Aircraft Geometry and Propulsion Architecture Visualization for the Future Aircraft Sizing Tool (FAST)

Nawa Khailany*, Paul R. Mokotoff†, and Gokcin Cinar‡
Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, 48109

The Future Aircraft Sizing Tool (FAST) is a Matlab-based, open-source software developed by the University of Michigan for early-phase conceptual aircraft design. FAST facilitates the design and analysis of both conventional and advanced aircraft configurations with novel propulsion systems, enabling preliminary sizing and performance evaluation based on specific requirements, desired technology targets, and system-level objectives. It has been employed in NASA's Electrified Aircraft Propulsion and Electrified Powertrain Flight Demonstration projects to assess novel aircraft concepts, including an electrified commercial freighter (notional Lockheed Martin LM-100J) and NASA's Subsonic Single Aft Engine configuration. This paper presents the development of a visualization package within FAST, addressing the need for visual representations of aircraft designs throughout the sizing process. The integrated package provides visualizations of the aircraft's outer mold line and schematic representations of its propulsion architecture. Users can create custom aircraft geometries or utilize preset ones available in FAST. Additionally, the visualization package dynamically updates the aircraft's shape and size as the aircraft sizing process progresses, enhancing FAST by enabling designers to effectively visualize and refine their aircraft concepts during early design stages.

I. Nomenclature

ADEBO	=	Aircraft DEsign BOx
BWB	=	Blended Wing Body
EAP	=	Electrified Aircraft Propulsion
EPFD	=	Electrified Powertrain Flight Demonstration
ESP	=	Engineering Sketch Pad
eVTOL	=	Electric Vertical Take-Off and Landing vehicle
FAST	=	Future Aircraft Sizing Tool
ISSAAC	=	Integrated Subsystems Sizing and Architecture Assessment Capability
LUCAS	=	Library for Unified Conceptual Aircraft Synthesis
OML	=	Outer Mold Line
PEACE	=	Parametric Energy-based Aircraft Configuration Evaluator
SUAVE	=	Stanford University Aerospace Vehicle Environment
SUSAN	=	SUbsonic Single Aft eNgine
TLAR	=	Top Level Aircraft Requirement

II. Introduction

The Future Aircraft Sizing Tool (FAST) [1] is a Matlab-based, open-source aircraft sizing software developed for early-phase conceptual design space exploration and tradeoff studies. It enables the design and analysis of both conventional and advanced aircraft with novel propulsion systems based on a set of top level aircraft requirements (TLARs). FAST achieves this by leveraging historical data regressions from over 400 previously flown aircraft and incorporating first-order physics models. This capability is particularly useful for estimating the impact of emerging technologies, including electrification, on the aircraft design at an early stage in the design process.

*Research Assistant, Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, 48109, AIAA Student Member

†Graduate Research Assistant, Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, 48109, AIAA Student Member

‡Assistant Professor, Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, 48109, AIAA Senior Member

FAST has been employed in NASA's Electrified Aircraft Propulsion (EAP) [2] and Electrified Powertrain Flight Demonstration (EPFD) [3] projects to explore the impacts of new technologies and operational modes on electrified aircraft design and operation. These projects evaluate configurations such as a notional electrified LM-100J [1] with different operational modes and battery technologies, and the SUbsonic Single Aft eNgine (SUSAN) [4, 5] concept with state-of-the-art technologies such as boundary layer ingestion and natural laminar flow. A key objective for NASA's EPFD team was to understand how modifications to aircraft design parameters influenced both the performance and physical characteristics (e.g., shape and size) of the configurations. To facilitate this analysis, a visualization tool was required to accurately depict the aircraft's outer mold line (OML) and provide schematic representations of the onboard propulsion architecture. The core features identified for the visualization feature included:

1) Visualizing an aircraft's OML outside of an aircraft sizing loop,
2) Plotting a schematic of the aircraft's propulsion architecture,
3) Dynamically updating the aircraft's OML during the aircraft sizing process,
4) Allowing the code to be open-source

Before developing a new visualization tool, existing aircraft sizing and geometry creation/manipulation tools were reviewed to determine if any satisfied these core features. Table 1 summarizes the software explored and indicates which core features they currently support.

Table 1 Comparison of existing aircraft sizing and geometry modeling/visualization tools.

	SUAVE	**LUCAS**	**ISSAAC**	**PEACE**	**OpenVSP**	**ADEBO**	**ESP**	**JPAD Modeller**
Static OML Visuals	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Propulsion Architecture Visuals	No	No	Yes	No	No	No	No	No
Dynamic OML Visuals	No	No	No	Yes	No	No	No	Yes
Open-Source	Yes	No	No	No	Yes	Yes	Yes	No

Several existing conceptual aircraft design tools incorporate geometric modeling into their functionality, each with its own strengths and areas of focus. A literature survey indicates that there are several system- and subsystem-level aircraft design tools already available as well as other aircraft-focused geometric modeling tools. Each of these tools must be assessed based on the core capabilities described previously in this section, particularly how each one incorporates geometry modeling into their computational tool.

The Stanford University Aerospace Vehicle Environment (SUAVE) [6] is an open-source aircraft sizing tool developed for multi-fidelity conceptual aircraft design. This is achieved by leveraging physics-based models for higher-fidelity computations and empirical relations from historical data for lower-fidelity approximations. The configuration's geometry is defined within SUAVE using components such as fuselages, nacelles, wings, and vertical tails [7]. The geometry can be visualized in the tool itself [8], or with other geometry modeling tools like OpenVSP [9]. SUAVE is able to accommodate both conventional and unconventional aircraft designs, and model novel propulsion architectures, making it an attractive option. While SUAVE can represent the OML graphically, the propulsion architecture cannot be visualized. This limits the tool's effectiveness for this work.

The Library for Unified Conceptual Aircraft Synthesis (LUCAS) [10] is used to estimate aircraft performance of electrified or fully electric propulsion architectures and make comparisons to conventional aircraft using gas-turbine engines as their powerplant. LUCAS is intended to serve as a wrapper around SUAVE and includes more pre- and post-processing functionality for design space exploration. As a result of this coupling, LUCAS inherits the same advantages and disadvantages associated with SUAVE – the OML can easily be visualized, but the propulsion architecture cannot. From the literature review, LUCAS does not appear to be open-source and is a privately maintained code.

The Integrated Subsystems Sizing and Architecture Assessment Capability (ISSAAC) [11] is used to explore many electrified propulsion subsystem architectures for a single aircraft design. Prior to exploring the candidate subsystem architectures, an aircraft and its engines are sized according to a set of mission and point performance requirements.

Additional descriptors such as the fuselage and aerodynamic configuration are also selected a priori. To identify the best subsystem architecture, a “matrix of alternatives” is prescribed, which list out different components to be included in the subsystem architecture. Using the details provided, static visuals of the aircraft’s OML and propulsion system within the airframe are generated. Since ISSAAC is intended for exploring different subsystem architectures, the visuals generated are static and do not change while the analysis is executed. Given that the visualizations are not updated dynamically as the aircraft and its subsystems are sized, ISSAAC may not be the most suitable for incremental design adjustments, particularly at the early-phase conceptual design level that FAST operates at. Based on the available literature, ISSAAC does not appear to be publicly available, nor open-source.

The Parametric Energy-based Aircraft Configuration Evaluator (PEACE) [12] is a Matlab-based aircraft design framework suited for electric Vertical Take-Off and Landing (eVTOL) vehicle design as well as other novel configurations. The vehicle sizing framework accounts for the geometry, thus allowing it to iteratively resize the vehicle subject to a set of design requirements and/or point performance parameters [13]. For some analyses, the geometry is exported to FlightStream for more detailed aerodynamic analyses [12]. As a result, the user has multiple opportunities to view the geometry, both in PEACE and in the external tools that it relies upon. However, PEACE is focused on system-level design, and only includes the thrust sources in its visualizations. The components within the propulsion system are unable to be visualized, which limits its effectiveness for this work. Lastly, PEACE does not appear to be publicly available, nor is it mentioned that it is open-source in the literature.

The Aircraft DEsign BOx (ADEBO) [14] is a Matlab-based, open-source aircraft design environment specifically for fixed-wing aircraft during both conceptual and early preliminary design phases. In order to visualize an aircraft design, ADEBO interfaces with OpenVSP [15], leveraging its strengths in geometric modeling. However, it inherits the same visualization limitations, particularly the absence of propulsion architecture schematics, which restricts its use for integrated propulsion system design.

JPAD Modeller [16] is a commercially available tool used for preliminary aircraft design, providing more detailed analyses than what is typically available at the conceptual design phase. Since the tool is commercially available, it is not open-source and requires users to pay for a subscription. This software generates user defined geometries that can then be exported to an external CAD software for more detailed design. It also allows the user to generate constraint diagrams, size an aircraft, and observe how the geometry is impacted by changing the requirements. However, it does not allow the user to visualize a schematic of the propulsion architecture. Only the engine’s OML is included, which restricts its usefulness for this work.

OpenVSP [9] is a well-established, open-source parametric aircraft geometry tool originally developed by NASA. It is widely used for its flexibility in creating geometric representations of aircraft. Users create a geometric representation of their aircraft concept using preset components, or can define ones on their own. Rather than defining the aircraft’s design parameters, preset ones are provided for each component and the user moves slider bars in a separate window to transform the geometry. This allows for an endless number of different aircraft configurations to be generated, including novel designs. Furthermore, users can analyze their geometry with low-fidelity analysis tools, such as an aerodynamic analysis using a Vortex Lattice Method or obtaining the configuration’s mass properties. Despite these strengths, OpenVSP operates mainly as a geometry builder and can be integrated within an aircraft sizing code to provide visualization capabilities. Currently, OpenVSP only contains propeller and nacelle components, and does not allow for more detailed propulsion architectures to be included in the design.

The Engineering Sketch Pad (ESP) [17] is an open-source code and creates/manipulates geometries for aerospace applications. It effectively represents an aircraft’s OML, but does not support propulsion architecture visualizations. While users could manually integrate propulsion system components, this process may prove to be cumbersome and less efficient for rapid design iterations. Furthermore, ESP requires the geometry to be rebuilt whenever the aircraft design is updated, which is less practical for interfacing with fast-paced sizing tools like FAST.

After a thorough review of existing aircraft sizing and geometry modeling/visualization software, it became evident that none fully address the core capabilities required for a comprehensive visualization tool within FAST. Although SUAVE and LUCAS are effective for sizing and propulsion modeling, they do not currently offer the ability to propulsion architectures. Similarly, tools like OpenVSP, ADEBO, and ESP excel in geometric modeling, but they lack the capacity to integrate propulsion architecture visualizations and do not easily support dynamic updates to the aircraft’s OML during the sizing process. JPAD Modeller and PEACE offer the ability to dynamically update the OML, yet neither provide the functionality to generate detailed schematics of electrified propulsion architectures, which is essential for analyzing advanced aircraft configurations. ISSAAC, on the other hand, can represent both OML and propulsion architecture, but it does not support real-time visualization during the sizing process, limiting its effectiveness for iterative design exploration. Given that none of the existing tools comprehensively address all core needs, a dedicated visualization feature was developed specifically for FAST. This new tool aims to bridge these gaps, providing a seamless and dynamic interface to support designers in visualizing and refining aircraft concepts across a variety of propulsion architectures.

The remainder of this paper outlines the process of generating geometry models of aircraft concepts within FAST, and explains how these models are integrated into the tool's aircraft sizing code. Section III details the main visualization components for the both OML and propulsion architecture schematics. Section IV explains how the visualization tool interfaces with FAST to dynamically update the aircraft's shape/size during the sizing process. Lastly, Section V summarizes the work performed and provides recommendations for future work.

III. Visualization Capabilities

Like FAST, the visualization feature must rapidly generate aircraft geometries to allow a designer to explore a variety of aircraft configurations in a short period of time. Therefore, the geometric components must be detailed enough to represent dozens of aircraft configurations, yet streamlined to ensure that visualization generation remains quick and intuitive. Based on other visualization tools reviewed [6, 9], using pre-built, customizable components simplifies the modeling process while providing enough flexibility and freedom to the designer. To this extent, two main geometric components are used in this visualization package: "Lifting Surfaces" and "Blunt Bodies". These two components are sufficiently versatile to represent multiple aircraft elements, thus allowing novel configurations to be designed while simplifying the process by requiring familiarity with only two component types.

