



Parameterized, Multi-fidelity Aircraft Geometry and Analysis for MDAO Studies using CAPS

Christopher M. Meckstroth¹
University of Dayton Research Institute, Dayton, OH, 45469

The creation of a parameterized, multi-fidelity fighter aircraft geometry for MDAO studies is presented. Using CAPS, the geometry can be directly output to analyses of multiple fidelities ranging from linear panel methods to RANS. The planform is parameterized using 21 total parameters accounting for the shape of wings and tails, fuselage scaling, control surface deflections, and desired output fidelity. The parameterization is centered around the F-16 planform and baseline analysis results are presented and compared to published wind tunnel data. Finally, an MDAO implementation focused on controllability is presented with initial results.

I. Introduction

TRADITIONALLY, aircraft design has been based in large part on previous designs. The initial layout of an aircraft from design textbooks [1] begins with a rough estimate of the aircraft weight, generated from several empirical factors, which are based primarily on previous aircraft designs. With the exception of aerodynamic panel codes for estimation of drag polars, physical analysis is typically reserved for the preliminary design phase, where the Outer Mold Line (OML) is often limited to only a couple of potential configurations. However, according to the National Research Council, aircraft life cycle costs are heavily impacted from decisions made at this point in the design phase and propagate throughout the life of the aircraft [2]. In fact, it is estimated that when the design reaches Milestone A of the DoD acquisition process—when the initial preliminary layout is finished—70-75% of the entire life-cycle cost decisions have been made.

Over the past several years, there has been significant research into aircraft optimization. Particularly within the Air Force Research Laboratory's Multidisciplinary Science and Technology Center (MSTC), where the focus is to reduce the effect of these early design decisions on the life cycle cost by enabling rapid generation and analysis of aircraft configurations with preliminary design levels of fidelity across multiple disciplines. The higher fidelity analysis methods incorporate the physics of the aircraft, allowing the designers, and ultimately, the Multidisciplinary Analysis and Optimization (MDAO) process to experiment with configurations never before considered. One such potential configuration is the Efficient Supersonic Air Vehicle (ESAV). These vehicles will have objectives for supercruise, supersonic dash, maneuverability, and extended range [3] and it is believed that meeting these objectives will lead to tailless airplane configurations [4]. In order for the tailless aircraft to meet the maneuvering requirements of a fighter, it will require the use of various innovative control effectors such as spoilers, clamshells, strakes, or pop-up control effectors placed around the body. With low fidelity tools, these types of effectors can not be accurately modeled. As a result, using current design methods, their consideration is only after a layout has been chosen, limiting their placement to existing free real estate on the aircraft. The MDAO methods being developed will correct this, allowing their consideration concurrently with the aircraft planform layout.

The desired outcome of this effort is to allow future designs to be based primarily on physical analysis early in the design process rather than trusted previous designs, thus allowing exploration of innovative aircraft concepts incorporating new technologies which diverge significantly from traditional planform shapes. With the help of MDAO processes, a multitude of aircraft geometries can be rapidly generated and analyzed early in the conceptual design with a fidelity level typically strictly reserved for the preliminary design phase, where only a couple of 'locked' in geometries are chosen to be analyzed in detail. This gives the designer the ability to base the designs on the latest knowledge of physics rather than solely on previous experience, allowing an exploration of design space combinations never before considered. This paper presents one such MDAO framework incorporating MSTC's Computational Aircraft Prototype Synthesis (CAPS) program for creating and analyzing multi-fidelity, parameterized aircraft

¹ Research Engineer, UDRI/AM, AIAA Member.

geometries. The MDAO incorporates both Athena Vortex Lattice (AVL) [5] for low fidelity analysis and dynamic derivatives and NASA's Cart3D for higher-fidelity analysis.

II. Computational Aircraft Prototype Synthesis

In order to rapidly generate and evaluate aircraft MDAO configurations in the physical realm (e.g. perform real analysis vs. use of human intuition or empirical data), the CAPS program is currently under development [6]. This program is designed to create parameterized CAD-like geometry, specifically for use in MDAO applications. At the core of the CAPS program is the geometry generation tool, Engineering SketchPad (ESP) [7]. This program is easy to use, easy to parameterize, and capable of outputting computational grids and sensitivities to the grids. Although currently primarily used for aircraft OML applications, ESP (and CAPS as a whole) is not designed to be specific to any application, but instead designed generically such that all geometry is created with primitives (similar to a typical CAD program), but with access to underlying geometry definitions that can seamlessly be passed downstream to analysis software. Analysis Interface Modules (AIMs) allow the user to select the desired analysis and format the geometry and/or mesh accordingly. For multi-fidelity aerodynamic analyses, the user can already select AIMs ranging in fidelity from linear panel codes—where the geometry is converted into simple text files—to full Navier Stokes programs. Structural analysis AIMs have been developed as well for both NASTRAN and Abaqus. All required tools for CAPS and ESP are open source and freely available for download (<https://acdl.mit.edu/ESP/>).

Figure 1 shows several example geometries that have been created and/or used in ESP and CAPS. The Porsche Carrera, the turbine engine fan assembly, and the moon lander are geometry-only entities. All aircraft vehicles shown have been incorporated into CAPS with linked aerodynamic and/or structural analysis. Bhagat et. al. [8] used ESP to create a component-based, parameterized aircraft geometry with various control surfaces placed on the body. Using their script, various components were muted (not shown/output) so that the user could dynamically select which components were desired for a particular analysis. The components of concern were wing, fuselage, canard, tails, and inlets. In addition, there were two control effectors on the wing; ailerons and Spoiler-Slot Deflectors (SSDs). Various combinations of these components were analyzed automatically to determine stability derivatives. Bryson [9] performed aeroelastic analysis with one parameter set defining linked structural and aerodynamic models. Heath [10] used CAPS to perform an inlet trade study on a low-boom supersonic aircraft.

