OpenUAS: An Open-Source Unmanned Aircraft Systems (UAS) Testbed Solution under Cost Constraints

Varad V. Kulkarni,* Allison L. Howard,† Sydney R. V. Turner,‡ Mukul S. Kulkarni,§ Nisha Raj,¶ Eric A. Rasmussen,‖ Mehmet B. Sefer,** Karanvir Singh,†† and Kristin Y. Rozier‡‡
{varadk, alhoward, sydtur9, mukulk, nisharaj, eric27, sefer, skaran28, kyrozier}@iastate.edu
Iowa State University, Ames, Iowa, 50011

Today, Unmanned aircraft systems (UAS) are taking on increasingly complex autonomous missions with various payloads. This paper presents two complete UAS testbed solutions that are fixed-wing, electric, fully open-source (for both hardware and software), and comprised of Commercial Off-the-Shelf (COTS) parts. The testbed is affordable, provides easy manufacturability, and is highly reconfigurable to meet various customer needs. The paper offers quantitative and qualitative data from the last four years that contribute to the successes and learning opportunities for designing, building, and flying an open-sourced UAS. We present aerodynamic comparisons of CFD data using StarCCM+, stability analysis, power system analysis, and a workflow to integrate NASA cFS with PX-4 Autopilot using the NASA ICAROUS Framework. Flight test results and the effects of in-flight controller gain tuning are incorporated to validate the design and manufacture of both UAS testbeds. A complete documentation webpage highlights the contributions from this paper.

I. Nomenclature

α	=	angle of attack
C_d	=	two dimensional drag coefficient
C_l	=	two dimensional lift coefficient
c	=	chord
CAD	=	computer aided design
CFD	=	computational fluid dynamics
ϕ	=	roll angle
PLA	=	polylactic acid
ψ	=	yaw angle
ρ	=	density
θ	=	pitch angle
UAS	=	unmanned air system
V_∞	=	cruise velocity

II. Introduction

There are many interesting applications for small, electric fixed-wing UAS. Some notable current applications are Zipline's[1] fixed-wing UAS that are actively being used to perform fully autonomous delivery. Other military drones such as AeroVironment's Puma[2] are specialized for GPS-denied navigation. Academic research in such

*Undergraduate Student, Department of Aerospace Engineering, AIAA Student Member (1603088)
†Undergraduate Student, Department of Aerospace Engineering, AIAA Student Member (1328838)
‡Undergraduate Student, Department of Aerospace Engineering, AIAA Student Member (1538143)
§Undergraduate Student, Department of Mechanical Engineering, AIAA Student Member (1603137)
¶Undergraduate Student, Department of Electrical and Computer Engineering, AIAA Student Member (1603252)
‖Undergraduate Student, Department of Aerospace Engineering, AIAA Student Member (1423404)
**Undergraduate Student, Department of Aerospace Engineering, AIAA Student Member (1602828)
††Undergraduate Student, Department of Aerospace Engineering, AIAA Student Member (1424324)
‡‡Muilenburg Associate Professor, Department of Aerospace Engineering, AIAA Associate Fellow

applications calls for an appropriate testbed that can support a wide variety of missions and is easy to configure. The E-flite Apprentice is an affordable testbed, but does not feature the appropriate room needed for additional sensors or payloads [3]. The Applied Aeronautics Albatross[4] supports various payloads and features 5-hour battery life, 68 kilometer per hour cruise speed, and closely aligns with the testbed that we are suggesting. In essence, the testbed should have support for multiple kinds of sensors and should have easily replaceable parts, while also being affordable.

The same argument applies for autopilot software support for fixed-wing UAS. There are many open-source solutions available, such as PX-4, ArduPilot[5], Paparazzi[6], and ROSPlane[7]. Out of these, PX-4 has a high availability of documentation, but only has companion computer support for ROS2[8] that primarily provides quadrotor functionalities. ROSPlane bridges this very gap but lacks the plug-and-play capability of PX-4.

We contribute two new versions of the first truly open-source (in both hardware and software) design for a fixed-wing UAS [3]. Our new version includes appropriate SolidWorks CAD models, specifications for every part, manufacturing instructions, and a full software setup to facilitate intelligent capabilities. We also present our flight test data on the maneuverability and autonomous features of our testbeds.