Lifting surfaces (detailed in Section III.A) are three-dimensional components composed of NACA airfoil cross-sections, and are used to model elements such as wings, horizontal and vertical stabilizers, and canards. Blunt bodies (discussed in Section III.B) are also three-dimensional components, composed of superellipse cross-sections (superellipses are discussed in Section III.B.1), and can be used to represent fuselages, fuel pods, and engine casings.

In addition to these core components provided, an engine creator (described in Section III.C) provides pre-built nacelle and propeller components that can be easily customized to suit a variety of aircraft designs. Flexibility is further extended to the visualization of propulsion architecture (explained in Section III.D), which is created according to propulsion architecture matrices as defined by Cinar et al. [18]. This framework provides a methodology in which conventional and electrified propulsion architectures can be represented using a graph theory inspired approach.

Before exploring these components and features in more detail, a FAST-generated visualization of Lockheed Martin's LM-100J is provided in Fig. 1 to showcase one of many aircraft that can be visualized in FAST. This visualization can be compared to one from a Lockheed Martin sales brochure in Fig. 2. All components and design parameters used to produce the LM-100J visualization are provided in Appendix V.A. Top, side, front, and isometric views of the aircraft configuration are easily generated by storing each component as an array of three-dimensional coordinates.

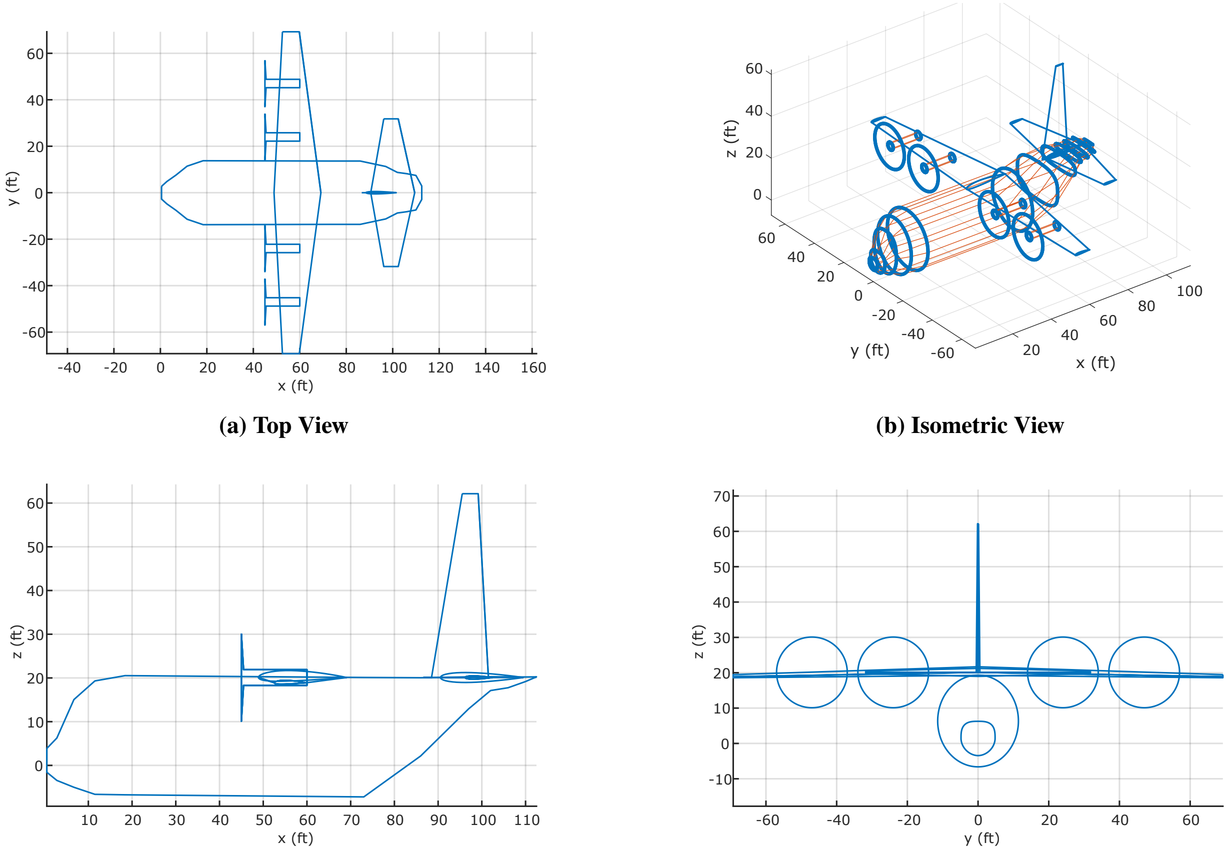
A. Lifting Surface Components

Aircraft configurations contain various aerodynamic surfaces that differ in number and location across the design. Formulating a general definition for these surfaces enables this variety to be represented using a universal set of input parameters, accommodating any configuration, such as the LM-100J shown in Fig. 1 or other aircraft designs. The visualization feature within FAST defines lifting surface components to represent these aerodynamic surfaces. These components, composed of NACA 4-digit airfoils, can represent a wing, horizontal tail, vertical tail, canard, or similar structures. The specific inputs needed to generate lifting surfaces are designed to provide the user with flexibility, as detailed in Section III.A.2.

1. NACA 4-Digit Airfoil Creator

A NACA 4-digit airfoil creator serves as the standard airfoil template for lifting surfaces within the visualization feature of FAST. It is important to note that this airfoil data is used exclusively for visualization purposes and is not employed in FAST's analysis routines. NACA 4-digit airfoils were selected due to their simplicity and well-documented performance characteristics, which make them ideal for early-stage design where detailed airfoil data may not be readily available. Unlike more complex airfoil series that require extensive data for accurate visualization, NACA airfoils can be easily parameterized and adjusted. Furthermore, the equations for the NACA series are publicly available, which makes them more accessible and easier to implement in open-source software.

Users can specify any valid NACA 4-digit airfoil to represent a lifting surface component. Equations 1 through 3 define the camberline, thickness, and angle for a unit 4-digit airfoil [20]. Equations 4 through 7 are used to compute the



(a) Top View

(b) Isometric View

(c) Side View

(d) Front View

Fig. 1 FAST-generated visualization of the LM-100J configuration.

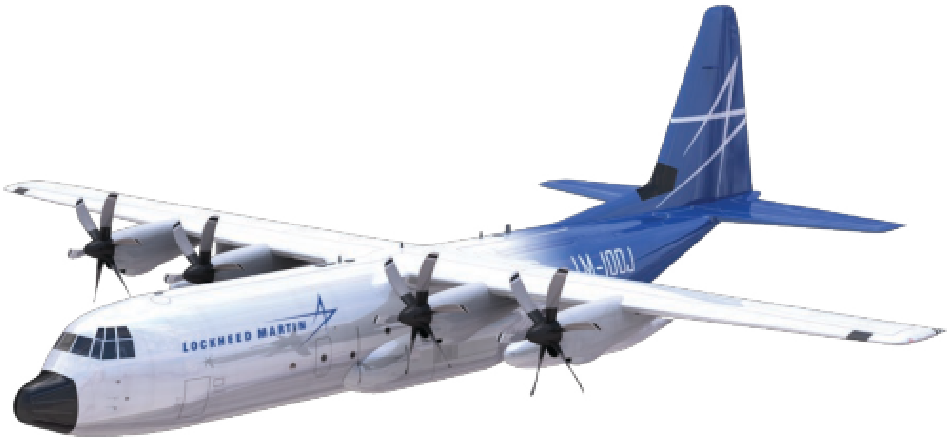


Fig. 2 LM-100J drawing from a Lockheed Martin sales brochure [19].

upper and lower surface coordinates.

All 4-digit NACA airfoils are encoded as $mptt$. The m designates the maximum camber as a percentage of chord length. Next, the p designates the position of this maximum camber in tenths of the chord length. Finally, the tt is the maximum thickness as a percentage of the chord.

$$y_c = \begin{cases} \frac{m}{p^2}(2px - x^2), & 0 \le x \le p \\ \frac{m}{1-p^2}[(1-2p) + 2px - x^2], & p \le x \le 1 \end{cases} \tag{1}$$

$$y_t = \pm\frac{t}{0.2}\left(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4\right) \tag{2}$$

$$\theta = \arctan\left(\frac{dy_c}{dx}\right) \tag{3}$$

$$X_U = x - y_t \sin\theta \tag{4}$$

$$Y_U = y_c + y_t \cos\theta \tag{5}$$

$$X_L = x + y_t \sin\theta \tag{6}$$

$$Y_L = y_c - y_t \cos\theta \tag{7}$$

2. Lifting Surface Creator

The lifting surface creator in FAST is designed to streamline the generation of aerodynamic components, allowing designers to quickly define shapes and positions without extensive manual input. To achieve this, the visualization feature requires a concise set of inputs that define the component's geometric properties, orientation, and positioning. These inputs include:

- **Geometric Properties:** Area, aspect ratio, taper ratio, quarter-chord sweep, and dihedral angle.
- **Positioning:** The 3-dimensional position of the midpoint of the root chord, specified as an xyz-coordinate triplet.
- **Special Properties:** A flag to indicate whether the component is symmetric, and a string input defining the orientation plane.
- **Airfoil Definition:** A string representing the NACA 4-digit airfoil code to be used across the entire lifting surface.

For simplicity and efficiency, all lifting surfaces are represented as trapezoids with a constant taper ratio, sweep, and dihedral. A single airfoil is used along the entire span of the wing. Since FAST is an early phase conceptual design tool, these simplifications strike a balance between ease of use and visual accuracy. Variable taper and sweep designs, which require more detailed specifications, are not essential at this stage; the use of consistent taper and sweep allows for rapid visualization of initial design concepts.

The wing shape is dictated by the aspect ratio, taper ratio, quarter-chord sweep, and dihedral. The wing size is impacted by the wing area only. By defining the position of each lifting surface, it is easier to make more generic aircraft configurations. This allows wings, canards, and horizontal/vertical stabilizers to be defined using the same component type. The inclusion of dihedral as an input is essential for generating features like butterfly tails, enhancing the range of configurations that can be represented.

The **Symmetric** flag defines whether a lifting surface is reflected across a symmetry plane or not. The symmetry plane used to reflect half of the lifting surface is one that is co-planar with the lifting surface's root airfoil. Turning the symmetry on/off is useful for distinguishing between a vertical stabilizer (which does not necessarily need to be reflected) and a horizontal stabilizer (which might need to be reflected).

The **Orientation** input specifies the plane in which the airfoil is generated, making the lifting surface component more versatile. Designers can choose between the xz, xy, and yz orientations:

- xz **orientation**: Typically used for main wings and horizontal stabilizers.
- xy **orientation**: Commonly applied to vertical stabilizers.
- yz **orientation**: Useful for unconventional configurations or for smaller fins on the upper/lower fuselage for purposes such as instrumentation or communications.

Figure 3 illustrates how variations in orientation, symmetry, and position affect the final configuration. The lifting surfaces at the front and back of the isometric view are created in the xz- and xy-planes, respectively. The two surfaces with airfoils in the xz-plane have identical geometric properties but differ in their x-axis positions and symmetry designations. Similarly, the two lifting surfaces in the yz-plane share similar shapes but differ in symmetry and y-axis positions.

