Low-fidelity Design Optimization and Development of a VTOL Swarm UAV using an Open-source Framework

Nikhil Sethi (✉ sethi.nirvil@gmail.com)

Delhi Technological University <https://orcid.org/0000-0002-0672-4654>

Saurav Ahlawat

Delhi Technological University

Research Article

Keywords: Drones, Aircraft design, PSO, MDO, FOSS, Hybrid Tricopter, CFD

Posted Date: January 20th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-177910/v2>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Low-fidelity Design Optimization and Development of a VTOL Swarm UAV using an Open-source Framework

Nikhil Sethi^{a,*}, Saurav Ahlawat^b

^aDepartment of Mathematics and Computer science,
South Asian University,
New Delhi, India

^bDepartment of Aerospace Engineering,
University of Illinois, Urbana-Champaign,
Illinois, USA

ARTICLE INFO

Keywords:

Drones
Tilt rotor
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. Rapid prototyping of such systems at the student level is a challenge because commercial software is expensive and difficult to interlink with other tools for creating a multi-disciplinary design. In this paper, an all-electric unmanned aerial system with both Vertical Takeoff and Landing (VTOL) and Fixed-Wing (FW) capabilities is designed and optimized for long range surveillance and relief operations. The entire design and simulation pipeline from concept to render is done using completely open-source software, libraries and in-house code. The Unmanned Aerial Vehicle (UAV) is equipped with on-board computers and sensors, and is designed to operate in an ad-hoc WiFi network with a swarm of similar UAVs. A prototype is fabricated, integrated and tested as well. A comparison with commercial codes and programs is done parallel to each design step. The study is concluded by discussing the primary challenges of using free software and 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. The use of 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 helps in learning from scratch, and the leverage to create and use such tools has never been greater.

Finally, a complete end-to-end solution from conceptual design to autopilot testing is rarely completed in academic environments due to 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 be beneficial to the emerging field of unmanned aerial systems and swarm robotics. In this paper, we recognize the above challenges and attempt to alleviate them by developing a sound methodology spread across a wide variety of completely free tools and in-house code. Our research is aimed primarily to help students and upcoming researchers developing feasible designs for student competitions or learning to design in general. Formally, the rationale of our research is centered around the following outcomes:

a) A low-fidelity design methodology 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 so as 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.

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 a establishing a custom open-source design pipeline in Section 3. Sections 4 to 8 go through the entire design

*Corresponding author

sethi.nirvil@gmail.com (N. Sethi)

ORCID(s): 0000-0002-0672-4654 (N. Sethi);

0000-0002-3826-6301 (S. Ahlawat)

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. To conclude, some final comments and discussion on the design, methods and platforms used is also carried out. Note: This research extensively uses the term “hybrid” which is not to be confused with hybrid electric propulsion which relies on the use of gasoline powered generators for the production of electric power. Instead, “hybrid” here would only refer to the operational capabilities of the aircraft as both a multi-rotor and a fixed-wing craft unless stated otherwise.

2. Background

A VTOL fixed-wing UAV has the ability to take off and land vertically along with sustained cruised flight. These kind of hybrid UAVs combine VTOL capability with the standard forward propulsion of a fixed-wing UAV. In many hybrid VTOL UAVs, rotary lift propellers are typically incorporated into the aircraft’s wings, which then transition for 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 is needed, making the aircraft maintain a fixed position for a period of time.

A study of [5], covers the use of SUAVE, a software used to analyze, optimize, and design a range of small UAVs which has been shown to be 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 which take precedence during a particular optimization while taking a novel weight build-up approach. This work simplifies the larger study of optimizing sensitivity of constraints while removing implicitly resolved parameters alongside reducing computational cost. Inculcation of stable open-source tools to address problems in aircraft design is discussed in [8].

In this work, a diverse set of tools and resources were used in the design and development and a brief description of each and their capabilities is highlighted below:

- a) *FreeCAD*¹: It is a powerful open-source solid modelling software written in C++ with a highly interoperable Python API. It has multiple workbenches to deal with different kinds of geometry. Being

a relatively old software, the documentation is sufficient and has a well-maintained forum.