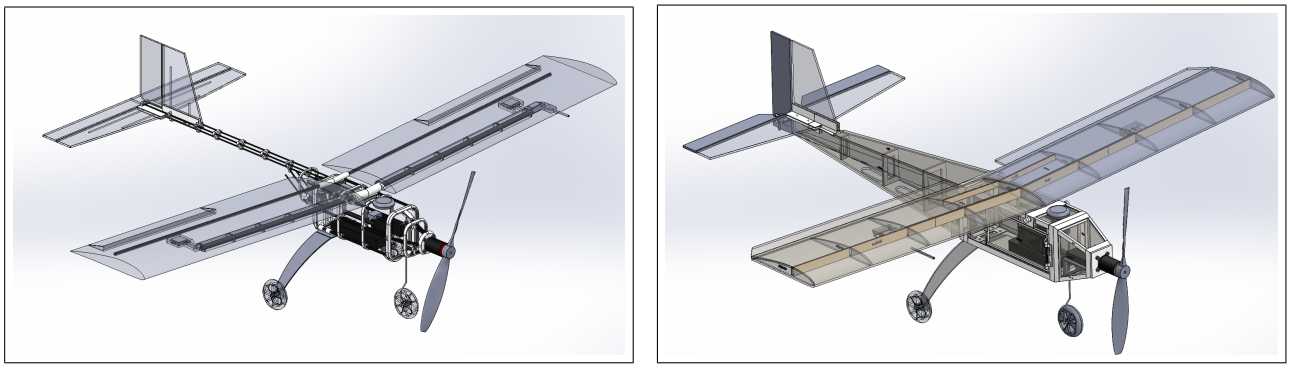
This paper is organized as follows. Section III overviews the OpenUAS design, including CAD modeling, CFD analysis, and materials considerations. We detail the manufacturing process in Section IV, including the latest iteration budget, materials choices, manufacturing process, and lessons learned. To automate the OpenUAS, Section V lays out the electronics and software setup, including details of the electronic hardware, motors, propellers, and batteries. We include a discussion of our current autopilot setup and our ongoing software integration effort of PX-4 and NASA cFS[9]. Accounts of our flight testing and evaluation appear in Section VI, including flight procedures, and our results from PID controller tuning and lidar based autoland testing. Section VII discusses impacts and designs for future work.

III. Design

A. OpenUAS Version 2.0 and 3.0 Characteristics

The objectives for both OpenUAS 2.0 and 3.0 are as follows: the design is easily replicable, utilizes a minimal amount of parts, and follows traditional aircraft structure. The OpenUAS 2.0 SolidWorks CAD design shown in Figure 1a has an endoskeleton fuselage made entirely of 3D-printed pieces and carbon fiber tubes.

The OpenUAS 3.0 SolidWorks CAD design shown in Figure 1b is strongly influenced by the team's goal of making the UAS more accessible to potential customers. The various design changes include the differences in materials used for construction, and optimizing the tail surfaces.



(a) OpenUAS 2.0 **(b) OpenUAS 3.0**

Figure 1. SolidWorks 3D CAD Model Isometric Views

Both OpenUAS 2.0 and 3.0 feature a single nose-mounted electric motor for propulsion, a high-mounted wing, a conventional tail, and a tricycle landing gear configuration. For a comparison of the sizing and weight of each UAS, see Table 1.

Table 1 Aircraft Major Dimensions and Weights

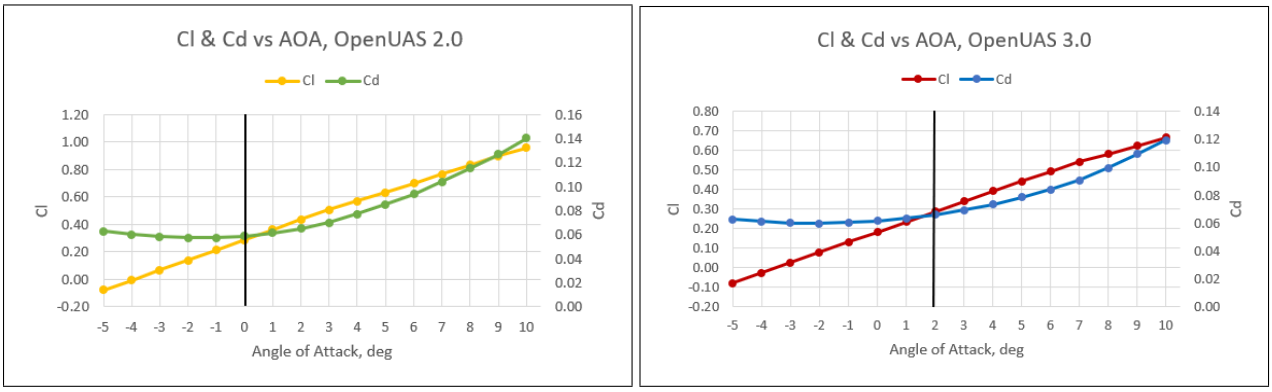
Dimension	OpenUAS 2.0	OpenUAS 3.0
Aircraft Length (in)	44.25	47.75
Aircraft Height (in)	17.25	16.5
Wingspan (in)	64	56
Chord Length (in)	10	10.25
Payload/Electronics Volume (in^3)	250	400

B. CFD Data

One of the ways to optimize the design is to perform CFD analysis on the 3D CAD models, using StarCCM+[10]. Performing a stability analysis, will display criterion for steady level flight. Cruise velocity, V_∞ and altitude is set to 40 miles per hour, and 1500 feet, respectively for both cases. The steady level flight angle of attack, α is 0 degrees for OpenUAS 2.0 and 2 degrees for OpenUAS 3.0, indicated with a black line on Figure 2. Considering OpenUAS 2.0, with an overall weight of 4 lbs and 7 oz, we conclude the maximum coefficient of lift is 0.9562, and the maximum coefficient of drag is 0.14. A design goal for OpenUAS 3.0 is to reduce overall drag on the aircraft, which is accomplished by placing the center of gravity at the quarter wing chord. The overall weight is 4 lbs and 10 oz, which is considerably light for a UAS. The maximum coefficients of lift and drag for OpenUAS 3.0 are 0.6667 and 0.1196, respectively. Relations for coefficients of lift and drag with the given parameters are found in Equations 1 and 2 as given in [11].