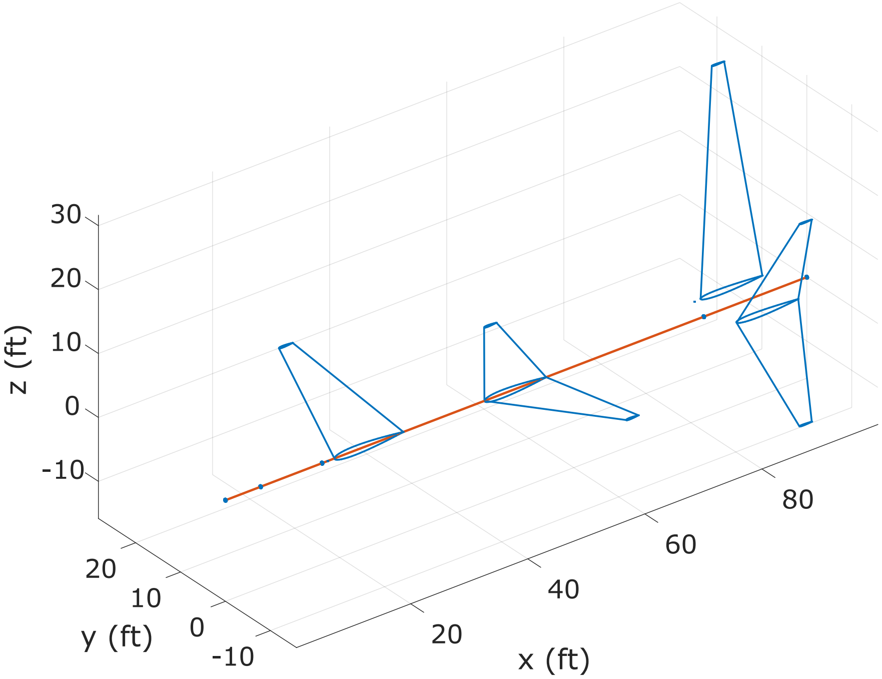


Fig. 3 Different wing orientations and symmetries available in the visualization feature.

FAST allows designers to incorporate an unlimited number of lifting surface components into a configuration, with no restrictions on their placement. This flexibility supports the creation of complex aircraft geometries, such as the supersonic transport shown in Fig. 4. This example features four lifting surface components: a delta wing, canard, and two vertical stabilizers. The design parameters to reproduce the supersonic transport example are provided in Appendix V.B.

B. Blunt Body Components

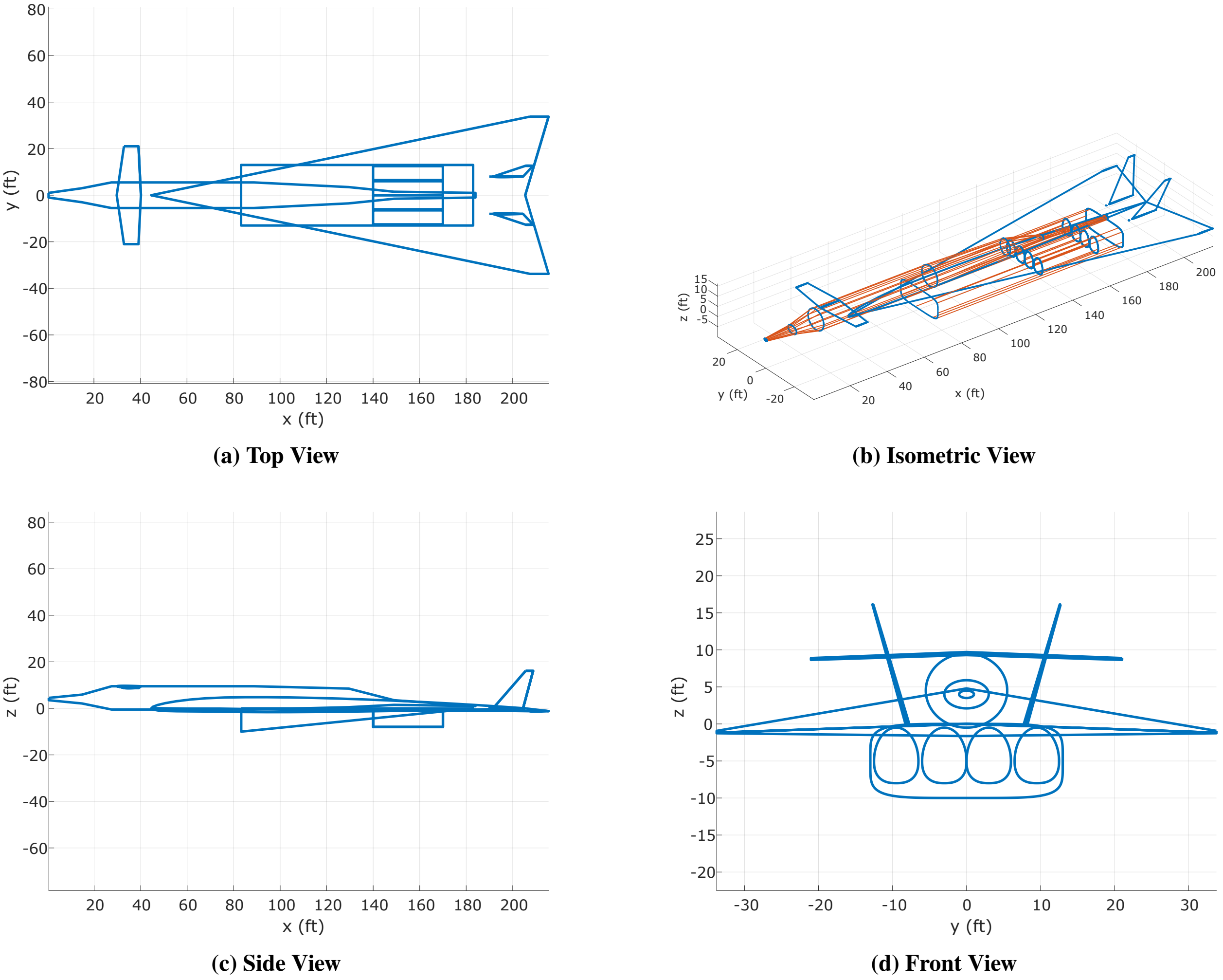
In order to create tube-and-wing aircraft designs, a generic component is needed to model a variety of fuselage shapes and sizes. Some configurations, such as the LM-100J, have a distinct shape to their nose, as illustrated in Fig. 2. To fill this need, a "Blunt Body" component was developed.

Blunt body components are composed of superellipses, a generic cross-sectional shape that provides the user great flexibility to generate any fuselage shape desired. The superellipse lends itself to endless cross-sectional shapes, thus allowing for future concepts to be easily represented while also providing a basis for conventional tube-and-wing configurations to be illustrated.

By leveraging the flexibility of the superellipses, other pre-built engine components (nacelles and propellers) were also developed and included in the visualization feature for added user convenience.

1. Superellipse Overview

Superellipses are a generic shape defined parametrically by two radii and a curvature parameter, known as the "power", in each of its four quadrants. The parametric equations for a superellipse are provided in Eqs. 8 and 9, where a and b represent the radii, and n is the power in a given quadrant of the superellipse. Typically, a superellipse is defined by the position of its center (an xyz-coordinate), four cardinal radii – northern, southern, eastern, and western – and four



(a) Top View

(b) Isometric View

(c) Side View

(d) Front View

Fig. 4 Supersonic transport created with FAST's visualization feature, showcasing multiple lifting surface components.

powers – northeastern, northwestern, southwestern, and southeastern [17]. Depending on the part of the superellipse being traversed, the eastern/western radii and northern/southern radii can be substituted in for a and b, respectively. For each quadrant, n is represented by one of the four powers listed previously. In total, each superellipse requires 9 inputs: four radii; four powers; and an xyz-coordinate to center the superellipse.

$$x(t) = \pm a \cos^{\frac{2}{n}}(t), 0 \leq t \leq \frac{\pi}{2} \tag{8}$$

$$y(t) = \pm b \sin^{\frac{2}{n}}(t), 0 \leq t \leq \frac{\pi}{2} \tag{9}$$

Figure 5 shows the wide range of cross-sections that can be created using superellipses. The upper 3-by-3 grid showcases different shapes, including an ellipse, a rectangle, and a concave star. In the first row of superellipses, the eastern/western radii are greater than the northern/southern radii, making the shapes wider. In the second row of superellipses, both the eastern/western radii are equal to the northern/southern radii. In the third row of superellipses, the eastern/western radii are less than the northern/southern radii, making the superellipses taller.

In the first column of the 3-by-3 grid, the superellipses are drawn with powers of 2. This creates a smooth, convex, elliptical shape without introducing any corners into the shape. In the second column of the 3-by-3 grid, the superellipses are drawn with powers much greater than 2. This begins to "pinch" the shape and creates corners in each quadrant,

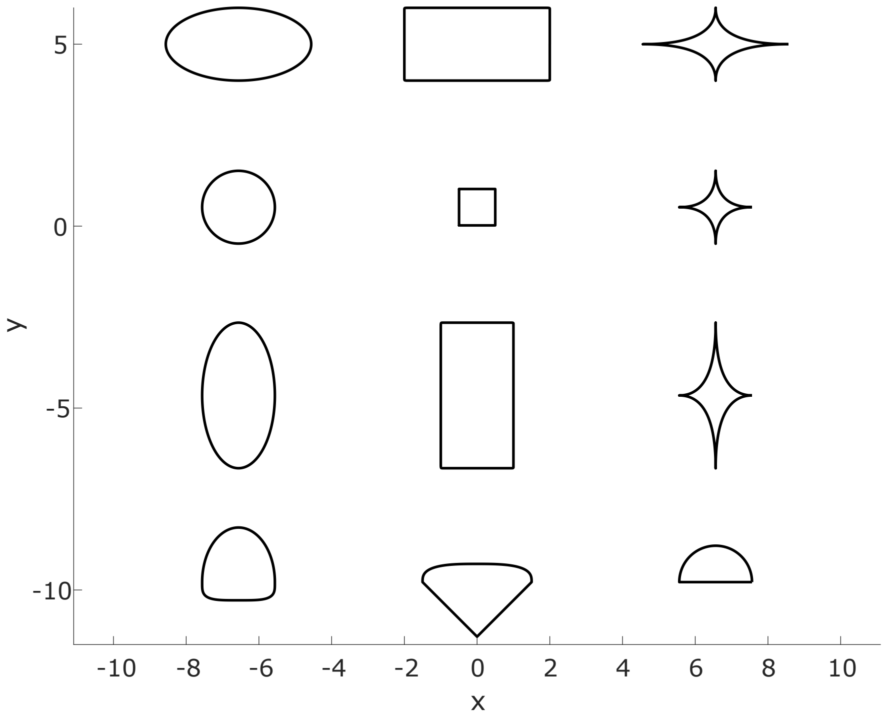


Fig. 5 Showcase of superellipse shapes.

making them appear rectangular. Lastly, in the third column of the 3-by-3 grid, the superellipses are drawn with powers less than 1, thus creating a concave shape rather than a convex one.

The last row in Fig. 5 demonstrates the diversity of superellipses, with examples including an exaggerated engine nacelle, a diamond, and a semi-circle. These examples highlight the adaptability of superellipses, making them an ideal choice for modeling aircraft fuselages. Their flexibility supports a range of conventional and innovative designs, easily accommodating complex profiles that go beyond simple cylindrical shapes.

For instance, Fig. 6 depicts a blended wing body (BWB) configuration created using the blunt body component within FAST's visualization feature. The design parameters used to create this example are in Appendix V.C and can be replicated in FAST. Moving along the fuselage in the $+x$-direction, the eastern/western radii grow drastically, particularly where passengers would be seated, while the northern/southern radii remain relatively constant. This adaptability allows the superellipses apture more nuanced aspects of the fuselage design, especially for the oblong shape typical of BWB configurations.

Additionally, superellipses have proven effective in modeling aerodynamic shaping, such as the area rule applied to the supersonic transport illustrated earlier in Fig. 4.