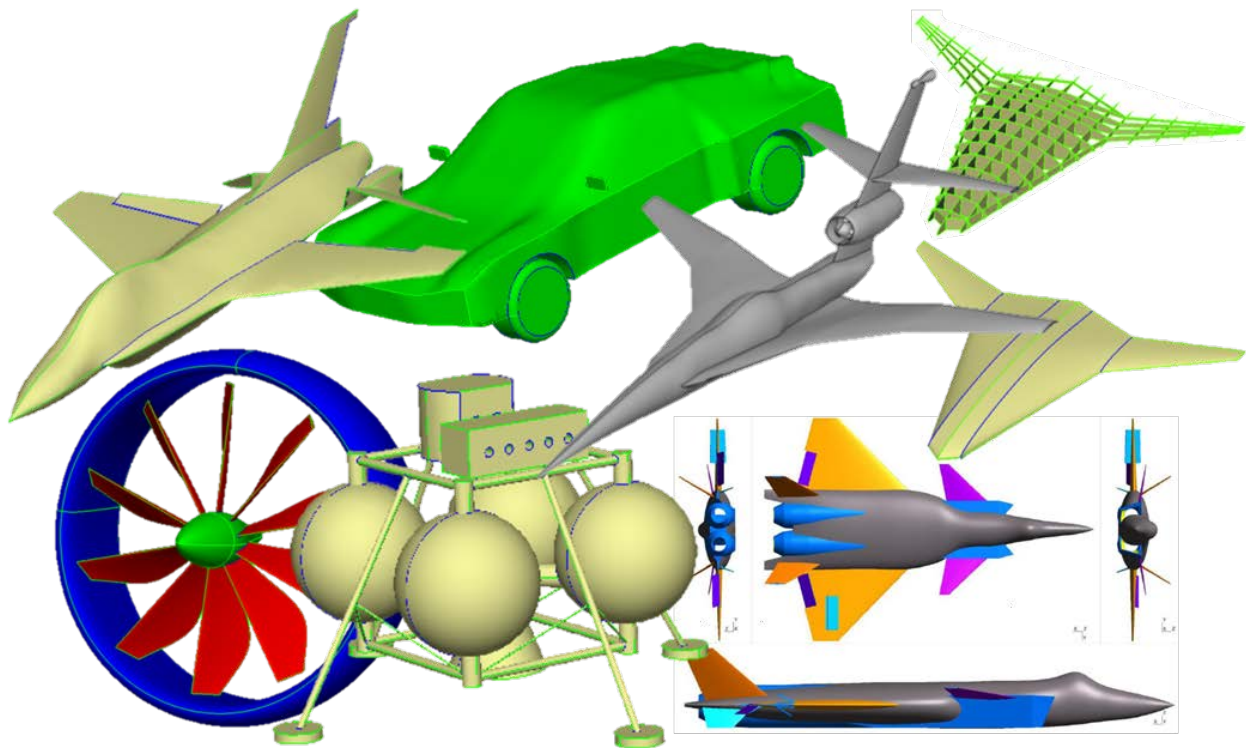


Figure 1. Examples of CAPS Geometries in ESP

III. Multi-fidelity Geometry and Analysis

Although any geometry could be chosen as a baseline for MDAO studies in CAPS, the F-16 was chosen for this project for public availability of geometry, wind tunnel, performance, and even conceptual mass properties data. With this platform as a baseline, nearly any analysis performed can be compared against nominal, publicly available data. Wind tunnel data and basic mass properties published by NASA can be found in Ref. [11] and a reduced form of this data is published in Ref. [12]. In Ref. [13], a lightweight supercruise fighter conceptual layout and sizing example is presented that is based on the F-16, but with assumptions for modern technological improvements. Removing these assumptions, the resulting sizing and mass properties nearly match the other published data.

A. Notional F-16 Geometry

For low-fidelity aerodynamic analysis, the body can often be ignored. However, for medium to high-fidelity analysis, the effects of the body can be significant. For the desired analysis of this work, a good representation of the fuselage was required. Three-dimensional F-16 geometry was created by following an advanced Solidworks tutorial (www.solidworks.com/tutorial) in which a representative (e.g. artistic) F-16 model is created from a series of 2-dimensional drawings, shown in Figure 2. These drawings are aligned with the standard drawing axes and scaled according to known distances. Then, step-by-step, splines are drawn to fit the outline of the aircraft and estimated cross-sectional shapes (Figure 3).

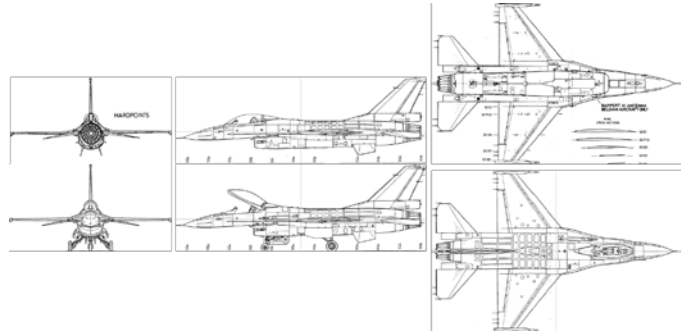


Figure 2. 2-Dimensional F-16 drawings for creating model in Solidworks.



Figure 3. Splines drawn to match 2-D drawings.

The outlines and cross-sections are then used as boundaries for surface generation. Figure 4 shows the result of the top part of the fuselage. During the tutorial, this process is repeated throughout the vehicle for many components with varying levels of detail including antennas, missiles, and even landing gear, until a realistic-looking version of the F-16 is created with significant, albeit artistic, detail. In order to derive an analysis-ready version of this model, much of the detail is removed to create a simple, water-tight model (Figure 5). At this point, the geometry could be exported and meshed for CFD analysis. However, the goal is to create a single model that is capable of being parameterized with rapid geometry changes and analyzed at multiple fidelities. Although not a strict requirement, for this work, the fuselage geometry is simplified again to allow to easier parameterization before inputting into ESP and ultimately CAPS.

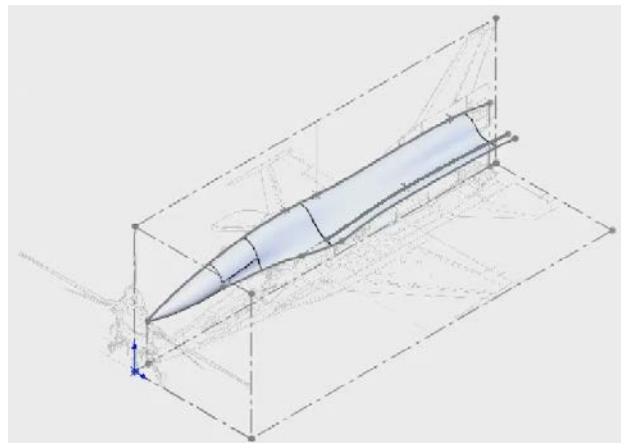


Figure 4. Surfaces created with splines used as boundaries.

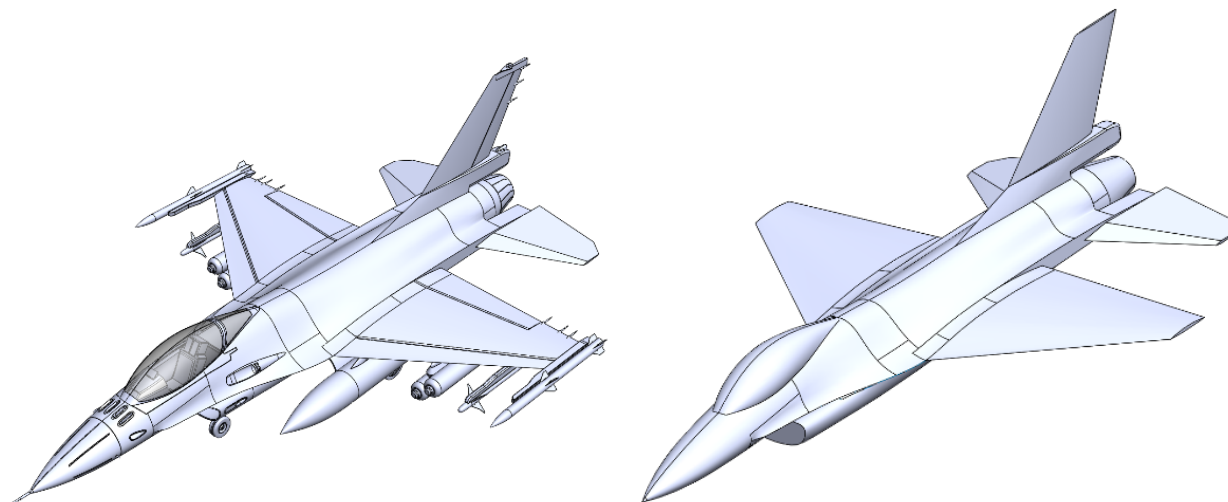


Figure 5. Finished artistic F-16 model from Solidworks tutorial (left) and water-tight analysis model (right).