$$C_l = \frac{L}{\frac{1}{2}\rho V_\infty^2 c} \tag{1}$$

$$C_d = \frac{D}{\frac{1}{2}\rho V_\infty^2 c} \tag{2}$$



(a) **(b)**

Figure 2. CFD Data for OpenUAS a) 2.0 and b) 3.0

IV. Manufacturing

A. Manufacturing Process of 3.0

The manufacturing process for OpenUAS 3.0 starts with creating a Bill of Materials (BOM) seen in Table 2, that outlines the weight and price of materials optimal for the design. The overall weight of the manufacturing materials is 1.7 lbs, excluding the weight of the electronics.

Table 2 OpenUAS 3.0 Manufacturing Bill of Materials

Components	Materials	Weight, oz	Price
Nose Cone, Fuselage, Wing, Empennage, Tail	22" x 28" Poster Board, Balsa Wood	8.15	$37.20
Front Landing Gear	DuBro Super Strength Landing Gear Size 0.35-0.50 paired with Dubro 3" Super Lite Wheels	5.34	$34.68
Rear Landing Gear	DuBro Plane Tailwheel Bracket Size 0.40 paired with DuBro 3/4" Tailwheel	1.81	$7.82
Connectors	Polymaker PolyLite PLA Filament	0.45	$10.57
Total:		**15.75**	**$90.27**

Figure 3 outlines the process for completing the build of the UAS'. Each CAD part that is designed to be made out of 3D printing is exported as STL files and closely examined to ensure manufacturability quality. The parts are then imported into Adobe Illustrator to set line thickness and colors for the laser cutting process. The build process can be modified to the customer's liking, depending on the materials used and assembly of parts.

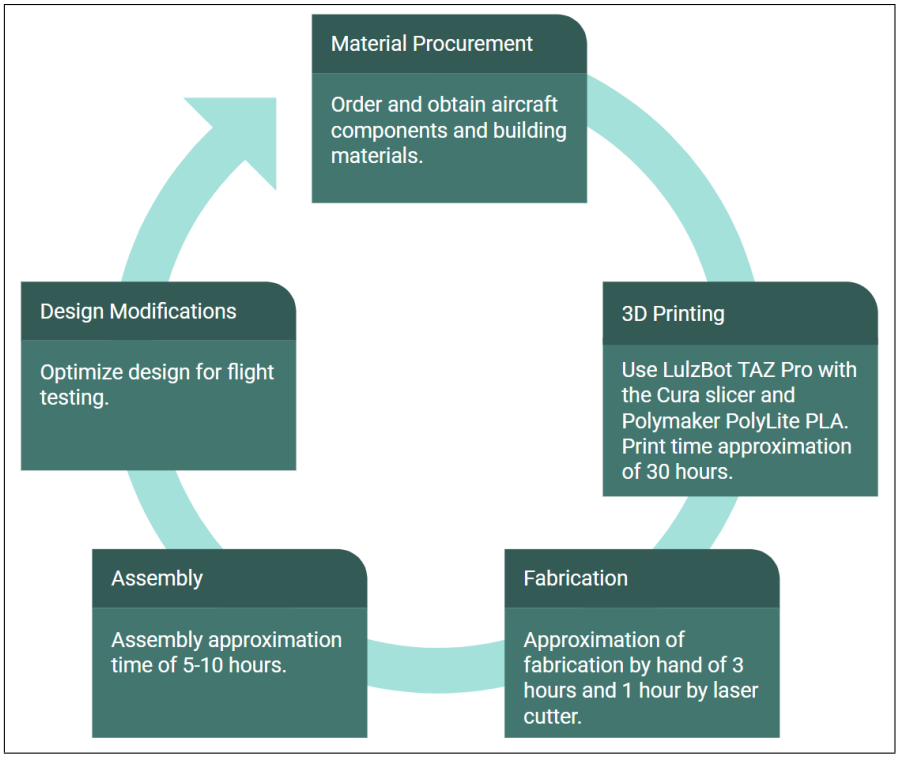


Figure 3. Manufacturing Process Overview

B. Completed Testbed

OpenUAS 2.0 features monokote added to the design, wrapping on the outside of the fuselage, to control airflow within the body, seen in Figure 4a. The fuselage utilizes a clamshell design to access internal electronics, when the wing is not mounted to the aircraft. Solid light foam makes up the main wing with two carbon fiber spars running horizontally and attaching to the larger 3D printed tubes sitting on top of the fuselage. The wing can be removed into two sections to access the fuselage clamshell door. Lastly, there is a carbon fiber boom that runs from the fuselage to the tail surfaces, which are made from poster board.