- b) *OpenFoam*² + *ParaView*³: 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 which is a versatile post processing software to visualise the computed fields. OpenFOAM is well-regarded in the research community and has been linked with a number of applications to further it’s development [10][11]
- c) *LuxCoreRender*⁴: This rendering software provides designers with a powerful tool to create realistic and accurate images. There are many standard features like configuration file systems and acceleration with a CUDA backend. It is very flexible and does not compromise on quality and physical correctness by making ‘unbiased’ designs using equations that model the flow of light.
- d) *OpenVSP*⁵: It is a parametric aircraft geometry tool where the user is allowed to create a 3D model of an aircraft defined by common engineering parameters. This model can be processed into formats suitable for engineering analysis. This software was released as an open-source project under the 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*⁶: 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 on a 3D Panel Method. It is very popular among students and novice researchers as it has a reasonable learning curve and an interactive GUI.
- f) *ArduPilot SITL*⁷ + *Mission Planner*⁸: 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 behaviour of the code without implementation on the hardware itself. The SITL uses the sensor data coming from various flight dynamics models in a flight simulator. This allows ArduPilot SITL

²<https://github.com/OpenFOAM/OpenFOAM-dev>

³<https://github.com/Kitware/ParaView>

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

⁵<https://github.com/OpenVSP/OpenVSP>

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

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

⁸<https://ardupilot.org/planner/>

¹<http://www.freecadweb.org/>

Figure 1: High level design process

to be tested on a very wide variety 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 along with the design process are shown in fig 1.

The high degree of interrelation between disciplines in aircraft design make it difficult to dodge locally optimal designs developed by intuition. Multidisciplinary Design Optimization (MDO) has therefore 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

For the purpose of this research, we model our aircraft for maximizing its range of flight to a 100

Kms, cruising at 25m/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. The concept of operations for the entire mission profile was also envisioned and a video of the same is given in the supplementary material (S1)

4.1. Configuration Selection

There are many configuration parameters to consider and the trade-offs are carried out using basic Pugh matrix analyses which are given in supplementary material (S2). These matrices are filled with relative 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 stabilise the yawing moment. The rear motor is operational for the multi-rotor segment of 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.

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

Here, $(C_{Do})_w$, is the total zero lift drag of the wing, $(C_{Do})_{ht}$ and $(C_{Do})_{vt}$ is the horizontal and vertical tail zero-lift drags respectively and $(C_{Do})_{vtol}$ accounts for the drag in extra mechanisms i.e. motors and structural components responsible for the vertical takeoff and transition to cruise flight. 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 and 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 $(C_{Do})_{vtol}$ component. The extra ‘dead’ weight 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 a greater number of rotors provide better stability and airworthiness, their cost and weight diminish the advantages in financially constrained environments. A tri-copter configuration was therefore 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.

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 so as to provide sufficient thrust during hover as well as 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 the all the motors and a ratio of 1 : 1 : 2 between the left-front, right-front and rear motors respectively. This distribution was a necessary step 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 finalised 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 is fitted to the data (supplementary material S3a). This trend-line along with the quadratic equation generated is solved analytically (Supplementary material S3b) to give a suitable prediction. It is interesting to note how

the value of b is larger than normally obtained for fixed-wings (≈ 0.7). This is in line with expectation and largely attributed to the extra weight that will be used to carry VTOL systems at the expense of better aerodynamic efficiency.

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

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

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_o) of 6.84 Kg so as to carry 1 Kg of payload (W_p). This estimate is further qualified by including a safety margin of 160g 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 disc 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 which 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 maximising endurance, climbing at a certain rate (ROC), and moving with V_{max} velocity at sea level (SL) altitude respectively. They are derived in [17] and [14]. 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 an educated guess and must be iterated and changed depending on final results.