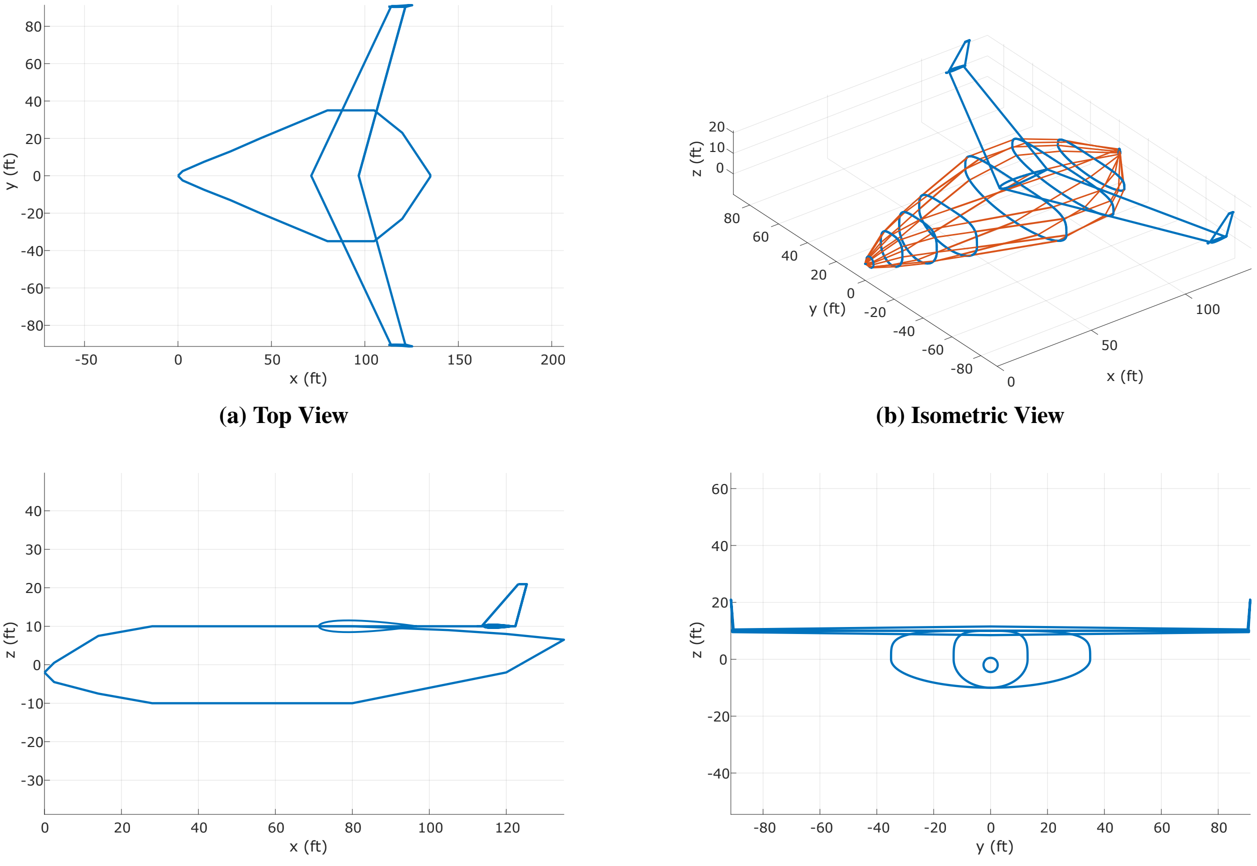
2. Fuselage Creator

Building on the introduction of superellipses, the Fuselage Creator in FAST utilizes these shapes to construct diverse fuselage designs. By specifying the parameters and positions of multiple superellipses, a complete fuselage can be generated, with the superellipses connected by "stringers" (line segments) to form a smooth, continuous surface.

To define a fuselage component, the following inputs are required:

1) The parameters of all superellipses, including their radii, powers, and positions.
2) The total length of the fuselage.
3) The locations of the forward and aft pressure bulkheads.

The total fuselage length and the positions of the forward and aft pressure bulkheads are used to scale the fuselage non-uniformly. In some cases, the designer may want to create a family of aircraft, which have the same nose- and tail-cones, but have longer/shorter fuselages to carry different payloads. The Fuselage Creator allows the designer to use the same set of superellipses, but lengthen/shorten the fuselage accordingly to accommodate the different payload



(a) Top View

(b) Isometric View

(c) Side View

(d) Front View

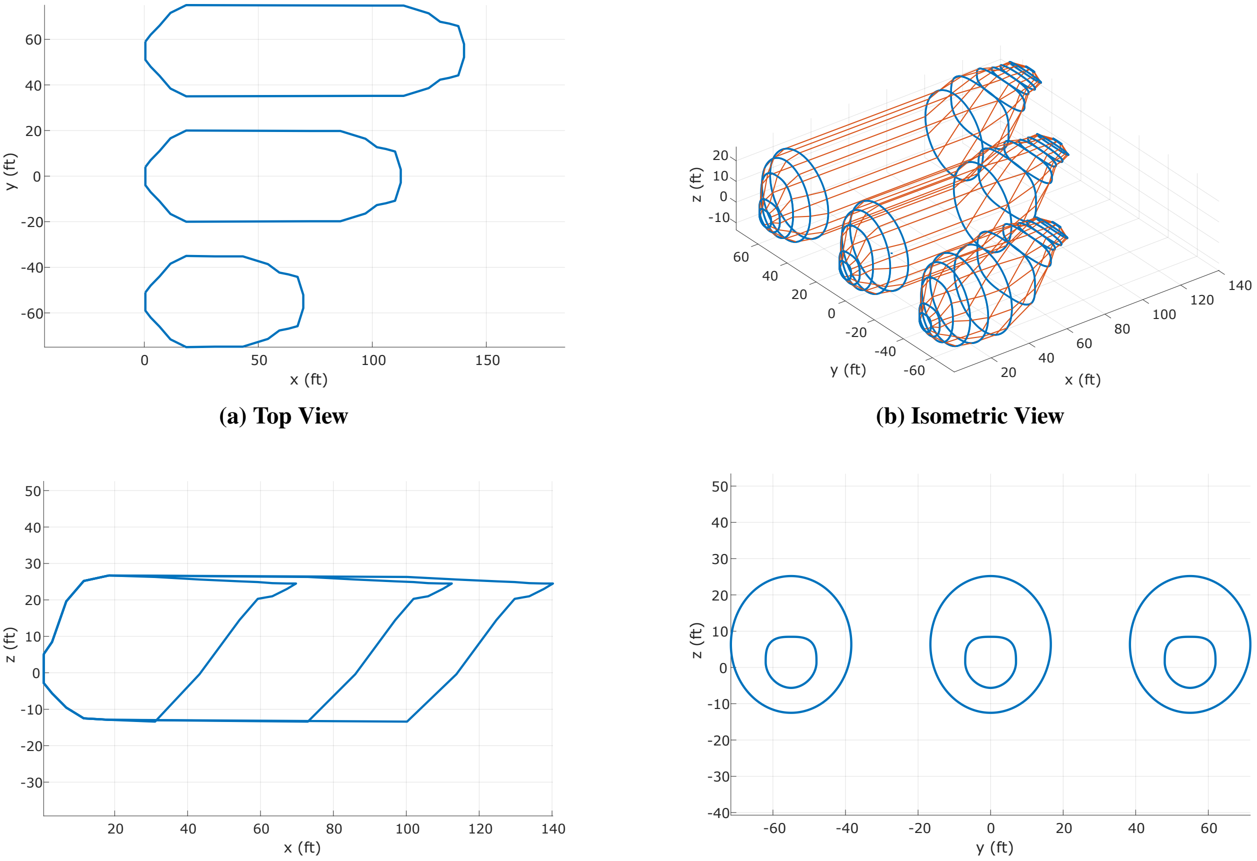
Fig. 6 BWB configuration with a fuselage made of superellipses.

requirements for each aircraft in the family. Any portion of the fuselage between the pressure bulkheads is assumed to carry payload, and can be lengthened/shortened according to the user specifications. The portions of the fuselage in-front of the forward pressure bulkhead and behind the aft pressure bulkhead are considered to be the nose-cone and tail-cone, respectively. These two regions cannot be lengthened/shortened because they contain the necessary flight controls and instrumentation to fly the aircraft as well as an auxiliary power unit on tube-and-wing aircraft.

For unconventional aircraft designs, such as the BWB configurations, the pressure bulkheads can be positioned at the very front and rear of the fuselage to allow for uniform scaling of the entire body. This flexibility ensures a smooth, continuous fuselage without abrupt transitions, making it easier to accommodate unique design features without introducing unwanted cusps or corners.

Figure 7 illustrates an example of three different fuselage configurations inspired by the LM-100J, showcasing different lengths to accommodate different payload capacities. The fuselage in the middle represents the nominal LM-100J fuselage (112.75 ft), while the other two are adjusted to shorter (70.00 ft) and longer (140.50 ft) lengths, respectively. These variations can be achieved by using the design parameters from the "Fuselage" component in Appendix V.A. By using the same shape but prescribing a different fuselage length (or payload requirement), additional design configurations can be represented with minimal changes to the user's inputs. This capability showcases the visualization feature's flexibility to generate multiple configurations based on a single user-defined shape.

In addition to creating fuselages, other blunt body objects can also be generated with the Fuselage Creator. For example, the engine casing in Fig. 4 used the Fuselage Creator by defining two superellipses to enclose the four engines underneath the fuselage. This is yet another way in which the Fuselage Creator can support the user in creating customizable components for complex aircraft configurations by using only one generic shape.



(a) Top View

(b) Isometric View

(c) Side View

(d) Front View

Fig. 7 Different fuselage lengths based on the LM-100J, illustrating variants with modified payload capacities.

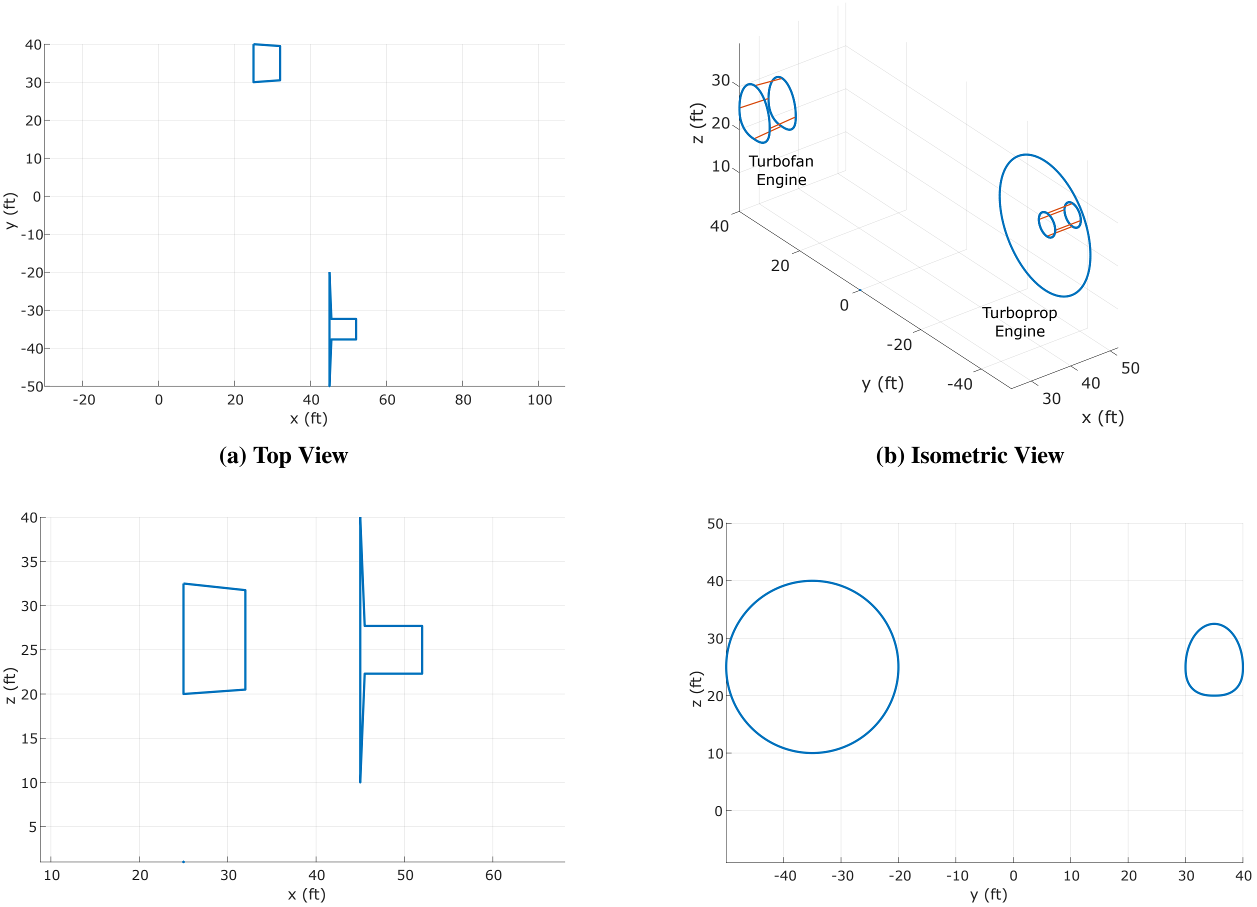
C. Engine Creator

All conventional aircraft in FAST's database utilize either turboprop/shaft or turbofan engines, making these components common across various configurations. To streamline the design process, FAST's visualization feature includes a pre-built library of engine components, which are created by connecting multiple superellipses to achieve the desired shape for turbofan and turboprop engines. This allows a designer to scale the length, width, and height of the engines according to their design specifications. This flexibility enables rapid adaptation of standard engine shapes to suit different aircraft designs, without the need for manual modeling. To visualize an engine, the user must input:

- **Inlet Radius:** the radius at the inlet of the engine.
- **Outlet Radius:** the radius at the outlet of the engine.
- **Engine Length:** the length of the engine.
- **Engine Position:** an xyz-coordinate pair to show where the center of the engine's inlet should be placed.
- **Engine Type:** a string to indicate whether a fan or propeller should be made.