The majority of parts for OpenUAS 3.0 are made from inexpensive poster board and 3D printed PLA. OpenUAS 3.0 does not use any carbon fiber, monokote, or other materials that may be more difficult for a customer to purchase themselves, outside of the electronic components. The fuselage is constructed out of poster board and 3D-printed joints

to attach different sections, see Figure 4b. It also has a door on each side to access the electronics and payload. The wing has internal balsa wood spars, poster board ribs, and a thick skin also made of poster board. It is attached to the body via rubber bands and tie-down points located on the fuselage. The tail is constructed out of poster board and is connected by a 3D-printed piece to the empennage, which is made entirely from poster board.

(a) **(b)**

Figure 4. a) 2.0 Model and b) 3.0 Model

C. Manufacturing Lessons Learned

The process of completing the ribs within the wing for OpenUAS 3.0 poses challenges for creating an ideal airfoil shape with a round leading edge. The material being 0.1875 inches thick requires precision cutting. Exacto knives prove to cut effectively, for the ability to utilize readily available tools.

In OpenUAS 2.0, MonoKote film is the primary material for wrapping the bodies, such as the fuselage and wing. While the MonoKote wrapping does not significantly impact the surface quality of the wing, considering its rigid CNC-manufactured XPS foam body, it impacts the surface quality of the fuselage. The MonoKote is not the most even surface, but it allows for the internal structures to be consolidated.

OpenUAS 3.0 considers the challenges with MonoKote and instead uses poster board in both the fuselage and wing structures. However, the use of poster board poses two main challenges: first being precise cutting of the poster board to specific part dimensions and second being waste of poster board material. Parts are individually drawn and cut without optimizing their placement on the poster board. To address these challenges, the team is using a 2D bin packing algorithm, libnest2D library. Using this library, individual parts are packed onto the poster board as efficiently as possible. The team marks the poster boards with the coordinates of the individual parts' corner points and cuts them with a knife.

V. Electronics and Software

The OpenUAS electronics and software setup is designed with the same principles of accessibility and reconfigurability in mind. The setup is made completely from Commercial-off-the-shelf (COTS) electronics parts that are affordable and easily available online. For software, we currently chose the PX-4 Autopilot flight stack due to its fully open source codebase, support for intelligent capabilities, and user-friendly documentation. In the upcoming subsections, we will discuss our current electronics setup and our ongoing research into other advanced software functionalities.

A. Motor Selection

We present comparisons of multiple motors paired with the OpenUAS 2.0 and 3.0 airframes in Figure 5. The plot has been generated using MotoCalc 8[12]. OpenUAS 2.0 utilizes a 690 kilovolts motor and displays optimal flight conditions at a cruise velocity of 31 miles per hour. OpenUAS 3.0 utilizes a 900 kilovolts motor displays optimal flight conditions at a cruise velocity of 38 miles per hour.

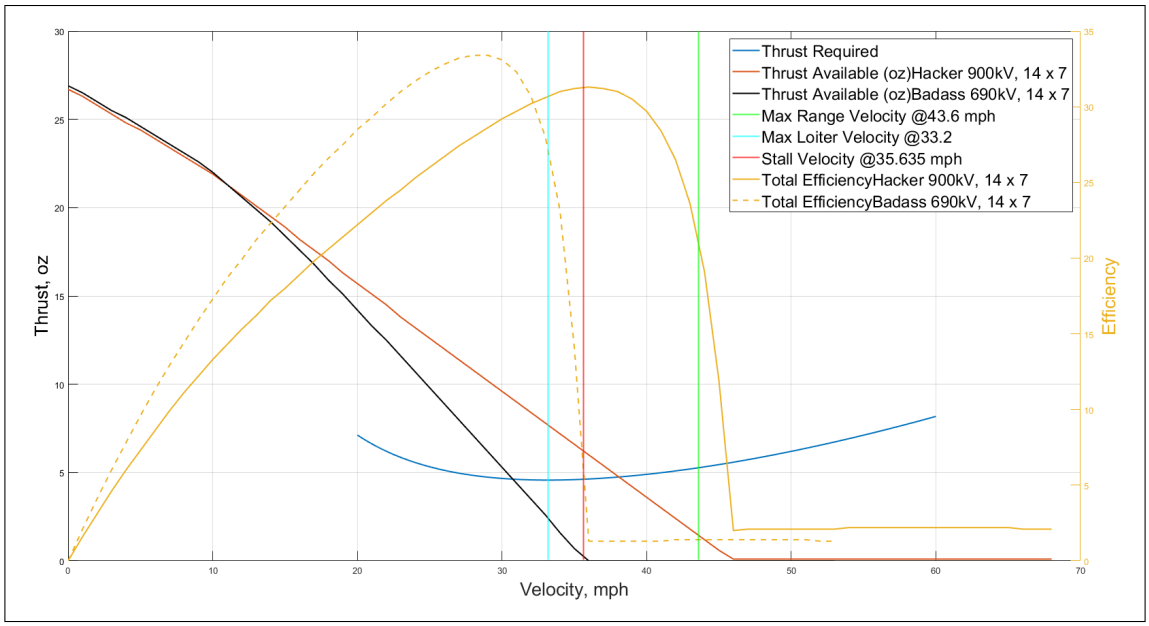


Figure 5. Thrust and Velocity Comparisons for the 690kV and 900kV motor at 100% throttle

The cruise velocity condition of 40 miles per hour, requires 5 ounces of thrust, and unfortunately, neither motor provides enough thrust to produce this speed, according to the MotoCalc data. However, it is possible to fly the given aircraft at the mentioned cruise speed at much lesser than 100% throttle in real conditions.