In order to generate the simplified CAPS geometry, the fuselage is separated into a series of cross-sections. In Solidworks, an intersection curve is created at each desired cross-section, as shown in Figure 6. At each cross-section, the top, bottom and side points are measured, along with the chine location or wing centerline. The area is computed separately for the portion above and below the chine or wing centerline and all of these values are copied into Matlab.

Next, the fuselage cross-sections are simplified by approximating them with Bezier curves. Since the fuselage is symmetric, only two Bezier curves are used for each section which are then mirrored. At the top, bottom, and sides, the slope of the curves are constrained horizontal or vertical to maintain continuity. Although this constraint takes away slightly from the sharp edge of the chines, it helps to enable easier joining to the wing and to simplify meshing for analysis. The parameters for each curve are determined in Matlab by performing a least squares optimization on Bezier curves that match the model at the top, bottom, and side while maintaining the same cross-sectional area. Figure 7 shows the approximated cross-sections plotted in Matlab. These cross-sections are formatted and output directly into the text file input for ESP. Figure 8 shows the example excerpt for one cross-section. Each of the four Bezier curves are defined by four control points. A zero-length line segment, 'linseg' is used to separate the curves rather than using a single Bezier curve for the entire cross-section.

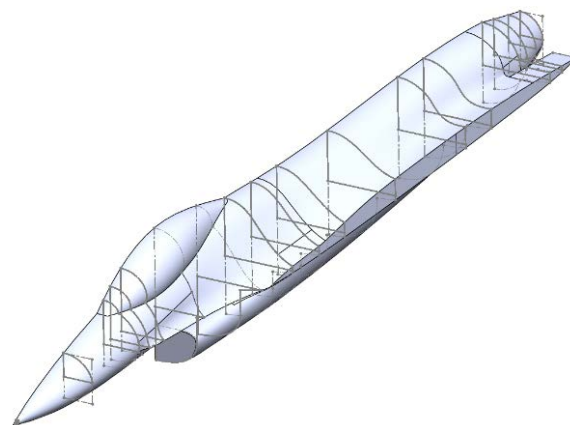


Figure 6. Fuselage cross-sections computed in Solidworks via intersection curves.

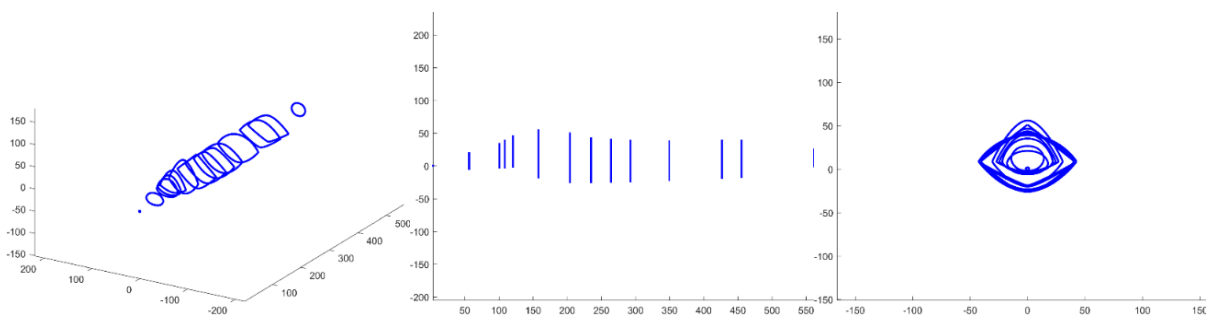


Figure 7. Approximated Bezier cross-sections in Matlab

After inputting all cross-sections, the fuselage is lofted. The wings and tails are created in a similar manner, except their cross-sections are not drawn by the user. Instead, they are created using built in airfoil functions. The flaps and rudder are cut out of the wing and vertical tail, respectively, so they can be rotated. The individual components are then unioned together to create the full CAPS geometry.

For the desired MDAO studies, the geometry is represented in three different ways. For a quick look at the planform, only the wing and tails are created. Otherwise, the geometry is output according to the desired analysis. For low-fidelity vortex lattice analysis (e.g. AVL), only the cross-sections of the lifting surfaces are created. These cross-sections are linked in CAPS which automatically generates the required AVL inputs. For higher fidelity analysis (e.g. Cart3D, Fun3D), the full geometry is generated. Figure 10 shows the output for the planform only and AVL.

```

skbeg      100.787      0.000      35.240
  bezier   100.787      17.818      35.240
  bezier   100.787      22.066       9.858
  bezier   100.787      22.066       8.858
  linseg   100.787      22.066       8.858
  bezier   100.787      22.066       7.858
  bezier   100.787      19.030      -3.970
  bezier   100.787       0.000      -3.970
  linseg   100.787     -0.000      -3.970
  bezier   100.787     -19.030     -3.970
  bezier   100.787     -22.066       7.858
  bezier   100.787     -22.066       8.858
  linseg   100.787     -22.066       8.858
  bezier   100.787     -22.066       9.858
  bezier   100.787     -17.818      35.240
  bezier   100.787     -0.000      35.240
skend
    
```