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

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

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

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

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

The following equations are used to size the total area of the propellers which provide vertical thrust. Similar to the wing loading, a disc loading (DL) is used to characterise 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 (θ_{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]

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

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

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

where,

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

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

Figure 2: Linear programming model

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

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

An estimation of aircraft performance is first carried out on XFLR5 to obtain initial but educated estimates of static stability parameters. The analysis starts with selection of a reflexed airfoil and rudimentary optimization at multiple angles of attacks by observing peak C_l/C_d values (Fig. 4a) and stall qualities (Fig. 4b) at cruising altitude and Reynold's number. 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 (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 placement of the Center of Gravity (CoG) to ensure close to null C_m (pitching moment). The resultant twist in the wing

Figure 3: Wing top view

 (a) C_L/C_D vs. α

 (b) C_L vs. α

Figure 4: XFOil Curves

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 analysed 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 and ease of writing code, a particle swarm optimizer is used [19]. The objective function used for the problem is just the weighted sum of each individual objectives (Eq. (20)). For simplicity all the weights

Figure 5: Preliminary Design

(w) are taken as equal and only have opposite signs based on the minimisation/maximisation of each of the objectives.

Four variables namely the aspect ratio (AR), root-tip twist (Θ_{rel}), taper-ratio (λ) and leading edge sweep (Λ_{LE}) are used to parametrize the problem. The combined objective function accounts for 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 (τ_{SP}), Dutch roll (τ_{DR}) and Phugoid (τ_{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 equations (28) to (31) are sensible values which closely follow trends in supplementary material (S3(a)).

Figure 6: Optimization architecture

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

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

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

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

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

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

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

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

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

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

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

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

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 which helps in creating custom loops and carrying 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 overtime 7b. Some design trade-offs in *OpenVSP* are shown in Fig. 7a. Since the code does not address the problem of structures and aeroelasticity, an appropriate and feasible design is chosen with respect to design experience. A factor of safety during weight estimation is also taken to account for this.

Some comments with respect to the tools are written to help the reader understand the level of sophistication available and possible caveats.

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

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

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

(a) Optimization Trade-offs

(b) Average fitness vs. Generations

Table 1
Aircraft Parameters

Parameter	Values
Wing loading	164.88 N/m ²
Power loading	3.47 Kg/Kw
Wing Area	0.418 m ²
Aspect Ratio	7.35
Taper Ratio	0.689
Wing incidence	5°
Relative Twist	-2°
Leading Edge sweep	26.5°

7.3. Propulsion Selection

To drive the aircraft at its optimal performance with the maximum range and/or endurance, a propulsion architecture is conceived.

7.3.1. Forward motors

The forward motors are important for the cruising segment of the flight. The right combination of propeller size, pitch and motor combination was combed through in Eq. (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 initial design of our fixed-wing yielded a cruise drag C_d^c which would have to be overcome by over 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 (Ω_{cruise} (32)) is taken to calculate the motor KV.

A scaling factor Φ_c is introduced in Eq. (34) keeping in mind the fact that the required Ω_{cruise} (Eq. (32)) increases quadratically with the required thrust for a constant pitch propeller powered by a Brush-Less Direct Current (BLDC) motor operating on a 6S battery with the voltage defined in Eq. (37). This factor scales the motor RPM at cruise to what its RPM would likely be at T_{cruise} and T_{max} .

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,

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

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

Figure 8: Forward Motor Optimal Thrust

Table 2
Battery Data

Parameter	Values
Form factor	18650
Cell Voltage (nominal)	22.2V
Cell Capacity	3300mAh
Cell Discharge	3C (9.9A)
Cell Discharge (Peak)	4C (13.2A)
Weight (cell)	48g

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

$$4'' \leq x_{pitch} \leq 8'' \quad (38)$$

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

BLDC motors operating at 22.2V (6S) battery packs were searched upon which could give the required thrust of 17N at an altitude 500 metres and a pitch speed of 25 m/s. EMAX MT3515 650 KV motor coupled with a 10"x6" propeller was finalized upon after iterations of various KV (RPM/voltage) rating of motors. The maximum load that the selected motor can handle is 26N of thrust - comfortable for flying at 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.