B. Electronics budget

We list the currently supported electrical components in both testbeds in Table 3. Some components come in bundles and are denoted by a * or ** to indicate that they can be purchased for less money in a bundle together. These components are the bare minimum to implement an autopilot into OpenUAS 2.0 and 3.0 and exclude options such as airspeed sensors and lidars, which we still support.

Table 3 Iron Bird OpenUAS Components List

Component	**Model**	**Price**
Flight Computer	Holybro Pixhawk 4 Autopilot	$170 (SKU20068)*
GPS	Holybro Pixhawk 4 Neo-M8N GPS	$170 (SKU20068)*
Power Module	Holybro APM Power Module 12S - PM02 V3	$170 (SKU20068)*
RC Transceiver	FrSky X8R 8/16Ch S.Bus ACCST Telemetry Receiver W/Smart Port	$244.99 (SKU2153)**
RC controller	FrSky Taranis X9D plus	$244.99 (SKU2153)**
Telemetry Radios	Holybro 433Mhz 915Mhz Transceiver Radio Telemetry Set	$39.00
ESC	BadAss Renegade 85A ESC	$79.99
Motor	BadAss 2826-690Kv brushless motor	$69.99
Propeller	APC 14x12 E	$9.05
Battery	Ovonic 5000mAh 11.1V 3S 50C 3 Cell LiPo	$32.99
	TOTAL:	**$613.02**

C. Software and NASA cFS Integration

Our current flight stack of choice is PX-4 Autopilot due to its open-source software, readable documentation, and support for various sensor suites and autonomous capabilities. We are currently investigating companion computer support for advanced capabilities.

Our focus is to utilize NASA cFS for the fixed wing UAS', using the open-source NASA ICAROUS[13] framework, which can also be found on GitHub. This will provide a layered architecture implementation of autonomous flight software and intelligent features such as runtime verification[14]. The framework interfaces with PX-4 autopilot through function calls in the apInterface module. Figure 6 references the function names and outlines the data read from the autopilot and constructs software bus messages or data read from the software bus and sends it to the autopilot. The current procedure being developed is integrating the apInterface modules from ICAROUS into the NASA cFS software. The ICAROUS framework and cFS frameworks have a similar makeup of applications, making this an ideal integration. We are experimenting with including the apInterface modules into the cFS apps modules. However, in our initial test we experience insufficiently run apInterface modules and display within the cFS user interface. Based on the results we hypothesize that we are missing multiple header files that need to be included. Our current approach is to use ICAROUS simultaneously with its own ground station (WebGS), to be able to monitor the UI.