Figure 8. Example fuselage Bezier cross-section input in ESP

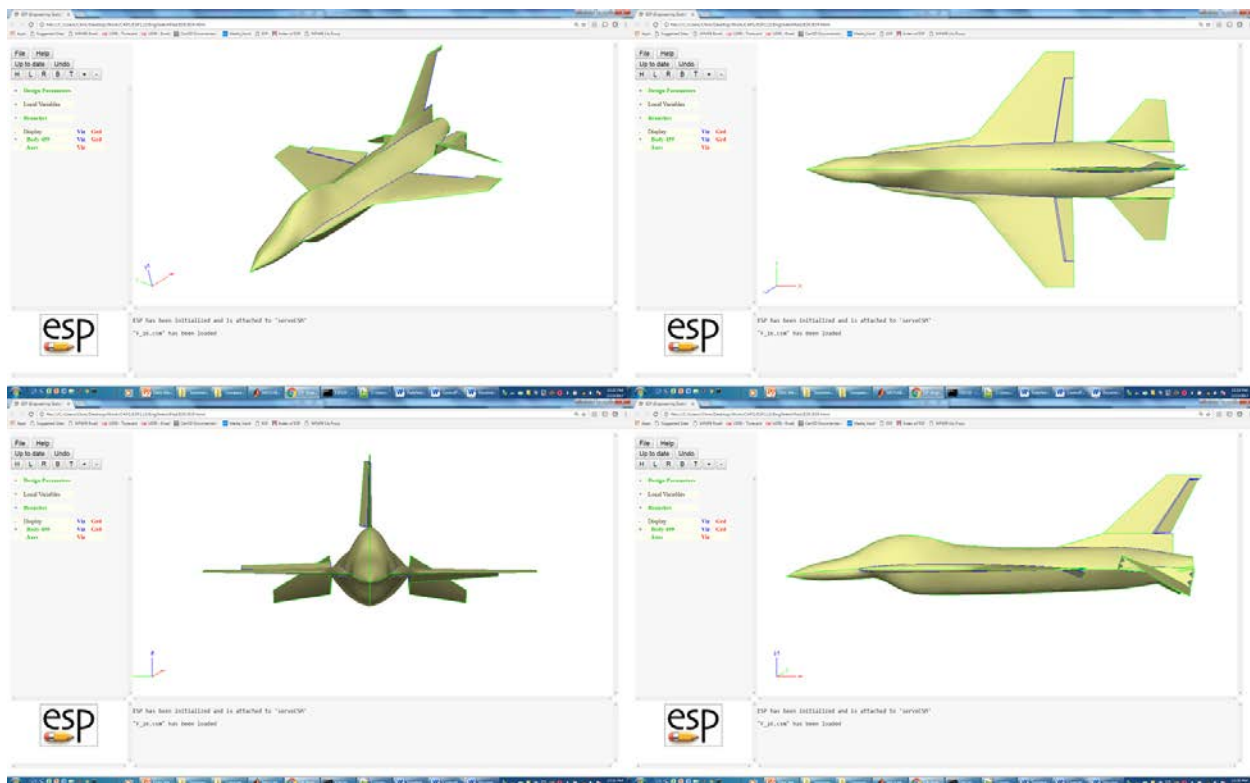


Figure 9. Full F-16 geometry in CAPS

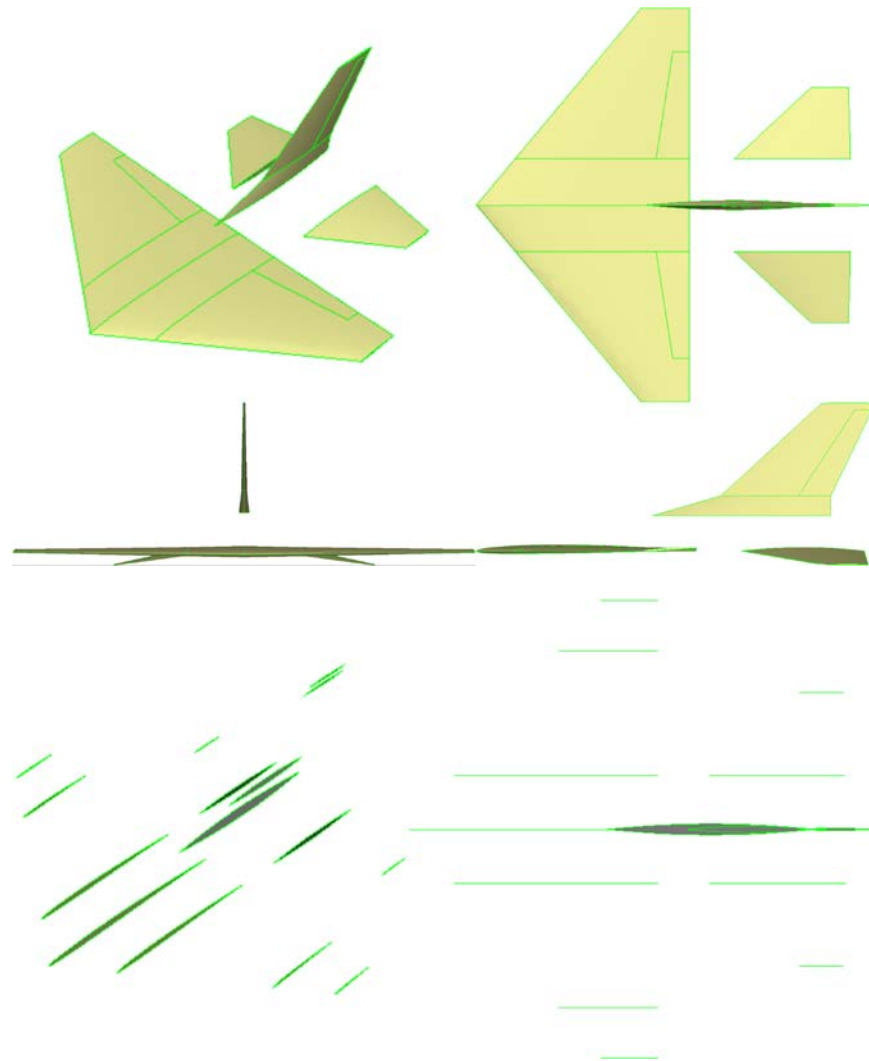


Figure 10. Planform geometry (top) and AVL analysis geometry (bottom).

B. Aerodynamic Analysis Test Results

As mentioned above, there is a significant amount of publicly available aerodynamic data relating to the F-16. With the multi-fidelity geometry in CAPS linked to AVL and Cart3D, an automated sweep of analysis parameters was performed to compare wind tunnel data published in Ref. [12]. This data contains all required parameters to generate a full 6-dof non-linear simulation with control deflections and dynamic derivatives. Controlled through a simple python script, the analysis was performed in both Cart3D and AVL across seven angles of attack (-10 to 20 degrees), five sideslip angles (-10 to 10 degrees), four elevator deflections (-25 to 25 degrees), three aileron deflections (0 to 21.5 degrees), and three rudder deflections (0 to 30 degrees). Figures 11-14 show a comparison of static data between Cart3D and the wind tunnel. This shows a good match in the linear range until 10 degrees angle of attack for all six forces and moments. The discrepancies in the data could stem either from geometric differences in the wind tunnel model and the approximated F-16 drawn in Solidworks or from accuracy of Cart3D. In all cases, Cart3D is able to capture relevant trends, making it a suitable option for MDAO studies. If this level of fidelity is not acceptable in the future, the current geometry and inputs in CAPS can be sent instead to higher fidelity solvers such as SU2 or Fun3D that are already integrated. However, Cart3D is unable to compute required dynamic derivatives (forces and moments with respect to roll, pitch, and yaw rates) that are required for controllability analyses. In order to compute these, AVL is used, and these results are shown in Figure 15. In this case, AVL is much less accurate, but there are presently few other options for collecting these parameters.