7.3.2. Battery Selection

Observing modern trends in electric propulsion, Lithium Ion battery packs prove to be the best in terms of energy density as they can hold more amperage compared to 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 segments of flight and was found low enough to use Li-ion batteries.

A theoretical flight range (R_{th}) of 107 Kilometers and cruise endurance (E_{cruise}) of 71 minutes was subsequently calculated by estimating the battery usage in cruised flight. Battery weight is estimated to be 2880g

Table 3

Final motors selection

Forward Motor (×2)	Value
EMAX MT3515	
Type	BLDC 22.2V
KV rated	650 RPM/V
Maximum Power	630 W
Propeller	10" × 6"
Max Thrust	2660g
Weight	131g
Rear Motor (×1)	Value
T-MOTOR MN501-S	
Type	BLDC 22.2V
KV rated	360 RPM/V
Maximum Power	1000 W
Propeller	20" × 6"
Max Thrust	5372g
Weight	171g

at a 6S 10P configuration. Here C_{cell} is battery capacity of one cell given in Table 2.

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

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

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

7.3.3. Rear Motor

With the forward motors handling half of the thrust required during hover segments, the remaining is handled by the rear motor. It's only goal is to draw the least amount of current so as 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 [20]. The rear propeller was finalized by limiting the propeller disc diameter such that it doesn't intersect with the fuselage and has enough disc loading to keep the thrust as close to the theoretical value. The size of 20" for the propeller hence limited the motor KV range between 330 to 390 KV and the appropriate motor was sourced from local markets.

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

8. Detailed Design

The preliminary design outputs a flyable aircraft capable of satisfying 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.

⁹<https://www.ecalc.ch/>

8.1. CAD and pre-processing

The solid model with all internal features is designed on *FreeCAD*. 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).

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 optimization and quality improvement was done in the CFD tool.

(a) FreeCAD Fuselage

(b) Solidworks Fuselage

Figure 9: Fuselage Comparison

8.2. CFD and post-processing

A computational fluid dynamics analysis was setup 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 below. Some refinement regions were also added in particular areas like the leading edge and winglet-wing blend. The edge mesh was extracted and refined on *Blender*¹⁰ to remove double points and line intersections. Some parameters were interactively tuned in the mesh quality dictionary. Table 4 shows only the edited values.

¹⁰<https://www.blender.org/>

Table 4
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

Figure 10: Snappy Hex Mesh Results

The basic steady state *simpleFoam* solver was used to obtain the coefficients at cruise state. The solver was run for 500 iterations but resulted in satisfactory residual convergence in 200 iterations (Fig. 11c). The comparison of results between a low and high fidelity analysis is summarised in Table 5. There is a marked difference in the glide ratio and this is attributed to the lack of sophisticated viscous drag calculation with algorithms like Vortex-Lattice method (VLM) and Liftng 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.

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.

(a) Streamlines coloured by velocity

(b) Pressure contours

(c) Residuals

Figure 11: CFD Results**Table 5**
Coefficients Comparison

Coefficient	XFLR5	OpenFOAM
CD	0.018	0.038
CL	0.477	0.403

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 which again highlights 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 minutes to complete on an *Intel i7 7th gen* laptop CPU. The configuration used for the *LuxCoreRender* file is given in Fig. 12.

A comparison between *SolidWorks + PhotoView 360* and *FreeCAD + LuxCore* is shown in Figures 13a and 13b. 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

Figure 12: LuxCore render configuration

(a) Antenna Tracker System (b) Tilt Mechanism iter. 1 (c) Final Tilt Mechanism

Figure 14: Mechanical subsystems

(a) Solidworks + PhotoView Render

(b) FreeCAD + LuxCore Render

Figure 13: Render Comparison

(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 method was highly efficient in terms of time and ensured minimum sanding of the foam to get the final shape (Fig. 15a). 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.15d). An in-depth analysis, comparison of the materials used and their rationale is beyond the scope of this paper but has been completed in [21].