To define an engine component for visualization purposes, users first specify whether the engine type is a "turbofan" or "turboprop." Despite the specific names, the components are actually more generalizable. The "turbofan" component can represent any ducted engine or nacelle, not specifically a turbofan engine. The "turboprop" component can represent any propeller aside from a turboprop engine. This flexibility allows the user to also model electrified propulsion system components like a turbogenerator or electrified propulsor.

Then, the user must also prescribe the engine's inlet/outlet radii (the front/rear widths of the engines), length, and position. With these inputs, FAST creates the engines automatically by re-shaping the pre-built components. Example turbofan and turboprop engines are provided in Fig. 8. The turbofan engine is represented by the two similarly-sized superellipses while the turboprop is represented by a large superellipse (to showcase the propeller) and a smaller superellipse (to showcase the engine/shaft).



(a) Top View

(b) Isometric View

(c) Side View

(d) Front View

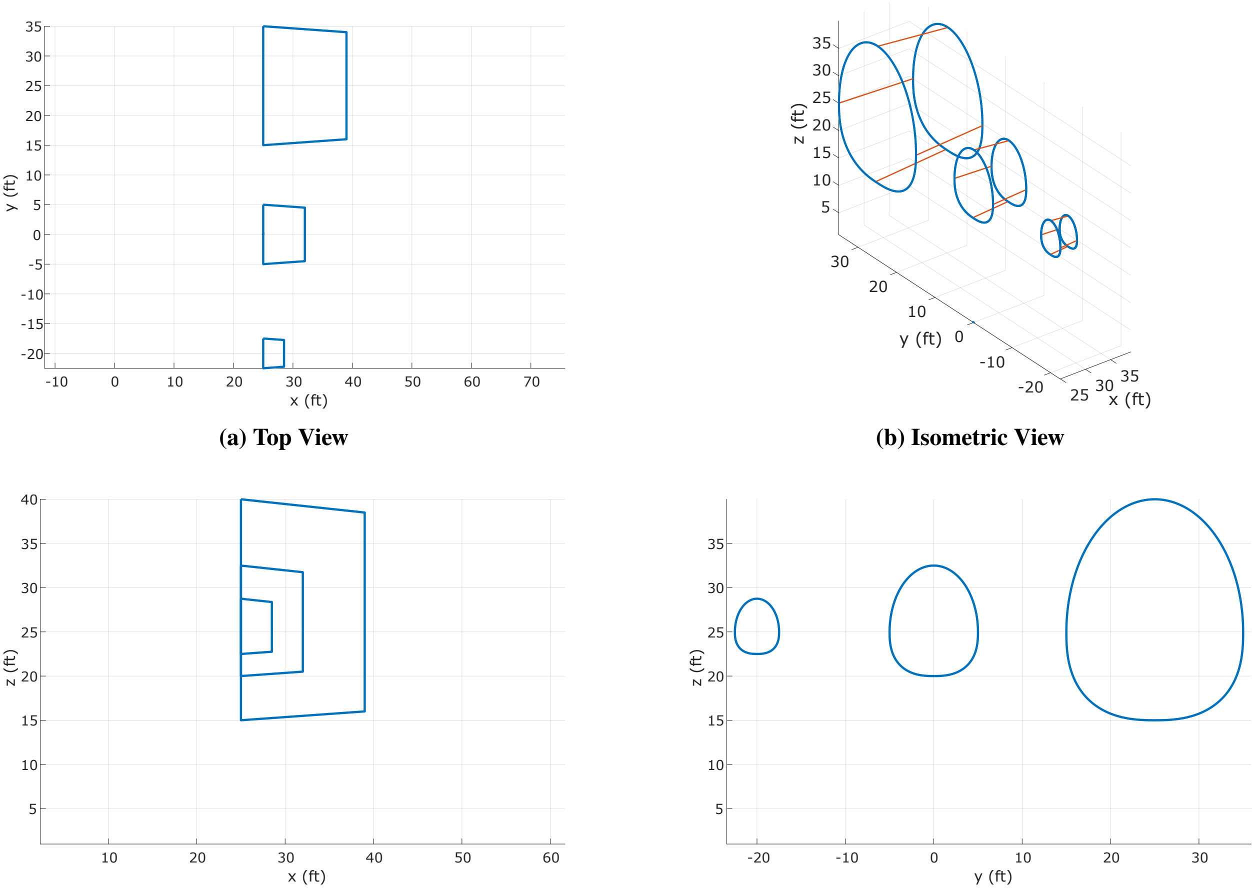
Fig. 8 Notional turbofan and turboprop engine shapes generated with FAST's visualization feature.

The inlet and outlet radii allow designers to adjust the width at the front and rear of the engines. For turbofan engines, the radii correspond to the widths of the forward and aft superellipses, while for turboprop engines, they define the propeller and engine/shaft widths, respectively. Figure 9 showcases different configurations of turbofan engines, illustrating how varying radii and lengths can produce engines of different sizes. In these examples, while the x- and z-position of the engines' inlets are the same, the inner/outer radii and engine length are different between the different engines. Although the sizes vary, the dimensions of each engine are linear multiples of one another, ensuring a consistent bypass ratio across the different configurations.

The flexibility of the engine creator allows for rapid generation of unconventional configurations. For example, the SUSAN aircraft concept employs a unique propulsion architecture, featuring a turbofan engine mounted on the tailcone for boundary layer ingestion, alongside 16 electric propulsors distributed under the wing [4]. Figure 10 illustrates this novel electrified propulsion layout, recreated using FAST's visualization tool. In this example, all 16 electric propulsors are represented by identical turboprop models with the same inlet and outlet radii and engine lengths, differing only in their assigned positions.

D. Propulsion System Architecture

One of FAST's distinguishing features is its ability to analyze aircraft with *any* propulsion architecture. Given the breadth of novel propulsion architectures being developed for electrified aircraft configurations, part of the visualization feature is to provide schematic representations of these propulsion architectures. These schematics of the user-prescribed



(a) Top View

(b) Isometric View

(c) Side View

(d) Front View

Fig. 9 Examples of varying lengths and radii for a notional turbofan engine.

propulsion architectures help illustrate which components in the propulsion system are connected and inform the designer about where/how each component should be arranged in the actual aircraft design.

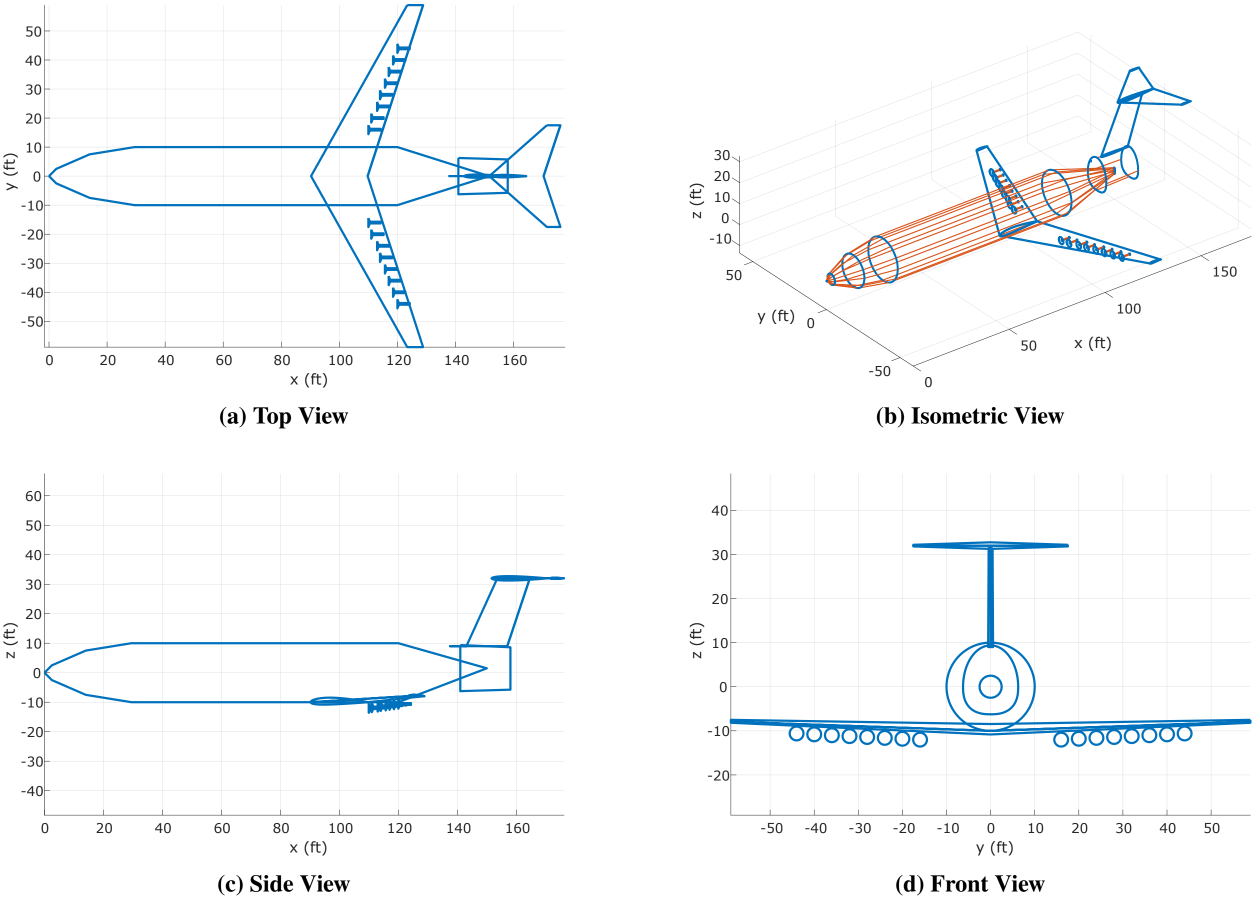
To represent the propulsion architecture in a simple yet comprehensive manner, Cinar et al. [18] introduced a graph theory-based approach. This method generalizes components within a propulsion architecture into three categories: energy sources, power sources, and thrust sources. Energy sources store energy (e.g., kerosene fuel, batteries, hydrogen), power sources convert energy into mechanical power (e.g., gas-turbine engines, electric motors), and thrust sources use this mechanical power to generate thrust (e.g., propellers, fans). By categorizing components this way, the connections between them can be efficiently represented using matrices.

The relationships between these components are defined by three matrices that specify direct connections:

- B_{PSES}: Connections between power sources and energy sources.
- B_{PSPS}: Connections between different power sources.
- B_{TSPS}: Connections between thrust sources and power sources.