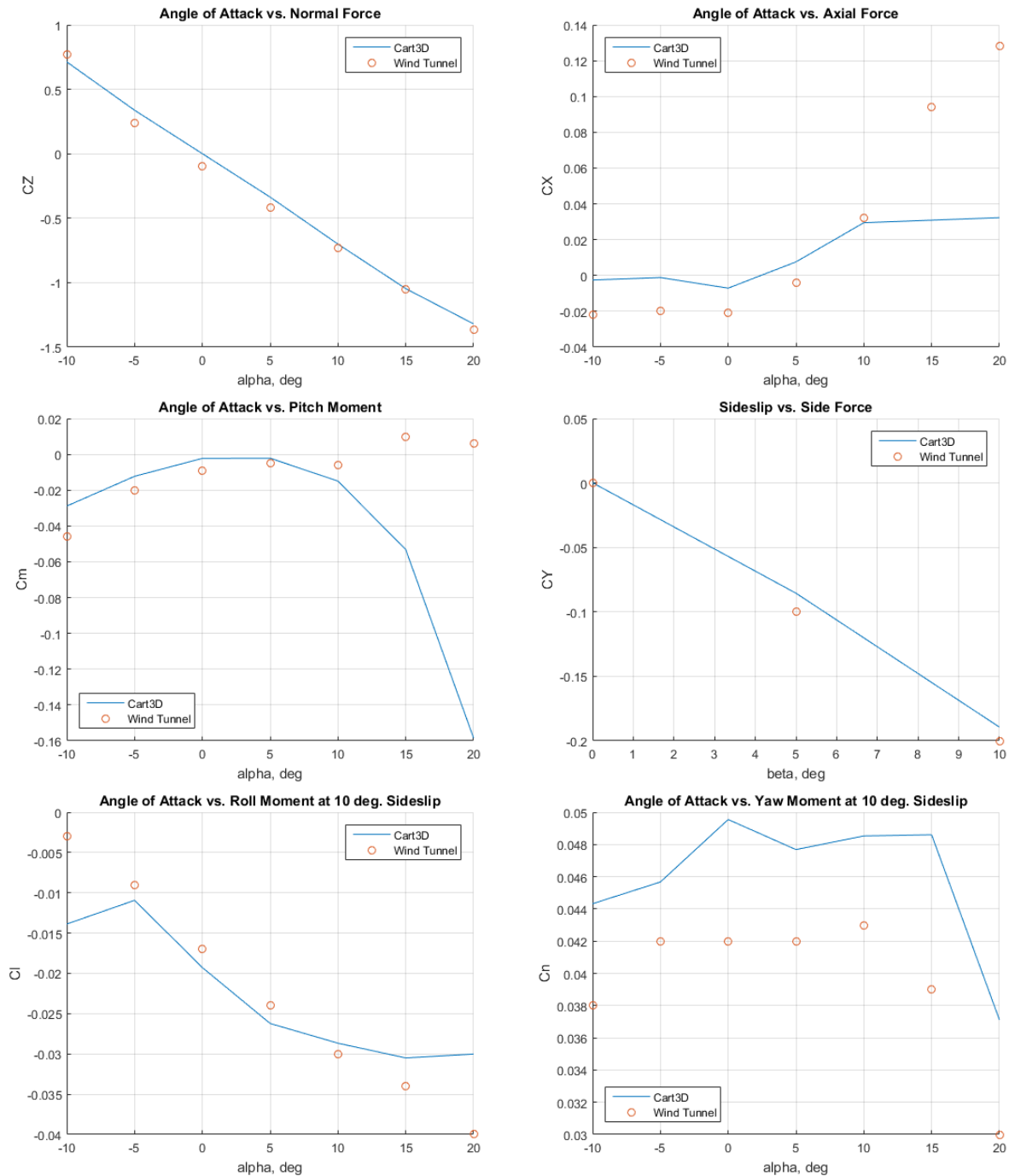


Figure 11. Comparison of static forces between Cart3D and published wind tunnel data.

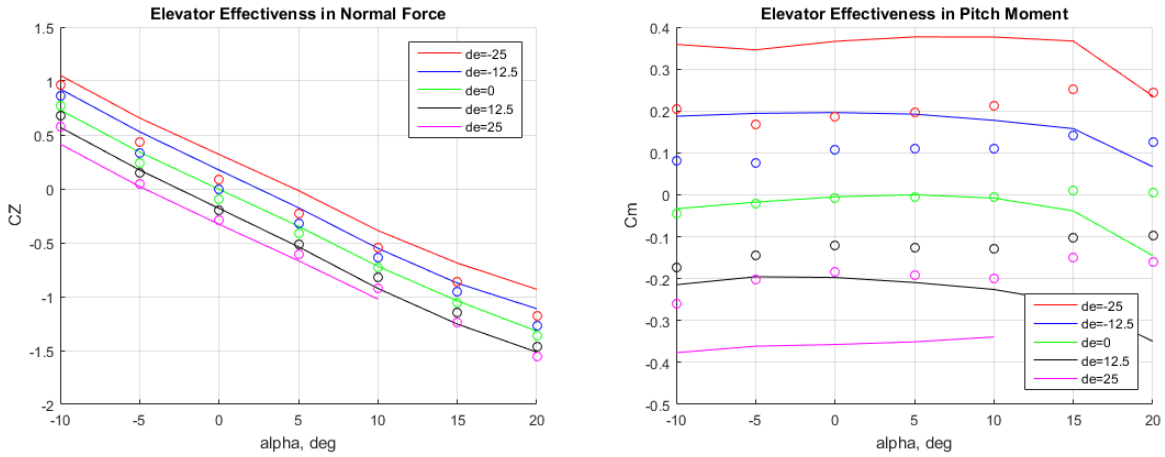


Figure 12. Comparison of elevator effectiveness between Cart3D and published wind tunnel data.

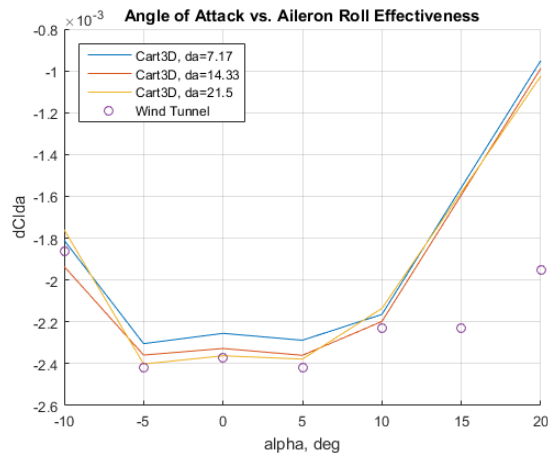


Figure 13. Comparison of aileron effectiveness between Cart3D and published wind tunnel data.

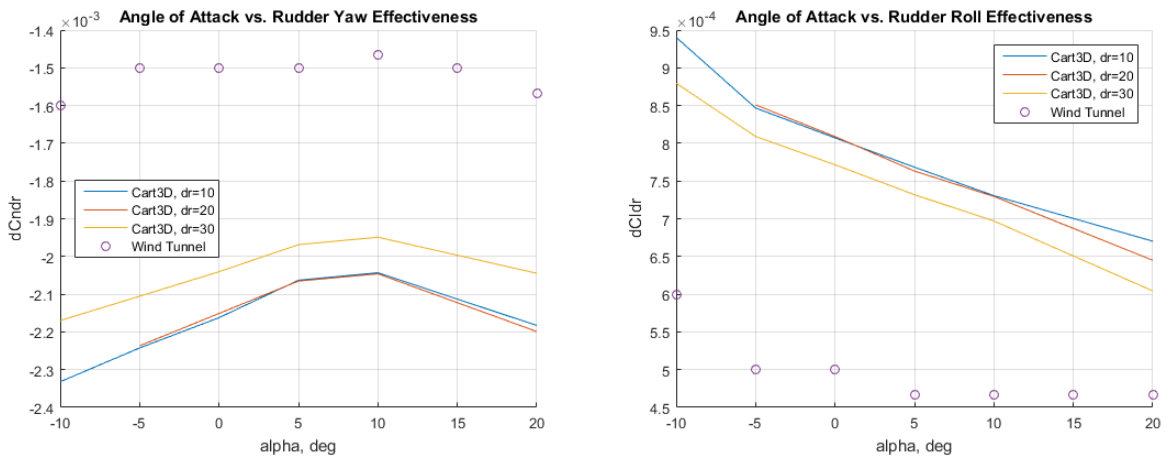


Figure 14. Comparison of rudder effectiveness between Cart3D and published wind tunnel data.

