tune the autopilot. To tune

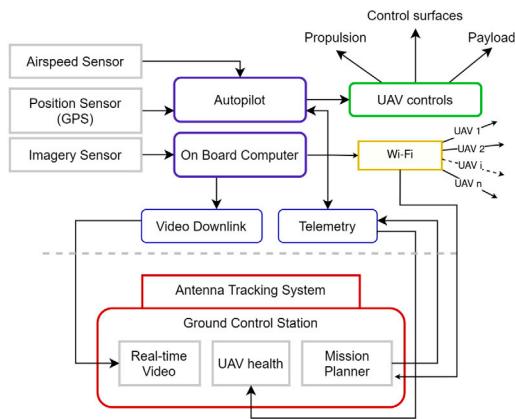


Fig. 16. System architecture.

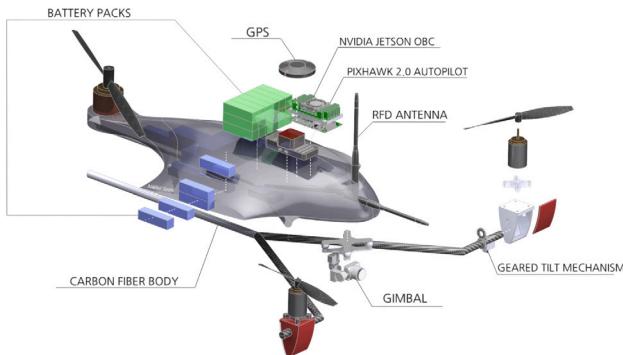


Fig. 17. Exploded view of subsystems.



Fig. 18. Integration.

the Pixhawk autopilot, ArduPlane's VTOL firmware is combined with Mission Planner. The Plane is successfully tuned in both hover and cruise modes and some important parameters with respect to the transition and stability are given in Table 7. Various parameters were tweaked to achieve stability. Maximum and minimum air-speeds (AR-SPD\_FBW\_MAX and MIN) during Fly-By-Wire mode were limited to the fixed-wing's flight envelope. PTCH2SRV\_D was lowered to 0.04 as the earlier value resulted in higher oscillations in the cruise phase. The Q\_TILT\_YAW\_ANGLE was increased from 20 to 30 as our forward motors were closer to the center of gravity and therefore had a shorter moment arm for vectored yaw. The increase in Q\_A\_RAT\_YAW\_P resulted in reduced overall vibrations. To avoid damage to wings Q\_A\_THR\_MIX\_MAN was increased as it now prioritized attitude control over throttle.

However, further testing resulted in many unforeseen circumstances and truly revealed a VTOL system's complex and dynamic behavior. A



Fig. 19. Final prototype.

flight video has been accompanied in the supplementary material (S5) representing this behavior. Some discussions and failures with respect to the same are done below:

- The hover mode is highly stable with a fast rate of climb as calculated in the propulsion section.
- The forward speed at about 45% throttle reached about 20 m/s, matching the expectation.
- It was noticed that whenever the aircraft's speed increased, it became susceptible to even the slightest of disturbances in vectored yaw motion of the tilt mechanism. This made the aircraft turn on its axis, and the high stall speed (due to high wing loading) made a recovery from such an event nearly impossible.
- The above effect became even more pronounced in our case, where the cost and quality of 3D printed materials used in the tilt mechanism were meager due to the constrained budget.
- Some electronic failures such as interference and the overheating of components were also observed in certain flights. This is attributed to the higher current draw of the motors as compared to a normal fixed-wing or multi-rotor UAV.
- The lack of prior experience and exposure in handling or operating a hybrid craft made it very challenging as there was no obvious error to debug. The crashes occurred due to a combination of mechanical and electronic failures.
- Even post debugging the problem, finding the right solution was a tedious task due to the vast number of PID parameters and mechanical factors to tune. A significant help at this time was the shared experience of the community at the Arduplane forums.

As a result, a complete range test to evaluate performance became infeasible as the cost of repairs and extra parts exceeded the budget over time. However, the authors were able to set up a large number of good practices and organized debugging routines for VTOL aircraft to establish a foundation for future work in the same laboratory.

## 10.2. Swarm testing

Testing of the swarm communication architecture and algorithms was decoupled with that of the VTOL UAV. This was done to make debugging easier. There were two modes of testing completed:

- Fixed-wing: To test problems like the scalability of velocity with the number of UAVs and communication range, individual Sky-Surfer v4 model kits were used. (Fig. 20).
- VTOL: To test collective intelligence algorithms and subsystems that were independent of aerodynamic testing (eg. payload drop, ground communication link, formation flying etc.), swarm testing was also carried out in parallel on DJI 450 quad-copters (Fig. 21). The methodology and results have been published by the software team in [24], and are not covered here for the sake of brevity.