An example of these matrices for a series hybrid-electric propulsion architecture is shown in Eq. 10. In this configuration, a turbogenerator (comprising of a gas turbine engine and electric generator coupled together to produce electrical power) and an electric motor are connected in series, with the electric motor driving a propeller or a fan. Additionally, a battery is connected to the electric motor to supplement the power from the turbogenerator, making the architecture "hybrid".

$$B_{\text{PSES}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; B_{\text{PSPS}} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \; B_{\text{TSPS}} = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{10}$$



(a) Top View

(b) Isometric View

(c) Side View

(d) Front View

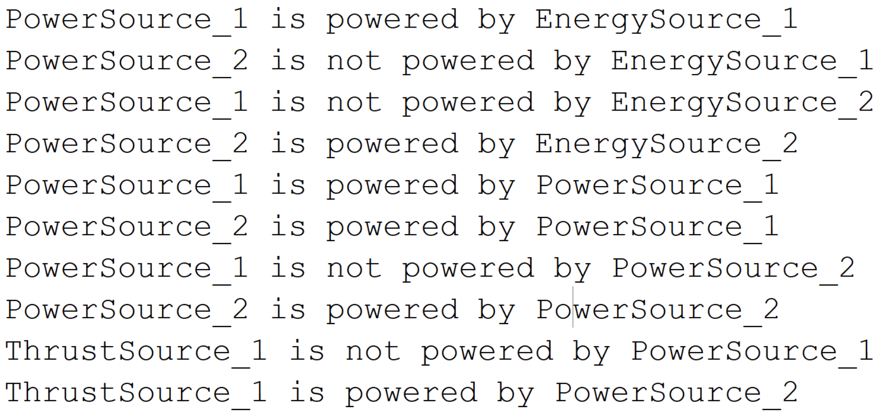
Fig. 10 SUSAN aircraft concept recreated using FAST's visualization feature.

Each entry in these matrices answers the question: "Does the component in column j *directly connect to* (or power) the component in row i?" An entry of 1 indicates a *direct connection* between components (answers "yes" to the previously posed question) while a 0 indicates *no direct connection* (answers "no" to the previously posed question). For example, the B_{PSES} matrix contains two rows and two columns. The first and second rows correspond to the turbogenerator and electric motor, respectively, while the first and second columns correspond to the kerosene fuel and battery, respectively. Since the entry in the first row, first column is a 1, it shows that the kerosene fuel powers the turbogenerator. Similarly, the 1 in the second row, second column shows that the battery is connected to the electric motor.

The B_{PSPS} matrix also contains two rows and two columns. The first row and column correspond to the turbogenerator and the second row and column correspond to the electic motor. In this propulsion architecture, the turbogenerator powers the electric motor, which is represented by the 1 in the second row, first column. Also, each power source is assumed to power itself, leading to all entries on the main diagonal being 1.

Lastly, the B_{TSPS} matrix contains one row and two columns. The only row corresponds to the propeller. The first and second columns correspond to the turbogenerator and electric motor, respectively. Since the electric motor drives the propeller/fan, a 1 is placed in the first row, second column. The turbogenerator is not *directly connected* to the propeller/fan, and explains why there is a 0 in the first row, first column.

To assist users in verifying these connections, FAST prints the component relationships to Matlab's command window, as shown in Fig. 11. The printouts provided match the matrices instantiated in Eq. 10. A component relationship is printed as a connection if it is driven by another component. In this example, "Energy Source 1" and "Energy Source 2" correspond to the kerosene fuel and battery, respectively; "Power Source 1" and "Power Source 2" correspond to the turbogenerator and electric motor, respectively; and "Thrust Source 1" corresponds to a propeller/fan.



```
PowerSource_1 is powered by EnergySource_1
PowerSource_2 is not powered by EnergySource_1
PowerSource_1 is not powered by EnergySource_2
PowerSource_2 is powered by EnergySource_2
PowerSource_1 is powered by PowerSource_1
PowerSource_2 is powered by PowerSource_1
PowerSource_1 is not powered by PowerSource_2
PowerSource_2 is powered by PowerSource_2
ThrustSource_1 is not powered by PowerSource_1
ThrustSource_1 is powered by PowerSource_2
```

Fig. 11 Example component connection printout

Once the matrices are defined, FAST's visualization tool generates a schematic of the propulsion architecture, as shown in Fig. 12. This example demonstrates how the graph theory-based approach converts the information from the matrices into a clear and informative visual representation, aiding the designer in understanding the system layout.

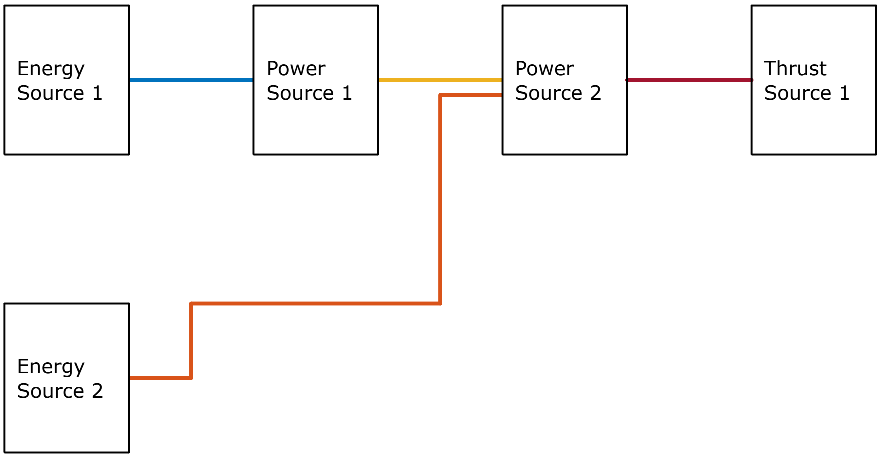


Fig. 12 Schematic of a notional series hybrid propulsion architecture.

To improve the visibility and comprehension of the schematic, the visualization tool arranges and places the components in one of four columns, and dynamically draws connections between the respective components. The leftmost and rightmost columns always represent energy sources and thrust sources, respectively. The second and third columns are for the power sources. Power sources in the second column are considered to be "driving" power sources, and either power another power source or a thrust source. Power sources in the third column are considered to be "driven" power sources, which receive power from another power source and are directly connected to a thrust source [18]. For example, the turbogenerator in the series hybrid-electric propulsion architecture is a driving power source (located in the second column), and the electric motor is a driven power source (located in the third column).

The flexibility and generic nature of this graph theory-based approach and propulsion architecture schematics is stress-tested using Eq. 11, which represents an independent parallel hybrid-electric propulsion architecture. In this example, "Power Source 1" represents an electric motor and is powered by a battery ("Energy Source 1"). The remaining power sources and energy sources are gas-turbine engines and kerosene fuel, respectively. Each of the power sources in the propulsion architecture drives its own thrust source.

$$B_{\text{PSES}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \; B_{\text{PSPS}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; B_{\text{TSPS}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

Figure 13 illustrates the schematic of the propulsion architecture. Since there are no power sources connected in series, only the first, second, and fourth columns hold the propulsion system components.

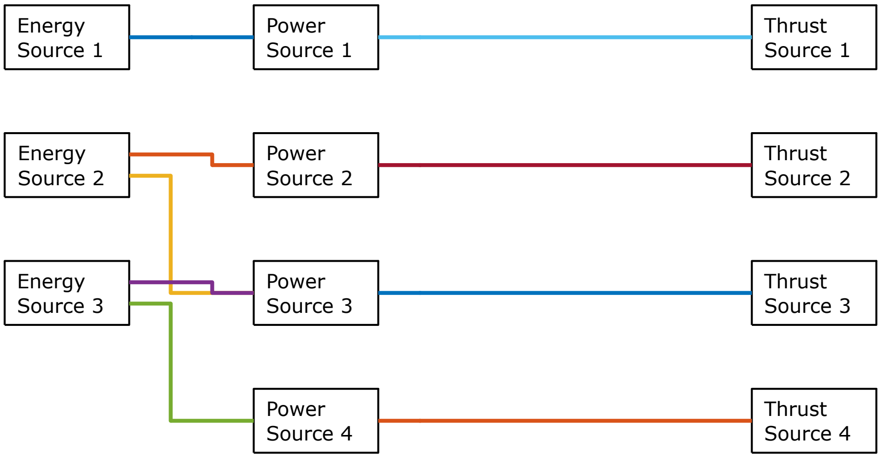


Fig. 13 Schematic of a notional independent parallel hybrid electric propulsion architecture.

IV. Interfacing with FAST

In addition to visualizing an aircraft's OML and propulsion architecture, NASA's EPFD team requested that the geometric configuration be dynamically re-drawn as the aircraft is sized in FAST. This feature allows users to observe how the design grows or shrinks relative to the initial aircraft configuration provided. This is also useful to evaluate real-world constraints on the design, such as ensuring that the wingspan is short enough so the aircraft can fit within existing airport gates. Such insights help identify whether all design constraints have been properly enforced in the aircraft analysis. By providing a visual comparison, it becomes easier to see if a constraint has been violated and helps gauge how the new design deviates from the original one.

To meet NASA's requirements, the visualization features listed above were integrated with FAST's aircraft sizing capabilities. The workflow for this integration is provided in Algorithm 1.

To visualize an aircraft configuration, the user may elect to select a pre-built aircraft configuration within FAST or prescribe their own. If neither of these are provided, then FAST selects a pre-built aircraft configuration based on the aircraft class provided (turbofan or turboprop/piston aircraft) and the nominal number of passengers/payload to be carried. For turbofan aircraft, current options include:

- A single-aisle tube-and-wing configuration for up to 160 passengers,
- A double-aisle tube-and-wing configuration for 160-400 passengers, and
- A twin-deck, double-aisle tube-and-wing configuration for more than 400 passengers.

For turboprop aircraft, current options include:

- A large turboprop configuration for 19 or more passengers (the threshold for FAR Part 23 and 25 aircraft), or
- A small turboprop for fewer than 19 passengers.

In addition to the aircraft configuration, users may specify the fuselage length. If this value is provided, the visualization feature draws the fuselage according to the user-specified length and lengthens/shortens it between the pressure bulkheads if necessary. If the fuselage length is not provided, then FAST uses the historical regressions outlined in Arnson et al. [21], which relate the passenger/payload requirements to fuselage lengths.

Before starting the sizing the aircraft in FAST, an initial configuration is plotted in blue. After each sizing iteration, all components except the fuselage are dynamically scaled based on the ratio of the updated wing area to the initial

Algorithm 1 Combined FAST-visualization workflow

```
Input: aircraft configuration; aircraft geometry (optional); and fuselage length (optional)
if aircraft geometry was not provided then
    Use the passenger/payload requirement and aircraft class to select a pre-built geometry
else
    Use the geometry prescribed by the user
end if
if fuselage length was not provided then
    Use historical regressions to estimate the fuselage length
else
    Use the fuselage length prescribed by the user
end if
Draw the initial aircraft configuration in blue
while aircraft is being sized do
    Remember the current wing area
    Size the aircraft
    Obtain the updated wing area
    Scale the geometry by the ratio of updated-current wing areas
    Overlay the updated geometry in red
end while
```

wing area. The fuselage remains the same size because the required payload – a design requirement – did not change. The new configuration is overlaid on the same plot as the initial configuration, but is plotted in red. This allows the geometric representation to dynamically grow/shrink as the aircraft concept is sized in FAST and allow the user to compare it to the initial configuration provided.

An example of this process is shown in Fig. 14, where a notional LM-100J configuration (shown previously in Fig. 1) is re-designed with an electrified propulsion architecture. Rather than using four gas-turbine engines to power four propellers, the outboard propellers are powered by electric motors instead. In the example, the larger wing area compensates for the increased fuel weight (and increased takeoff gross weight) required to meet the same TLARs, such as range, takeoff field length, cruise speed, etc. As a result of the increased wingspan, it is possible that the aircraft is now too large to fit at a standard airport parking stand (a requirement that was satisfied by the original concept). Aside from the larger wing, the engine diameter has also increased, indicating that the newly sized aircraft requires a larger engine due to the increased power demands. Additionally, larger stabilizers are needed to counteract the aircraft's pitching and rolling moments. Overall, this example illustrates how simply assessing the aircraft's geometry enables the designer to better understand the impact of one design decision on the system-level configuration (in this case, electrifying the propulsion architecture).