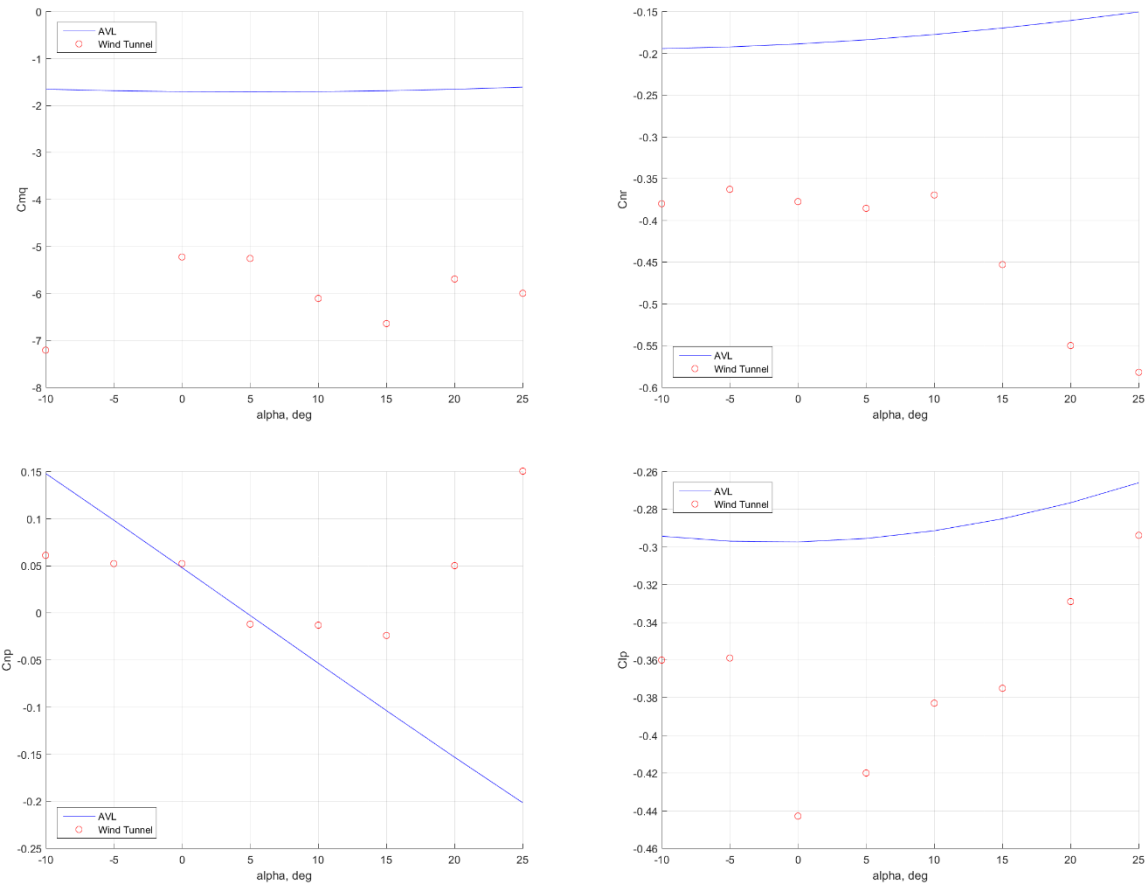


Figure 15. Comparison of required rate derivatives between AVL and published wind tunnel data.

C. Parameterization

In order to perform an optimization, the baseline F-16 geometry was parameterized. In ESP, virtually any dimension used to create the geometry can be incorporated as an accessible parameter. For this project, a total of 21 parameters are used in the geometry creation. The first parameter, *OutType*, determines which of the above representations are to be created. It can be set to '1' for the planform only, '2' for AVL, or '3' for the full geometric representation. Four parameters are used to define control surface deflections; left flap - *deflFlapL*, right flap - *deflFlapR*, horizontal tail/elevon - *deflHT*, and rudder - *deflRud*. The fuselage height and width are scaled using *EngScale*, accounting for growth in the engine through rubber engine conceptual sizing techniques. The remaining 15 parameters define the planforms of the wing, horizontal tail, and vertical tail through leading edge x-location, area, aspect ratio, sweep, and taper ratio. The fuselage cross-sections are linked to these three surfaces so that it shrinks or stretches with their placement. If the wing is moved forward or rearward, the nose sections adjust. The two tails can be moved independently, as the fuselage stretches to match the rearmost tail. Some of these parameter changes are shown in Figure 18.

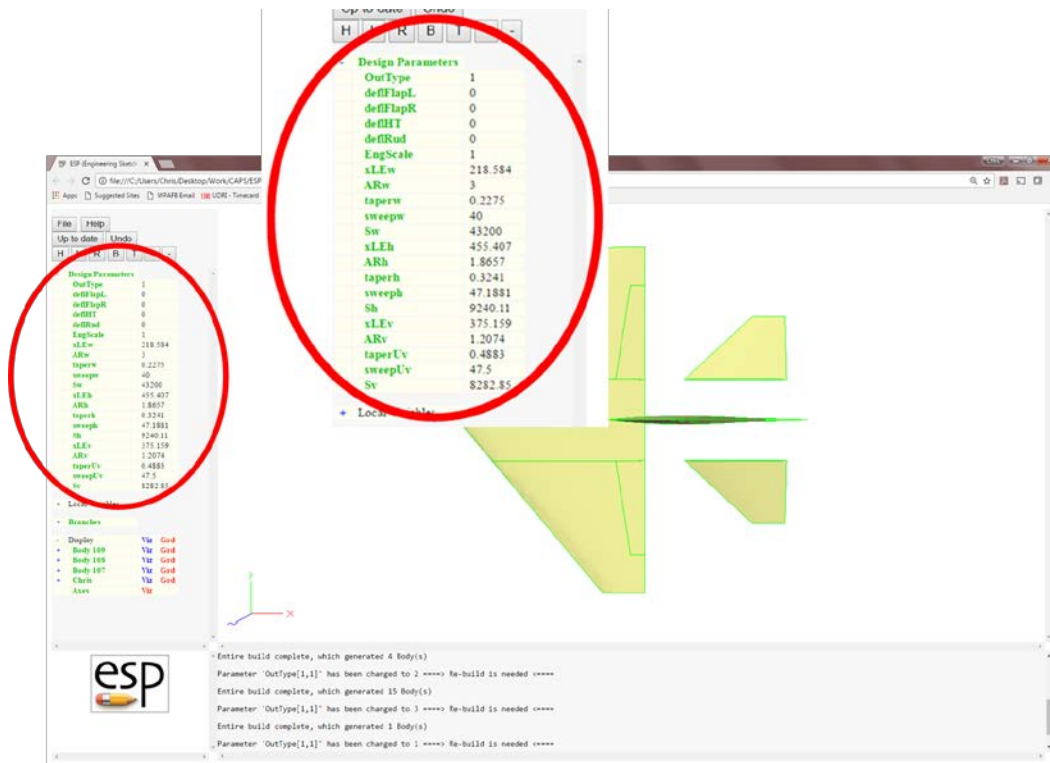


Figure 16. Accessible geometric parameters for baseline geometry in ESP.