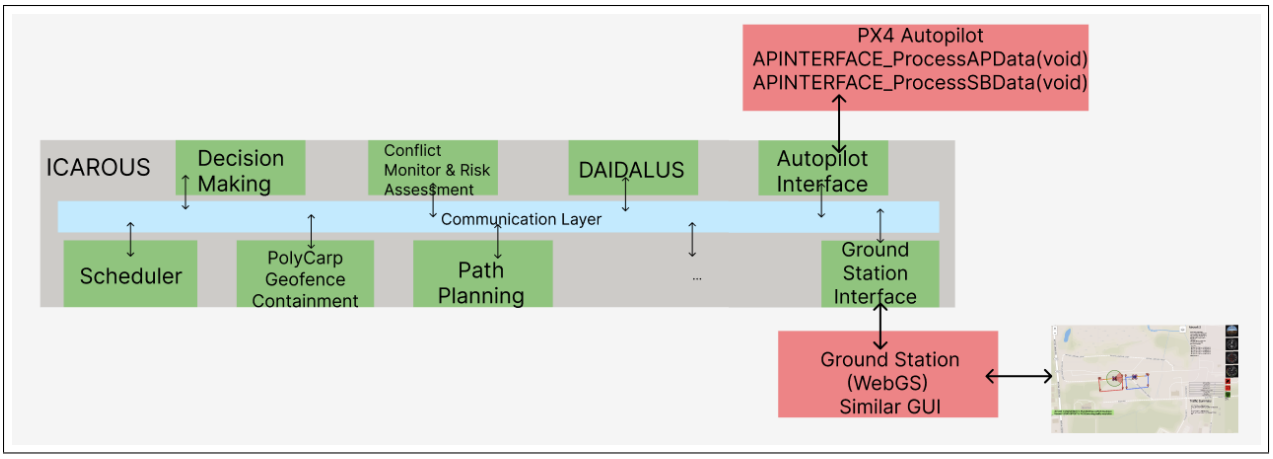


Figure 6. ICAROUS architecture with PX-4 interface functions and WebGS GUI

VI. Flight Test

A. 3.0 Flight Characteristics

The aircraft's altitude is analyzed during the test flight using two different flight styles. One style is based on maintaining a stable flight with smooth commands and, therefore, smooth responses. The takeoff conditions for pitch θ, yaw ψ, and roll ϕ, consider the aircraft size and fixed wing nature. Those make smoother takeoffs without abrupt variations of position or attitude in any direction. Gaining altitude and maintaining a steady level flight is an easy task for this model, presenting a good climb rate and handle during constant high-velocity winds and wind gusts. When steady level flight is achieved, the plane's attitude is stable. Pitch, yaw, and roll controls have very slow and smooth movements. This removes maneuverability from the plane, making it more stable while presenting resistance to abrupt changes of direction. However, a smoother flight style is advantageous due to its capability to maintain a steady flight while also having plenty of resistance to wind gusts and sudden turbulence. While under a landing regime, all control surfaces demonstrate to be capable of the task with the necessary attitude to maintain the flight stable while also being able to respond to wind changes, turbulence, or other factors that may cause abrupt heading changes and damage the aircraft in this critical point of the flight.

The second style is a more aggressive flight style where fast response is the focus over smoothness. Starting with takeoff, the aircraft is able to respond quickly to the pitch input to rotate when the aircraft achieves the ideal takeoff velocity. During the climb, the pitch response still is quick, but the roll is not as quick to respond as one would like. Similarly, during cruise, the aircraft handles well with the aggressive inputs for pitch and yaw but sometimes struggles with the inputs of roll, especially during the downwind to upwind turn. This lack of roll control could be due to the

high-velocity winds and wind gusts or the smaller surface area of the ailerons. Like takeoff, the aircraft handles well during descent and landing. The pitch and yaw controls are responsive, but the roll is not as responsive.

B. PID Controller tuning

We present our PID tuning data and our ongoing process of tuning the OpenUAS 3.0 airframe to provide a more maneuverable testbed. We will tune the pitch controller since it will be intuitive from a flight test perspective and will not have a coupled response as is the case for roll and yaw motion.

We start the process by investigating PX-4's default controller behavior in Simulation-In-Hardware (SIH) simulation by connecting the Pixhawk 4 to a computer running the QGroundControl[15] ground station via a serial connection. We present the results in a plot using the PID-Analyzer tool provided by PX-4 Flight Review as shown in 7. The tool calculates the average step response from multiple instances and shows pitch signal strength with respect to a step input as a function of time. This serves as a benchmark on how the fixed-wing pitch controller will respond in an environment with no external disturbances. We observe that this plot still deviates from a theoretical response where there should be negligible steady-state error.

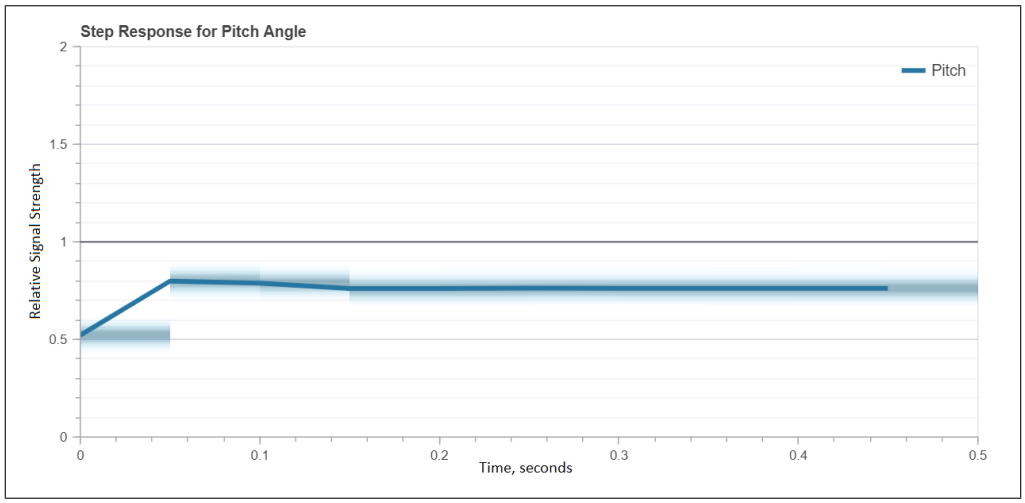
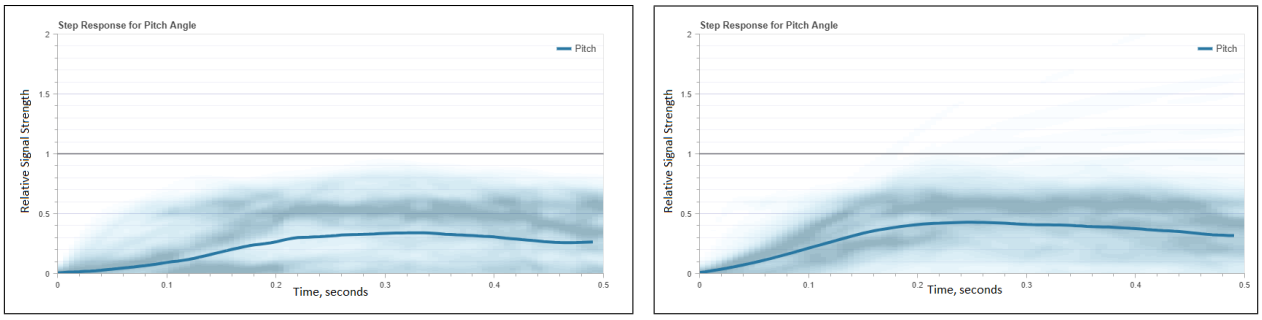