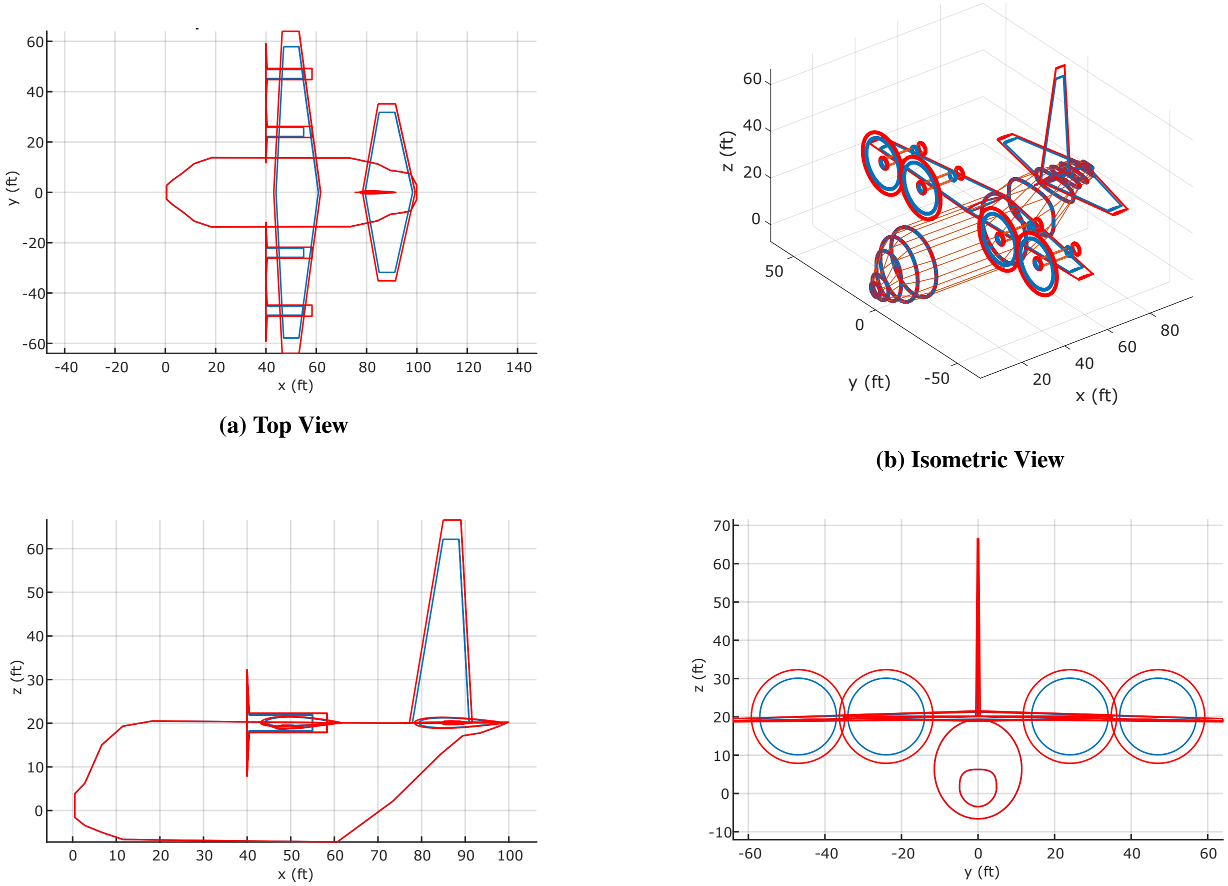
V. Conclusions and Future Work

This paper presents the development and integration of a dynamic visualization feature within FAST, an open-source, Matlab-based aircraft sizing tool designed to accommodate a wide range of propulsion architectures. The need for this feature was highlighted by NASA's EPFD team, who requested a tool to visualize aircraft concepts, examine propulsion system schematics, and observe how aircraft dimensions change during early-phase design. The latest version of FAST, equipped with this visualization capability, is available online [*].

The main contributions of this work can be summarized as follows:

- The tool enables users to **dynamically visualize aircraft configurations** during the sizing process. By overlaying initial and final designs, it provides visual feedback on the impact of design decisions, such as changes to wing area, engine size, and stabilizer dimensions. This approach helps identify potential issues where design constraints may not have been fully implemented, as well as giving a clear sense of how new designs differ from the initial configurations.
- It supports **a diverse range of aircraft configurations**, accommodating both conventional tube-and-wing designs and advance concepts such as blended wing bodies or distributed propulsion aircraft. Using a modular approach

[*] https://github.com/ideas-um/FAST



(a) Top View

(b) Isometric View

(c) Side View

(d) Front View

Fig. 14 Converged electrified-LM-100J-like configuration in red overlaid on the nominal LM-100J-like geometry in blue.

that combines lifting surfaces and blunt body components, the tool allows for efficient design generation. Users can either select from pre-built configurations or create custom designs, offering flexibility for a variety of design needs.

- The tool integrates **schematic representations of propulsion architectures** by using a graph theory-based framework that categorizes components as energy, power, or thrust sources. This approach enables users to visualize complex systems, including hybrid-electric and fully electric configurations, and understand how different components interact within the system. By simplifying the representation of propulsion architectures, the tool supports the design and analysis of a wide variety of aircraft concepts.
- To simplify the design process, the tool includes **customizable pre-built engine visuals**, such as turbofans (or any other ducted thrust-generating source) and turboprops. These models can be adapted to fit specific designs, allowing for quick customization without the need for extensive manual modeling.
- The visualization feature offers **a flexible workflow** that can function both as a standalone tool and in conjunction with FAST's sizing process. This integration allows users to track changes in design parameters throughout the sizing process, facilitating a clearer assessment of how high-level decisions impact the overall aircraft geometry.

Future development will focus on modeling the complete propulsion architecture within the aircraft's OML. This enhancement would enable designers to visualize how internal components, such as batteries, motors, and fuel tanks, are integrated within the aircraft, providing a more comprehensive view of how the propulsion system interacts with the overall design. Such an approach would support better optimization of internal layouts and space utilization.

Additionally, a more sophisticated approach to accounting for weight and balance requirements during the sizing process is another priority. Currently, components like stabilizers are scaled based on wing area. Adjusting these component sizes using more refined metrics, such as volume coefficients and center-of-gravity considerations, would lead to more accurate design outputs. This would allow the tool to better simulate how the aircraft configuration performs under different conditions, ensuring that it meets various design requirements.

These planned improvements aim to enhance the tool's ability to provide deeper insights into system-level design changes, making the design process more precise and efficient. By continuing to develop the visualization feature, FAST will support the evolving needs of the aviation industry, especially as new configurations and technologies emerge in the transition to electrified propulsion systems.

Appendix

A. LM-100J Design Parameters

Table 2 provides all of the design parameters necessary to reproduce the visualization in Fig. 1. For any wing position, the coordinates provided represent the location of the wing's leading edge at the root chord. Any input that must be a string is provided in "quotation marks".

Table 2 Design parameters for the LM-100J

Component	Parameter	Value	Units
Wing	Aspect Ratio	10.1	–
	Area	1,900	ft^2
	Taper Ratio	0.36	–
	Quarter-Chord Sweep	-5	deg
	Dihedral	-1	deg
	Leading Edge	(59, 0, 25)	ft
	Symmetric Wing Flag	1	–
	Wing Orientation	"xz"	–
	NACA Airfoil Code	"2412"	–
	Component Type	"liftingSurface"	–
Horizontal Tail	Aspect Ratio	5.05	–
	Area	800	ft^2
	Taper Ratio	0.33	–
	Quarter-Chord Sweep	-7	deg
	Dihedral	0	deg
	Leading Edge	(100, 0, 25)	ft
	Symmetric Wing Flag	1	–
	Wing Orientation	"xz"	–
	NACA Airfoil Code	"0012"	–
	Component Type	"liftingSurface"	–
Vertical Tail	Aspect Ratio	5.05	–
	Area	350	ft^2
	Taper Ratio	0.28	–
	Quarter-Chord Sweep	0	deg
	Dihedral	0	deg
	Leading Edge	(95, 0, 25)	ft
	Symmetric Wing Flag	0	–
	Wing Orientation	"xy"	–
	NACA Airfoil Code	"0008"	–
	Component Type	"liftingSurface"	–

Component	**Parameter**	**Value**	**Units**
Fuselage	Length	112.75	ft
	Component Type	“bluntBody”	–
Fuselage Cross-Sections	Pressure Bulkhead Positions	(18.907, 74.058)	ft
	Total Fuselage Length	112.75	ft
	Number of Superellipses	12	–
	Superellipse Radii	(3.950, 4.000, 3.950, 4.000) (7.040, 7.000, 7.040, 7.000) (14.575, 11.000, 14.575, 11.000) (18.850, 16.600, 18.850, 16.600) (19.800, 20.000, 19.800, 20.000) (19.850, 19.800, 19.850, 19.800) (13.000, 19.800, 13.000, 19.800) (5.300, 16.400, 5.300, 16.400) (2.300, 12.700, 2.300, 12.700) (1.800, 11.900, 1.800, 11.900) (0.750, 10.800, 0.750, 10.800) (0.150, 2.900, 0.150, 2.900)	ft
	Superellipse Powers	(3.0, 3.0, 2.0, 2.0) (3.0, 3.0, 1.9, 1.9) (2.5, 2.5, 2.0, 2.0) (2.0, 2.0, 2.0, 2.0) (2.0, 2.0, 2.0, 2.0) (2.0, 2.0, 2.0, 2.0) (3.0, 3.0, 3.0, 3.0) (3.0, 3.0, 3.0, 3.0) (2.0, 2.0, 3.0, 3.0) (2.0, 2.0, 3.0, 3.0) (2.0, 2.0, 3.0, 3.0) (2.0, 2.0, 3.0, 3.0)	–
	Superellipse Position	(0.500, 0.000, 1.150) (2.800, 0.000, 1.410) (6.700, 0.000, 5.025) (11.500, 0.000, 6.350) (18.400, 0.000, 6.900) (73.000, 0.000, 6.450) (86.000, 0.000, 12.600) (97.000, 0.000, 19.800) (102.000, 0.000, 22.600) (106.000, 0.000, 22.800) (110.000, 0.000, 23.750) (112.5, 0.000, 24.35)	ft

Component	Parameter	Value	Units
Fuselage Cross-Sections	Superellipse Views	NOTFVIEW FVIEW NOTFVIEW FVIEW NOTFVIEW NOTFVIEW NOTFVIEW NOTFVIEW NOTFVIEW NOTFVIEW NOTFVIEW NOTFVIEW	–
Engines	Length	15.0	ft
	Engine Inlet Radius	5.0	ft
	Engine Outlet Radius	4.5	ft
Engine Specifications	Number of Engines	4	–
	Engine Position	(45, +24, 25) (45, −24, 25) (45, +47, 25) (45, −47, 25)	ft
	Scaling Factors	(1, 2) (1, 2) (1, 2) (1, 2)	–
	Engine Type	TURBOPROP TURBOPROP TURBOPROP TURBOPROP	–

B. Supersonic Transport Design Parameters

Table 3 provides all of the design parameters necessary to reproduce the visualization in Fig. 4. For any wing position, the coordinates provided represent the location of the wing's leading edge at the root chord. Any input that must be a string is provided in "quotation marks".