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. 14b). The calculations for the torque required for the servo were satisfied theoretically and the mechanism was also bench tested successfully at the burst thrust of the motors. However, it was found that upon taking the UAV for flight testing, the mechanism kept failing due to the vibrations and continuous re-adjusting of the servo by the autopilot. After trying out many iterations, it was found that the gears in the mechanism could never be created by the 3D printer 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. 14c.

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. 14a). The ATS has two axes (Yaw and Pitch) and is 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) Fuselage pattern (XPS Foam)

(b) Wings and winglets

(c) Fuselage

 (d) Assembled Body (1st Iteration)

Figure 15: Vacuum Wet-Layup Fabrication

- 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 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 on-board 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.

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 9 parallel sets of 6 cells each connected in series. Some of the battery packs are strategically placed very close to the CoG which allows the user to swap the pack for additional payload or sensors.

Figure 16: System architecture

Figure 17: Exploded View of subsystems

Figure 18: Integration

Table 6
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

10. 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 the Pixhawk autopilot, ArduPlane's VTOL firmware is used in combination 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 6. Various parameters were tweaked

Figure 19: Final Prototype

to achieve stability. Maximum and minimum air-speeds (ARSPD_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 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 centre 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 prioritised attitude control over throttle.

However, further testing resulted in many unforeseen circumstances and truly revealed the complex and dynamic the behaviour of a VTOL system. A flight video has been accompanied in the supplementary material (S4) which is representative of this behaviour. 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 which matches with the expectation.
- It was noticed that whenever the speed of the aircraft increased, it became highly sensitive to even the slightest of disturbances in vectored yaw motion of the tilt mechanism. This made the aircraft turn on it's 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 was very low 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 be 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 great 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 overtime. However, the authors were able to setup a large number of good practices and organised debugging routines for VTOL aircraft to establish a foundation for future work in the same laboratory.

11. 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 scalability of velocity with the number of UAVs and communication range, individual SkySurfer v4 model kits were used. (Fig. 20).

Figure 20: SkySurfer v4 Models used for Swarm

- 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 in parallel on DJI 450 quad-copters (Fig. 21). The methodology and results have been published by the software team in [22], and are not covered here for the sake of brevity.

Figure 21: Formation Flying on DJI 450

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. 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. A significant effort was required to get acquainted with all tools at every stage and produce meaningful results.
- A key difference which tends to take up significant time is the lack of standardised coordinate systems, conventions of unit systems, and general understanding of some concepts between each software. This is because open-source software, while guided by some core teams, are collaborative. This results in different conventions on each platform. 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 significant lack of documentation, discussion forums and in general people using it. To overcome this, the clarity of fundamentals of types software architectures and the core field are important from the user's side.
- 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.

The discussions in the above sections show the considerable 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 is and always will be a multi-disciplinary task where no linear approach exists. Hybrid systems like VTOL aircraft when equipped with suitable sensory payloads can boost target acquisition capabilities, can use AI to take informed decisions and even show collaborative behavior with homogeneous swarming agents to act as force multipliers on the front-line.

Acknowledgements

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

References

- [1] Haowei Gu, Ximin Lyu, Zexiang Li, Shaojie Shen, and Fu Zhang. 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)*, pages 160–169. IEEE, 2017.
- [2] Roman Czyba, Marcin Lemanowicz, Zbigniew Gorol, and Tomasz Kudala. Construction prototyping, flight dynamics