Figure 7. Benchmark pitch response from SIH Simulation

The preliminary flight test is helpful to observe the benchmark conditions. Performing a similar analysis on OpenUAS 3.0 allows for the observation of each control response. By focusing on the pitch response, we conclude that it needs additional tuning since it did not reach a given pitch angle setpoint within the tool's default timeframe of 0.5 seconds. We conclude that it would be appropriate to first tune the Proportional as well as Integral pitch gains to provide a quicker pitch response and to reduce steady state error. By doubling the default pitch gain from 0.08 to 0.16, we observe a faster peak time of 0.2 seconds as opposed to approximately 0.35 seconds, seen in Figure 8. As the proportional gain increases, the oscillations dampen in the pitch rate response. The process of tuning the Integral gain involves looking at pitch rate command, and adjusting the value to allow for a faster response time with the pitch rate response, without leading to more oscillations. We plan to continue this process and tune the Integral gain.



(a) Proportional gain set to 0.08 **(b) Proportional gain set to 0.16**

Figure 8. Pitch response during flight test

C. Autonomy

Using the capabilities of the Pixhawk 4 flight controller paired with a GPS module, airspeed sensor, and lidar sensor, we successfully implement fully autonomous take-off, loiter, and landing. GPS waypoints are configured prior to take-off via the QGroundControl software. QGroundControl is a comprehensive flight planning and control application that communicates with drones via the MAVLink protocol. The lidar sensor enables smooth landings, however, sporadic signal drop outs with the lightware SF11/C sensor are possible. The lidar sensor reports an error code indicating out-of-range/signal lost when the aircraft's altitude is well within the 100 meter maximum range of the sensor. When the lidar sensor reports an error during landing, PX4 will abort and return to loiter for either another landing attempt or return of manual control. We are currently attempting to debug the cause of failure. We present a notable plot from the SF11/C lidar sensor in Figure 9. This plot depicts a successful auto-landing and shows some instances of sensor failure marked by the sudden increase in distance value to 130 meters.

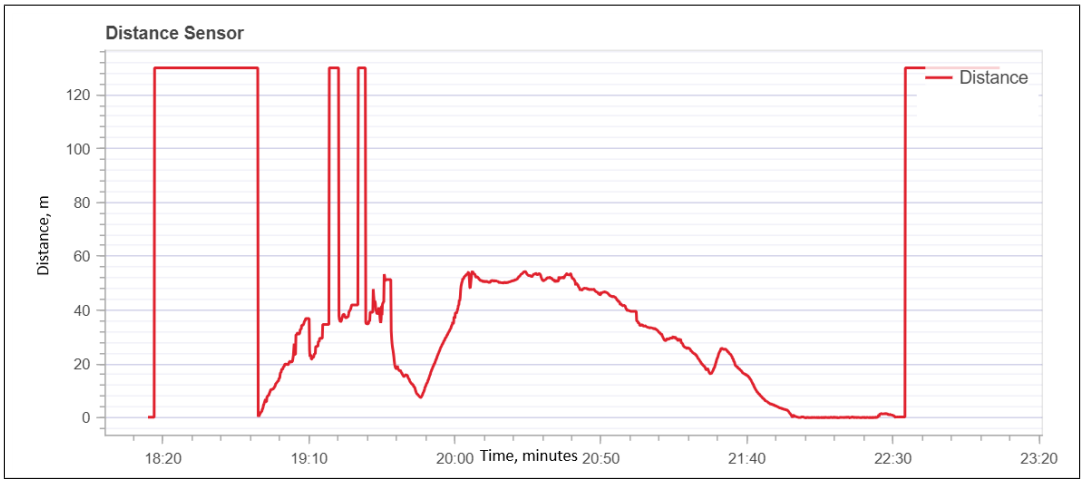


Figure 9. SF11/C lidar altitude readings (130m indicates out-of-range distance/signal lost error)

VII. Conclusion

Future work advancing from OpenUAS 3.0 to OpenUAS 4.0 includes using a thinner poster board for the main wing, designing a removable tail configuration, and incorporating an adaptable version of apInterface with cFS and ICAROUS software.