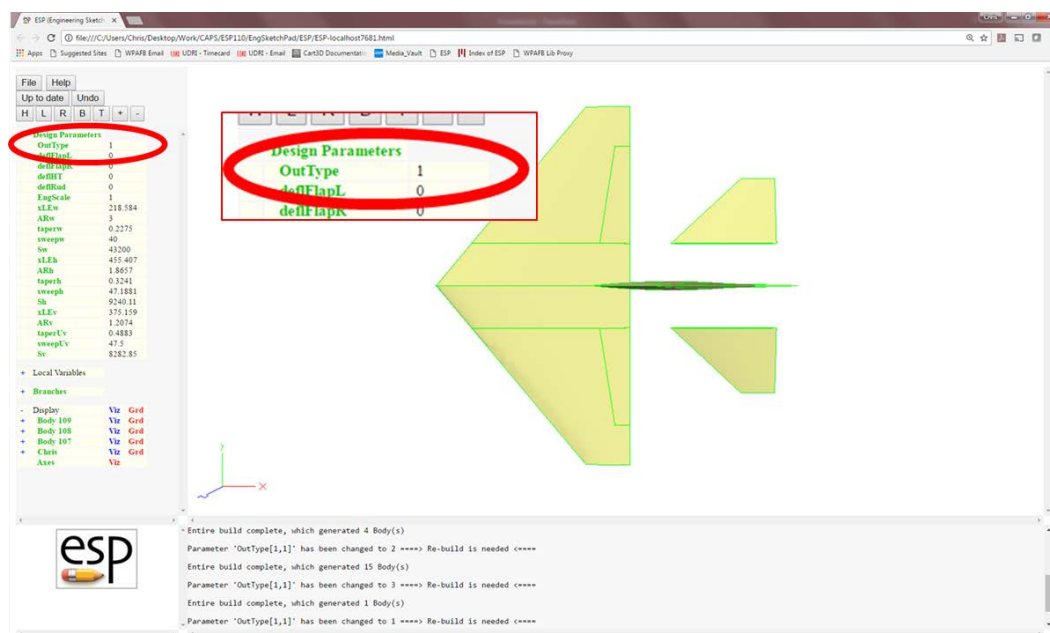


Figure 17. Example of *OutType* parameter in ESP

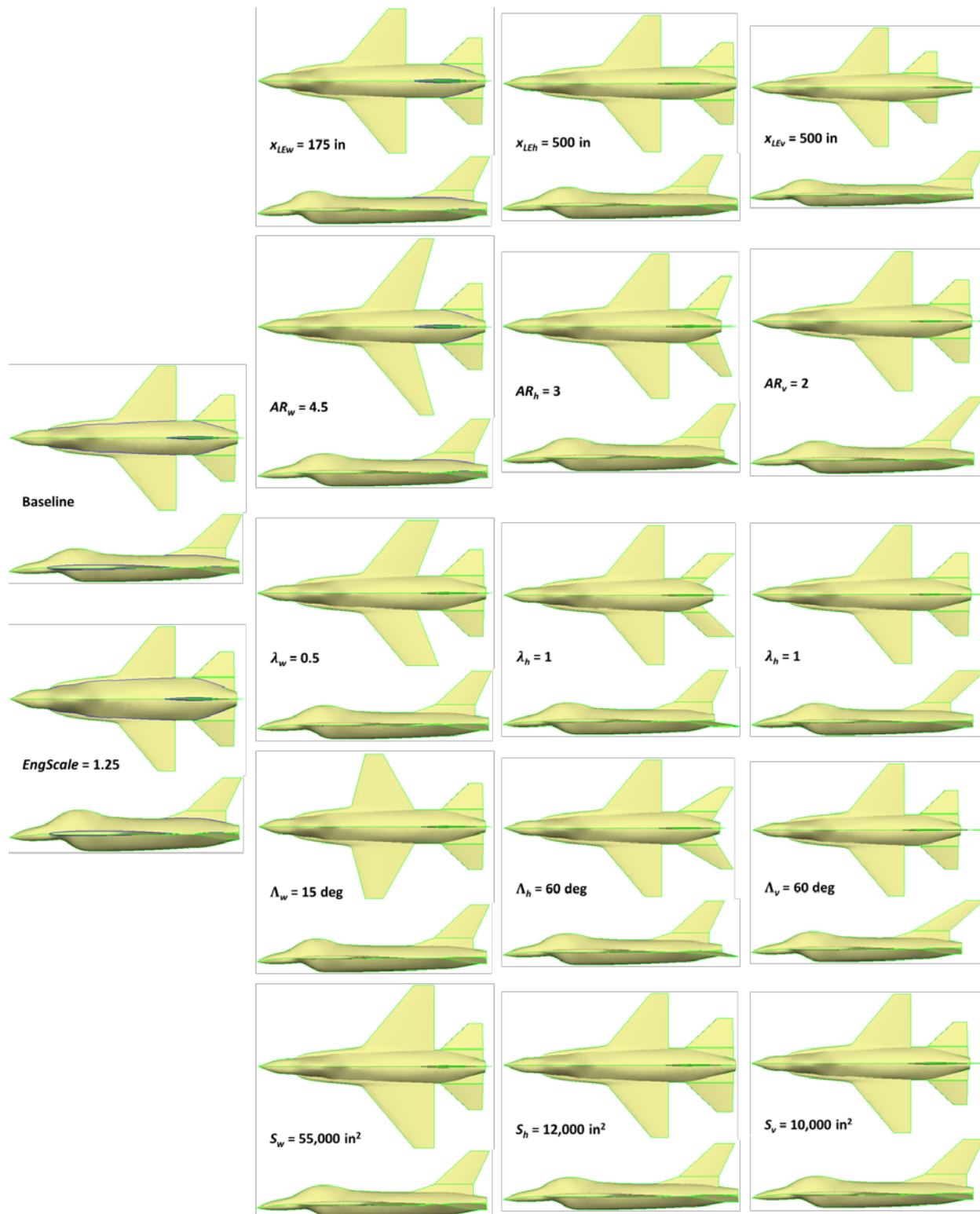


Figure 18. Example geometric parameter changes in ESP.

IV. Preliminary MDAO Results

A. MDAO Process

Throughout MSTC, research is being conducted to incorporate many disciplines into the aircraft MDAO scheme. However, the work presented here focuses primarily on the Controllability Analysis in the loop. Implementation and testing of this portion within an MDAO framework requires a minimum set of additional upstream data and analyses. The components and data flow for this work are described in the N^2 diagram in Figure 19. Each component of the N^2 diagram is implemented in Matlab as part of a modular, object-oriented programming framework. The mission and assumptions block highlighted in green contains inputs to the MDAO process. The remaining blocks highlighted in blue are analyses and/or functions that are performed in the optimization loop. The white blocks above the diagonal show how output data flows downstream to become an inputs to other functions. The white blocks below the diagonal show data that is returned from downstream analysis to become updated function inputs for the next iteration.

Mission and Assumptions	Mission Profile, W/S, T/W, SFC, Initial Planform	Initial Geometric Parameters	Analysis Flight Conditions	Analysis Flight Conditions		
	Mass Properties	Wing Area, Engine Diameter		CG, MOI	Gross Weight	
		Geometry	Multi-fidelity Analysis Geometry			Geometry Generation Pass/Fail?
			Aerodynamics	Static and Dynamic Control Derivatives		
				Controllability	CPR/CPA, Static Margin	
					Cost Function	
	Planform Parameters	Planform Parameters				Optimization

Figure 19. MDAO N^2 diagram.