Fig. 20. SkySurfer v4 models used for swarm.



Fig. 21. Formation flying on DJI 450.

### 10.3. Open source software

While designing, particular care was taken that no commercial tools or codes be used. Using completely open-source tools has its own sets of challenges and advantages. Some of these are discussed below:

- Open-source tools are not without their limitations. Therefore, a significant effort was required to get acquainted with all tools at every stage and produce meaningful results.
- A key difference that tends to take up significant time is the lack of standardized coordinate systems, conventions of unit systems, and a general understanding of concepts among different software. This is because open-source software, while guided by some core teams, are collaborative. This results in different conventions on each platform. An effort from the user's side is required to understand this point and change models/configurations accordingly.
- The above point becomes more important when there is a significant lack of documentation, discussion forums, and code contributors. This can be overcome by having multiple specialists work in a team to avoid learning conventions and background theory for two completely different software.
- The issue of 'accuracy' of results was verified collectively by reading up on literature and conducting some of the analyses parallelly on commercial software and calculating errors.

## 11. Outcome

The work in this paper can be summarized into the following tangible outcomes:

1. An aircraft design framework was conceptualized and used in practice to design a VTOL-FW UAV using a mix of traditional tools like aircraft constraint analysis and modern multidisciplinary design methods.
2. The primary drivers of this design are the geometry optimization and propulsion architecture which were both developed specifically to solve the problem in this paper.
3. Both high and low fidelity tools were compared in the detailed design stage to give an estimate of possible shortcomings and advantages of using low-fidelity open-source software.
4. A number of parameters and configuration values with respect to each design software (PSO, CFD, rendering) were corroborated and presented in each section.
5. The UAV was fabricated using innovative manufacturing methods and tuned using ArduPilot. The tuned parameters and their explanations were also presented.
6. An in-depth discussion was followed, which gives insight into the design process, shortcomings, and merits of using the framework conceived in this paper.

## 12. Conclusion

In this paper, a VTOL-FW hybrid unmanned aerial system with a 100 km theoretical range and payload carrying capabilities was designed and fabricated. Some flight testing and tuning results were also corroborated, and the challenges were discussed for the same in the previous sections. The discussions in the above sections show the discrepancy between theoretical and experimental results, especially for VTOL systems. Such gaps need to be filled with better tools, further research into the causes of these differences and targeted flight-time practices to address the same. Aircraft design will always be a multidisciplinary task where no linear approach exists. When equipped with suitable sensory payloads, hybrid systems like VTOL aircraft can boost target acquisition capabilities, use AI to take informed decisions, and even show collaborative behavior with homogeneous swarming agents. The authors hope to advance research for such generalized intelligent systems to solve problems in various fields like defense, agriculture, and humanitarian search and rescue.

### CRediT authorship contribution statement

**Nikhil Sethi:** Conceptualization, Methodology (systems, optimization, aircraft sizing, CAD, CFD, rendering, integration, flight testing), Writing – original draft, Software. **Saurav Ahlawat:** Conceptualization, Methodology (systems, propulsion, integration, flight testing), Writing – original draft, Formal Analysis.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgments

The fabrication of the UAV in this paper was completed at the Unmanned Aerial Systems laboratory at Delhi Technological University (UAS-DTU), and the authors acknowledge the help of the team members for the same.

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.array.2022.100183>. It contains system design procedures, configuration trade-offs, data for regression analysis, and videos.