We will be updating the wing design to incorporate a thinner poster board for the desired airfoil shape. This adjustment aims to preserve the advantages of the thicker material, such as providing more lift, while mitigating the manufacturing challenges. Additionally, this shift to a thinner poster board will contribute to a reduction in overall

weight and cost. This targeted approach emphasizes tailoring material choices to specific components within the manufacturing process to optimize overall performance.

The team uses a laser cutter to achieve straight edges in the poster board to optimize part quality and time consumption. The team concludes that utilizing the 2D bin packing algorithm libnest2D along with the laser cutter significantly optimizes the manufacturing process. With the tail design, we want to create a larger control surface area to improve the flight characteristics. We will be providing a solution by allowing for a detachable conventional tail and v-tail design.

Our next step is to continue analyzing the cFS and ICAROUS software in order to find key differences that may need to be included in order for apInterface to correctly run within the cFS framework. This integration will allow the cFS framework to interface with PX-4 autopilot through built-in function calls. It will also allow for ground station communication with WebGS that can provide flight controls and vehicle setup. Figure 6 depicts a top level view of the ICAROUS framework and the modules it provides. The team hopes to use these modules to integrate cFS to interface with PX-4 autopilot. The Hardware-In-the-Loop (HITL) simulation is the appropriate next step of integration with Simulink and PX-4 in order to get a better theoretical dynamic response. Currently, there are no available airframe configurations for the fixed-wing aircraft.

Furthermore, we plan to continue flight testing our designs and improving the pitch controller response, and further implementing a more robust autonomous flight. Given the conclusions of the lidar sensor, we plan to experiment with other distance sensors with the goal of finding a more reliable sensor for this application.

VIII. Appendix

All contributions presented here can be found on the OpenUAS webpage: `https://open-uas.github.io/`.

Acknowledgments

This work was supported by NSF CAREER Award CNS-1552934 and NASA Award ISGC 80NSSC20M0107 (Iowa Space Grant Consortium).

We would also like to thank Isaac Arp (icarp@iastate.edu), Declan Green (dcgreen@iastate.edu), Dana Love (dalove@iastate.edu), Luke Marzen (ljmarzen@iastate.edu), Artur Nunes (abnunes@iastate.edu), and Andrew Williams (andreww2@iastate.edu) for their contributions to this project.

References

[1] Lamptey, E., and Serwaa, D., "The Use of Zipline Drones Technology for COVID-19 SamplesTransportation in Ghana," *HighTech and InnovationJournal*, 2020.

[2] AeroVironment, "Puma 3 AE," Online: `https://www.avinc.com/uas/puma-ae`, 2024.

[3] Johannsen, C., Anderson, M., Burken, W., Diersen, E., Edgren, J., Glick, C., Jou, S., Kumar, A., Levandowski, J., Moyer, E., Roquet, T., VandeLoo, A., and Rozier, K. Y., *OpenUAS Version 1.0*, IEEE, Athens, Greece (Virtual), 2021.

[4] Applied Aeronautics, "Albatross," Online: `https://www.appliedaeronautics.com/albatross-uav`, 2014.

[5] Dronecode Project, Inc., "PX4 User Guide," Online:`https://docs.px4.io/master/en/`, 2020.

[6] Gati, B., "Open source autopilot for academic research - The Paparazzi system," *2013 American Control Conference*, 2013, pp. 1478–1481. https://doi.org/10.1109/ACC.2013.6580045.

[7] Ellingson, G., and McLain, T., "ROSplane: Fixed-wing autopilot for education and research," *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 1503–1507. https://doi.org/10.1109/ICUAS.2017.7991397.

[8] Macenski, S., Foote, T., Gerkey, B., Lalancette, C., and Woodall, W., "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, Vol. 7, No. 66, 2022, p. eabm6074. https://doi.org/10.1126/scirobotics.abm6074, URL https://www.science.org/doi/abs/10.1126/scirobotics.abm6074.

[9] NASA, "core Flight System (cFS)," Online:https://cfs.gsfc.nasa.gov/, accessed 2024.

[10] Siemens Digital Industries Software, "Star CCM+," Online:https://www.plm.automation.siemens.com/global/en/products/simcenter/STAR-CCM.html, October 2020.

[11] Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls*, Design Analysis and Research Corporation, Lawrence, KS, 2003.

[12] Capable Computing, Inc., "MotoCalc," Online:http://www.motocalc.com/, 2021.

[13] Consiglio, M., Muñoz, C., Hagen, G., Narkawicz, A., and Balachandran, S., "ICAROUS: Integrated configurable algorithms for reliable operations of unmanned systems," *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1–5. https://doi.org/10.1109/DASC.2016.7778033.

[14] Schumann, J., Moosbrugger, P., and Rozier, K. Y., "R2U2: Monitoring and Diagnosis of Security Threats for Unmanned Aerial Systems," *Conference on Runtime Verification (RV15)*, Springer-Verlag, Vienna, Austria, 2015.

[15] Dronecode Project, Inc., "QGroundControl," , accessed 2024. URL http://qgroundcontrol.com/.