Table 3 Design parameters for the supersonic transport

Component	Parameter	Value	Units
Wing	Aspect Ratio	0.8	–
	Area	5,700	ft^2
	Taper Ratio	0.05	–
	Quarter-Chord Sweep	55	deg
	Dihedral	-2	deg
	Leading Edge	(125, 0, 0)	ft
	Symmetric Wing Flag	1	–
	Wing Orientation	"xz"	–
	NACA Airfoil Code	"1204"	–
	Component Type	"liftingSurface"	–
Canard	Aspect Ratio	5.05	–
	Area	350	ft^2
	Taper Ratio	0.6	–
	Quarter-Chord Sweep	0	deg
	Dihedral	-2	deg
	Leading Edge	(35, 0, 9.5)	ft
	Symmetric Wing Flag	1	–
	Wing Orientation	"xz"	–
	NACA Airfoil Code	"0004"	–
	Component Type	"liftingSurface"	–
Right Vertical Tail	Aspect Ratio	2	–
	Area	130	ft^2
	Taper Ratio	0.25	–
	Quarter-Chord Sweep	40	deg
	Dihedral	30	deg
	Leading Edge	(198, 8, 0)	ft
	Symmetric Wing Flag	0	–
	Wing Orientation	"xy"	–
	NACA Airfoil Code	"0004"	–
	Component Type	"liftingSurface"	–

Component	Parameter	Value	Units
Left Vertical Tail	Aspect Ratio	2	–
	Area	130	ft^2
	Taper Ratio	0.25	–
	Quarter-Chord Sweep	40	deg
	Dihedral	-30	deg
	Leading Edge	(198, −8, 0)	ft
	Symmetric Wing Flag	0	–
	Wing Orientation	“xy”	–
	NACA Airfoil Code	“0004”	–
	Component Type	“liftingSurface”	–
Fuselage	Length	185	ft
	Component Type	“bluntBody”	–
Fuselage Cross-Sections	Pressure Bulkhead Positions	(27.75, 148)	ft
	Total Fuselage Length	185	ft
	Number of Superellipses	7	–
	Superellipse Radii	(0.5, 1.0, 0.5, 1.0) (1.9, 3.0, 1.9, 3.0) (5.0, 5.5, 5.0, 5.5) (5.0, 5.5, 5.0, 5.5) (4.0, 3.5, 4.0, 3.5) (1.0, 1.5, 1.0, 1.5) (1.0, 1.0, 1.0, 1.0)	ft
	Superellipse Powers	(2, 2, 2, 2) (2, 2, 2, 2) (2, 2, 2, 2) (2, 2, 2, 2) (2, 2, 2, 2) (2, 2, 2, 2) (2, 2, 2, 2)	–
	Superellipse Position	(0.5, 0.0, 4.0) (14.8, 0.0, 4.0) (27.5, 0.0, 4.5) (88.8, 0.0, 4.5) (129.5, 0.0, 4.5) (149.0, 0.0, 2.5) (184.0, 0.0, 0.0)	ft
	Superellipse Views	FVIEW FVIEW FVIEW NOTFVIEW NOTFVIEW NOTFVIEW NOTFVIEW	–

Component	**Parameter**	**Value**	**Units**
Casing	Length	185	ft
	Component Type	“bluntBody”	–
Casing Cross-Sections	Pressure Bulkhead Positions	(15.6, 29.4)	ft
	Total Fuselage Length	185	ft
	Number of Superellipses	2	–
	Superellipse Radii	(5.0, 13.0, 5.0, 13.0) (5.0, 13.0, 5.0, 13.0)	ft
	Superellipse Powers	(5, 5, 5, 5) (5, 5, 5, 5)	–
	Superellipse Position	(83.25, 0, −5) (183.00, 0, −5)	ft
	Superellipse Views	FVIEW NOTFVIEW	–
Engines	Length	30	ft
	Engine Inlet Radius	3	ft
	Engine Outlet Radius	3	ft
Engine Specifications	Number of Engines	4	–
	Engine Position	(140.0, +3.0, −5.0) (140.0, +9.5, −5.0) (140.0, −3.0, −5.0) (140.0, −9.5, −5.0)	ft
	Scaling Factors	(1, 1) (1, 1) (1, 1) (1, 1)	–
	Engine Type	TURBOFAN TURBOFAN TURBOFAN TURBOFAN	–

C. Blended Wing Body Design Parameters

Table 4 provides all of the design parameters necessary to reproduce the visualization in Fig. 6. For any wing position, the coordinates provided represent the location of the wing's leading edge at the root chord. Any input that must be a string is provided in "quotation marks".

Table 4 Design parameters for the blended wing body

Component	Parameter	Value	Units
Wing	Aspect Ratio	11	–
	Area	3,000	ft^2
	Taper Ratio	0.3	–
	Quarter-Chord Sweep	18	deg
	Dihedral	0	deg
	Leading Edge	(84, 0, 10)	ft
	Symmetric Wing Flag	1	–
	Wing Orientation	"xz"	–
	NACA Airfoil Code	"0012"	–
	Component Type	"liftingSurface"	–
Right Wingtip	Aspect Ratio	2	–
	Area	60	ft^2
	Taper Ratio	0.25	–
	Quarter-Chord Sweep	40	deg
	Dihedral	8	deg
	Leading Edge	(118, 90.5, 10)	ft
	Symmetric Wing Flag	0	–
	Wing Orientation	"xy"	–
	NACA Airfoil Code	"0004"	–
	Component Type	"liftingSurface"	–
Left Wingtip	Aspect Ratio	2	–
	Area	60	ft^2
	Taper Ratio	0.25	–
	Quarter-Chord Sweep	40	deg
	Dihedral	-8	deg
	Leading Edge	(118, −90.5, 10)	ft
	Symmetric Wing Flag	0	–
	Wing Orientation	"xy"	–
	NACA Airfoil Code	"0004"	–
	Component Type	"liftingSurface"	–

Component	**Parameter**	**Value**	**Units**
Fuselage	Length	130	ft
	Component Type	“bluntBody”	–
Fuselage Cross-Sections	Pressure Bulkhead Positions	(14, 135)	ft
	Total Fuselage Length	130	ft
	Number of Superellipses	9	–
	Superellipse Radii	(0.0, 0.0, 0.0, 0.0) (2.5, 2.5, 2.5, 2.5) (7.5, 7.5, 7.5, 7.5) (10.0, 13.0, 10.0, 13.0) (10.0, 20.0, 10.0, 20.0) (10.0, 35.0, 10.0, 35.0) (7.0, 35.0, 7.0, 35.0) (5.0, 23.0, 5.0, 23.0) (1.5, 0.5, 1.5, 0.5)	ft
	Superellipse Powers	(2.0, 2.0, 2.0, 2.0) (2.0, 2.0, 2.0, 2.0) (3.0, 3.0, 2.0, 2.0) (3.7, 3.7, 2.0, 2.0) (4.0, 4.0, 2.0, 2.0) (4.2, 4.2, 2.0, 2.0) (4.2, 4.2, 2.0, 2.0) (4.0, 4.0, 1.9, 1.9) (2.0, 2.0, 2.0, 2.0)	–
	Superellipse Position	(0.0, 0.0, −2.0) (2.5, 0.0, −2.0) (14.0, 0.0, 0.0) (28.0, 0.0, 0.0) (44.0, 0.0, 0.0) (80.0, 0.0, 0.0) (105.0, 0.0, 2.0) (120.0, 0.0, 3.0) (135.0, 0.0, 5.0)	ft
	Superellipse Views	NOTFVIEW FVIEW NOTFVIEW FVIEW NOTFVIEW FVIEW NOTFVIEW NOTFVIEW NOTFVIEW	–

Funding Sources

This work is sponsored by the NASA Aeronautics Research Mission Directorate and Electrified Powertrain Flight Demonstration project, “Development of a Parametrically Driven Electrified Aircraft Design and Optimization Tool”, Glenn Engineering and Research Support Contract (GEARS) Contract No. 80GRC020D0003.

Acknowledgments

The authors would like to thank Ralph Jansen, Amy Chicatelli, Andrew Meade, Karin Bozak, Dennis Rohn, and Gaudy Bezos-O’Connor from NASA’s Electrified Powertrain Flight Demonstration Project for supporting this work and providing valuable technical input and feedback throughout the duration of the project. The authors also thank Maxfield Arnson for his guidance on integrating the visualization feature within FAST.

References

[1] Mokotoff, P., Arnson, M., and Cinar, G., “FAST: A Future Aircraft Sizing Tool for Conventional and Electrified Aircraft Design,” *AIAA SciTech 2025 Forum*, 2025.

[2] National Aeronautics and Space Administration, “Electrified Aircraft Propulsion (EAP),” https://www1.grc.nasa.gov/aeronautics/eap/, 2024.

[3] Gipson, L., “About Electrified Powertrain Flight Demonstration Project,” https://www.nasa.gov/directorates/armd/integrated-aviation-systems-program/armd-iasp-epfd/about-electrified-powertrain-flight-demonstration-project/, 2022.

[4] Jansen, R., Kiris, C. C., Chau, T., Machado, L. M., Duensing, J. C., Mirhashemi, A., Chapman, J., French, B. D., Miller, L., Litt, J. S., et al., “Subsonic single aft engine (SUSAN) transport aircraft concept and trade space exploration,” *AIAA SciTech 2022 Forum*, 2022, p. 2179.

[5] Wang, Y.-C., Stockhausen, M., Mokotoff, P. R., and Cinar, G., “SUbsonic Single Aft eNgine (SUSAN) System Integration Analysis with the Future Aircraft Sizing Tool (FAST),” *AIAA SciTech 2025 Forum*, 2025.

[6] Lukaczyk, T. W., Wendorff, A. D., Colonno, M., Economon, T. D., Alonso, J. J., Orra, T. H., and Ilario, C., “SUAVE: an open-source environment for multi-fidelity conceptual vehicle design,” *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2015, p. 3087.

[7] Stanford Aerospace Design Lab, “SUAVE Documentation,” https://suave.stanford.edu/doxygen/index.html, 2024.

[8] Stanford Aerospace Design Lab, “General Aviation Analysis,” https://suave.stanford.edu/tutorials/general_aviation.html, 2024.

[9] McDonald, R. A., and Gloudemans, J. R., “Open vehicle sketch pad: An open source parametric geometry and analysis tool for conceptual aircraft design,” *AIAA SciTech 2022 Forum*, 2022, p. 0004.

[10] Kruger, M., “The Challenges and Potential Benefits of Electrified Propulsion for Aircraft,” Ph.D. thesis, University of Southern California, 2022.

[11] Chakraborty, I., and Mavris, D. N., “Integrated assessment of aircraft and novel subsystem architectures in early design,” *Journal of Aircraft*, Vol. 54, No. 4, 2017, pp. 1268–1282.

[12] Chakraborty, I., Miller, N. S., and Mishra, A. A., “Sizing and Analysis of a Tilt-Wing Aircraft with All-Electric and Hybrid-Electric Propulsion Systems,” *AIAA SCITECH 2022 Forum*, 2022, p. 1515.

[13] Chakraborty, I., and Mishra, A. A., “Generalized Energy-Based Flight Vehicle Sizing and Performance Analysis Methodology,” *Journal of Aircraft*, Vol. 58, No. 4, 2021, pp. 762–780.

[14] Herbst, S., “Development of an aircraft design environment using an object-oriented data model in MATLAB,” Ph.D. thesis, Technische Universität München, 2018.

[15] Herbst, S., and Hornung, M., “ADDAM: An object oriented data model for an aircraft design environment in MATLAB,” *AIAA Modeling and Simulation Technologies Conference*, 2015, p. 3243.

[16] Trifari, V., De Marco, A., Di Stasio, M., Ruocco, M., Nicolosi, F., Grazioso, G., Ahuja, V., and Hartfield, R., “An aircraft design workflow using the automatic knowledge-based modelling tool JPAD Modeller,” *AIAA AVIATION 2022 Forum*, 2022, p. 3737.

[17] Haimes, R., and Dannenhoffer, J., "The engineering sketch pad: A solid-modeling, feature-based, web-enabled system for building parametric geometry," *21st AIAA Computational Fluid Dynamics Conference*, 2013, p. 3073.

[18] Cinar, G., Garcia, E., and Mavris, D. N., "A framework for electrified propulsion architecture and operation analysis," *Aircraft Engineering and Aerospace Technology*, Vol. 92, No. 5, 2020, pp. 675–684.

[19] Lockheed Martin Corporation, "C-130J Super Hercules," https://www.lockheedmartin.com/content/dam/lockheed-martin/aero/documents/C-130J/MG180389_C-130Brochure_NewPurchase_Final_Web.pdf, 2018.

[20] Eastman, N., Kenneth, E., and Pinkerton, R., "The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel," *NACA-report-460*, 1933.

[21] Arnson, M., Aljaber, R., and Cinar, G., "Predicting Aircraft Design Parameters Using Gaussian Process Regression on Historical Data," *AIAA SciTech 2025 Forum*, 2025.