- modeling, and aerodynamic analysis of hybrid vtol unmanned aircraft. *Journal of Advanced Transportation*, 2018, 2018.
- [3] Özgür Dündar, Mesut Bilici, and Tarık Ünler. Design and performance analyses of a fixed wing battery vtol uav. *Engineering Science and Technology, an International Journal*, 23(5):1182–1193, 2020.
 - [4] Ghozali Suhariyanto Hadi, Muhammad Ramadhan Kusnaedi, Puspita Dewi, Aris Budiarto, and Agus Budiyo. Design of avionics system and control scenario of small hybrid vertical take-off and landing (vtol) uav. *Journal of Instrumentation, Automation and Systems*, 2(2):67–71, 2015.
 - [5] Emilio M Botero and Juan J Alonso. Conceptual design and optimization of small transitioning uavs using suave. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 4149, 2017.
 - [6] Ugur Ozdemir, Yucel Orkut Aktas, Aslihan Vuruskan, Yasin Dereli, Ahmed Farabi Tarhan, Karaca Demirbag, Ahmet Erdem, Ganime Duygu Kalaycioglu, Ibrahim Ozkol, and Gokhan Inalhan. Design of a commercial hybrid vtol uav system. *Journal of Intelligent & Robotic Systems*, 74(1):371–393, 2014.
 - [7] Matthew Clarke, Jordan Smart, Emilio M Botero, Walter Maier, and Juan J Alonso. Strategies for posing a well-defined problem for urban air mobility vehicles. In *AIAA Scitech 2019 Forum*, page 0818, 2019.
 - [8] Daniel Böhnke, Arthur Rizzi, Mengmeng Zhang, and Bjoern Nagel. 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*, page 222, 2013.
 - [9] Hrvoje Jasak. Openfoam: open source cfd in research and industry. *International Journal of Naval Architecture and Ocean Engineering*, 1(2):89–94, 2009.
 - [10] Ping He, Charles A Mader, Joaquim RRA Martins, and Kevin J Maki. Dafoam: An open-source adjoint framework for multidisciplinary design optimization with openfoam. *AIAA journal*, 58(3):1304–1319, 2020.
 - [11] Luning Sun, Han Gao, Shaowu Pan, and Jian-Xun Wang. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361:112732, 2020.
 - [12] Marine M Segui, Georges Ghazi, Ruxandra M Botez, and Emily M Thompson. Design, development and validation of a cessna citation x aerodynamic model using openvsp software. In *2018 Modeling and Simulation Technologies Conference*, page 3256, 2018.
 - [13] Sebastian Herbst and Adrian Staudenmaier. Using openvsp in a conceptual aircraft design environment in matlab. In *Congress of the International Council of the Aeronautical Sciences*, 2016.
 - [14] Mohammad H Sadraey. *Aircraft design: A systems engineering approach*. John Wiley & Sons, 2012.
 - [15] Maxim Tyan, Nhu Van Nguyen, Sangho Kim, and Jae-Woo Lee. Comprehensive preliminary sizing/resizing method for a fixed wing-vtol electric uav. *Aerospace Science and Technology*, 71:30–41, 2017.
 - [16] A Ramirez Serrano. Design methodology for hybrid (vtol+fixed wing) unmanned aerial vehicles. *Aeronautics and Aerospace Open Access Journal*, 2(3):165–176, 2018.
 - [17] Daniel P Raymer. *Aircraft design: a conceptual approach* (aiaa education series). Reston, Virginia, 2012.
 - [18] Rishabh Dagur, Vikrant Singh, Shabir Grover, Nikhil Sethi, and BB Arora. Design of flying wing uav and effect of winglets on its performance. *Int J Emerg Technol Adv Eng*, 8(3), 2018.
 - [19] Lester James Miranda. Pyswarms: a research toolkit for particle swarm optimization in python. *Journal of Open Source Software*, 3(21):433, 2018.
 - [20] Huseyin Sahin and Tugrul Oktay. Powerplant system design for unmanned tricopter. *The Eurasia Proceedings of Science Technology Engineering and Mathematics*, pages 9–21, 2017.
 - [21] Nikhil Sethi, Prabhash Chauhan, Shashwat Bansal, and Ranganath M Singari. Robust vehicle development for student competitions using fiber-reinforced composites. In *Advances in Manufacturing and Industrial Engineering*, pages 61–76. Springer, 2021.
 - [22] Anuj Agrawal, Aniket Gupta, Joyraj Bhowmick, Anurag Singh, and Raghava Nallanthighal. A novel controller of multi-agent system navigation and obstacle avoidance. *Procedia Computer Science*, 171:1221–1230, 2020.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [supplementary.pdf](#)