## References

- [1] Gu Haowei, Lyu Ximin, Li Zexiang, Shen Shaojie, Zhang Fu. Development and experimental verification of a hybrid vertical take-off and landing (VTOL) unmanned aerial vehicle (UAV). In: 2017 international conference on unmanned aircraft systems (ICUAS). IEEE; 2017, p. 160–9.
- [2] Cyba Roman, Lemanowicz Marcin, Gorol Zbigniew, Kudala Tomasz. Construction prototyping, flight dynamics modeling, and aerodynamic analysis of hybrid VTOL unmanned aircraft. *J Adv Transp* 2018;2018.
- [3] Dündar Özgür, Bilici Mesut, Üner Tarık. Design and performance analyses of a fixed wing battery VTOL UAV. *Eng Sci Technol Int J* 2020;23(5):1182–93.
- [4] Hadi Ghozali Suharyanto, Kusnaedi Muhammad Ramadhan, Dewi Puspita, Budiyarto Aris, Budiyono Agus. Design of avionics system and control scenario of small hybrid vertical take-off and landing (VTOL) UAV. *J Instrum Autom Syst* 2015;2(2):67–71.
- [5] Botero Emilio M, Alonso Juan J. Conceptual design and optimization of small transitioning uavs using SUAVE. In: 18th AIAA/ISSMO multidisciplinary analysis and optimization conference. 2017, p. 4149.
- [6] Ozdemir Ugur, Aktas Yucel Orkut, Vuruskan Aslihan, Dereli Yasin, Tarhan Ahmed Farabi, Demirbag Karaca, Erdem Ahmet, Kalaycioglu Ganime Duygu, Ozkol Ibrahim, Inalhan Gokhan. Design of a commercial hybrid VTOL UAV system. *J Intell Robot Syst* 2014;74(1):371–93.
- [7] Clarke Matthew, Smart Jordan, Botero Emilio M, Maier Walter, Alonso Juan J. Strategies for posing a well-defined problem for urban air mobility vehicles. In: AIAA scitech 2019 forum. 2019, p. 0818.
- [8] Böhnke Daniel, Rizzi Arthur, Zhang Mengmeng, Nagel Bjoern. Towards a collaborative and integrated set of open tools for aircraft design. In: 51st AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition. 2013, p. 222.
- [9] Jasak Hrvoje. OpenFOAM: open source CFD in research and industry. *Int J Nav Archit Ocean Eng* 2009;1(2):89–94.
- [10] He Ping, Mader Charles A, Martins Joaquim RRA, Maki Kevin J, Dafoam: An open-source adjoint framework for multidisciplinary design optimization with openfoam. *AIAA J* 2020;58(3):1304–19.
- [11] Sun Luning, Gao Han, Pan Shaowu, Wang Jian-Xun. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Comput Methods Appl Mech Engrg* 2020;361:112732.
- [12] Segui Marine M, Ghazi Georges, Botez Ruxandra M, Thompson Emily M. Design, development and validation of a cessna citation x aerodynamic model using openVSP software. In: 2018 modeling and simulation technologies conference. 2018, p. 3256.
- [13] Herbst Sebastian, Staudenmaier Adrian. Using OpenVSP in a conceptual aircraft design environment in MATLAB. In: Congress of the international council of the aeronautical sciences. 2016.
- [14] Sadraey Mohammad H. Aircraft design: a systems engineering approach. John Wiley & Sons; 2012.
- [15] Tyan Maxim, Van Nguyen Nhu, Kim Sangho, Lee Jae-Woo. Comprehensive preliminary sizing/resizing method for a fixed wing–VTOL electric UAV. *Aerospace Sci Technol* 2017;71:30–41.
- [16] Serrano A Ramirez. Design methodology for hybrid (VTOL+ Fixed Wing) unmanned aerial vehicles. *Aeronaut Aeros Open Access J* 2018;2(3):165–76.
- [17] Raymer Daniel P. Aircraft design: a conceptual approach (AIAA education series). Reston, Virginia; 2012.
- [18] Dagur Rishabh, Singh Vikrant, Grover Shabir, Sethi Nikhil, Arora BB. Design of flying wing UAV and effect of winglets on its performance. *Int J Emerg Technol Adv Eng* 2018;8(3).
- [19] Miranda Lester James. Pyswarms: a research toolkit for particle swarm optimization in python. *J Open Source Softw* 2018;3(21):433.
- [20] Chih Mingchang, Lin Chin-Jung, Chern Maw-Sheng, Ou Tsung-Yin. Particle swarm optimization with time-varying acceleration coefficients for the multidimensional knapsack problem. *Appl Math Model* 2014;38(4):1338–50.
- [21] Sahin Huseyin, Oktay Tugrul. Powerplant system design for unmanned tricopter. *Eurasia Proc Sci Technol Eng Math* 2017;(1):9–21.
- [22] Mobley R Keith. Gears and gearboxes. In: Plant engineer's handbook. Elsevier; 2001, p. 629–37.
- [23] Sethi Nikhil, Chauhan Prabhash, Bansal Shashwat, Singari Ranganath M. Robust vehicle development for student competitions using fiber-reinforced composites. In: Advances in manufacturing and industrial engineering. Springer; 2021, p. 61–76.
- [24] Agrawal Anuj, Gupta Aniket, Bhowmick Joyraj, Singh Anurag, Nallanthal Raghava. A novel controller of multi-agent system navigation and obstacle avoidance. *Procedia Comput Sci* 2020;171:1221–30.