The MDAO process begins with a problem definition in the form of a desired mission profile, along with a list of assumptions for static inputs. The mission profile is defined in terms of flight phase—takeoff, cruise, climb, etc. with assumptions for the wing loading (W/S), thrust-to-weight ratio (T/W), and engine Specific Fuel Consumption (SFC). Initial planform design parameters are selected to define the rest of the wing and both tails. Conceptual design equations are used to compute fuel fractions for each segment of the mission profile—e.g. the fraction of fuel burn relative to the gross weight during each mission segment [13].

Relating the gross weight back to wing loading and thrust loading, the size of the wing and engine can be determined and a planform layout of the vehicle can be drawn. The components are then located on this layout to compute center of gravity and moments of inertia that are required downstream. The wing area and design parameters are exported to the CAPS program through a Python interface with Matlab. CAPS then uses the design parameters to generate the appropriate geometry in ESP which is then automatically sent to the desired aerodynamic analysis software. After the analysis is performed, the results are returned to Matlab to be analyzed. The controllability analysis is performed and the results are incorporated into the evaluation of the optimization cost function. This cost function is then the subject of the overall optimization algorithm.

B. MDAO Implementation

The MDAO framework described has been developed and prototyped with each section of the N^2 diagram represented. A test case was demonstrated that uses the F-16 fighter plane as a baseline. With this platform as a baseline, nearly any analysis performed can be compared against nominal published data. Wind tunnel data and basic mass properties published by NASA can be found in Ref. [11] and a reduced form of this data is in Ref. [12]. The remaining test case MDAO inputs are described below in more detail.

1. Mass Properties

In the appendix of Reference [13], Raymer uses an example futuristic F-16 to demonstrate the use of the mission fuel fractions approach to determining gross takeoff weight. This example problem uses a mission profile consisting of 13 mission segments; (1) Warmup and Takeoff, (2) Climb, (3) Cruise, (4) Accelerate, (5) Dash, (6) Combat, (7) Weight Drop, (8) Accelerate, (9) Dash, (10) Cruise, (11) Descent, Loiter, (12) Descent for Landing, and (13) Land (Figure 20). For the current example, this mission profile is used for computation of aircraft weights. For each of the

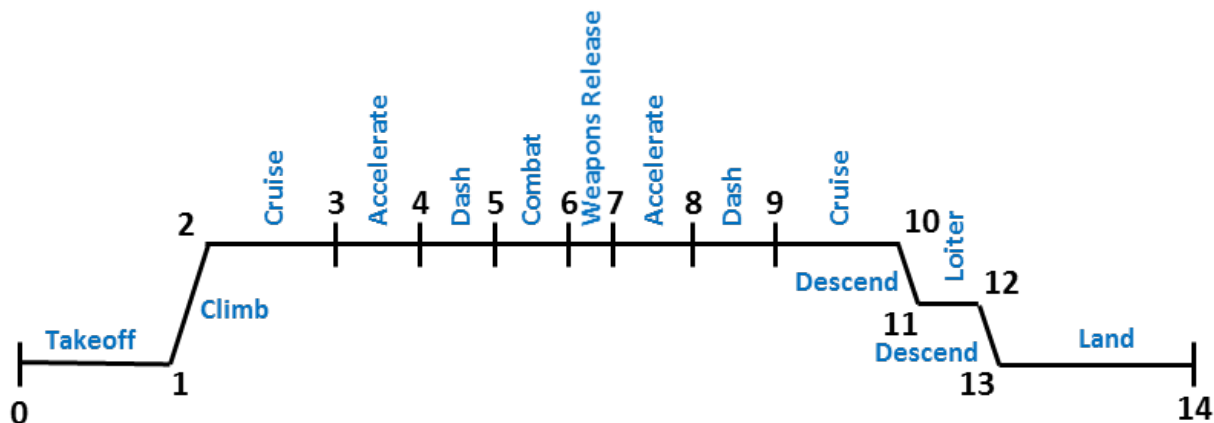


Figure 20. Mission Profile for conceptual weight computation.

mission segments, a conceptual equation is used to compute the fraction of fuel burned in that section relative to the gross weight. For example, equation (1) is used to compute the weight fraction during climb or acceleration in segment i .

$$\frac{W_i}{W_{i-1}} = \exp \left[\frac{-C\Delta h_e}{V(1-D/T)} \right] \quad (1)$$

Multiplying all fractions together for all of the mission flight segments gives the fraction of the empty weight relative to the takeoff gross weight. Using this fraction, an optimization routine is implemented to find the total weight. In this algorithm, a guess is made of the gross weight and then based on the gross weight, empirical equations are used to compute the weight of each component. Examples of the wing and tail weights are shown in equations (2)-(4). The component weights, fuel, and payload are summed to give a computed gross weight. This value is compared to the initial guess and the process is iterated with a new guess until the two match. Once the gross weight is determined in this manner, the each component is dynamically located in 3D space relative to the parameterized geometry to compute center of gravity and moments of inertia.

$$W_{wing} = 0.0103K_{dw}K_{vs}(W_{dg}N_z)^{0.5}S_w^{0.622}A^{0.785}(t/c)_{root}^{-0.4}(1+\lambda)^{0.05}(\cos\Lambda)^{-1.0}S_{csw}^{0.04} \quad (2)$$

$$W_{horizontal\ tail} = 3.316\left(1 + \frac{F_w}{B_h}\right)^{-2.0}\left(\frac{W_{dg}N_z}{1000}\right)^{0.260}S_{ht}^{0.806} \quad (3)$$

$$W_{vertical\ tail} = 0.452K_{rht}\left(1 + \frac{H_t}{H_v}\right)^{0.5}(W_{dg}N_z)^{0.488}S_{vt}^{0.718}M^{0.341}L_t^{-1.0}\left(1 + \frac{S_r}{S_{vt}}\right)^{0.348}A_{vt}^{0.223} * \\ * (1 + \lambda_{vt})^{0.25}(\cos\Lambda_{vt})^{-0.323} \quad (4)$$

Mimicking the mission profile above with the component weights in the loop and with the notional F-16 parameterization, the weight, center of gravity, and inertias that are computed are similar to those in listed in both Raymer's example and Stevens' [12] F-16 data. Figure 21, shows the range of computed gross weights for a Monte Carlo Simulation of 100,000 random input parameters. The red line shows the computed F-16 weight, which is about 3,000 lbs. lower than the mean weight across the design space. This implies consideration for weight reduction in the design process.

2. Controllability

For this work, four flight conditions were selected to be evaluated as likely worst-case control power scenarios. The first case is takeoff, where the pitch moment required to lift the nose wheel off the ground and the yaw moment required to reject a crosswind are often limiting conditions. The second condition is a low-speed maneuver. In this case, the aircraft should be pitched up to a very high angle of attack where the control system is strained just to maintain trim. One of the major real-life considerations in this condition is the ability of the aircraft to recover by pitching down. For this work, the angle of attack is set to only 10 degrees because the accuracy of even the higher-fidelity aerodynamic analysis begins to break down outside the linear region of lift. The third condition is a low-speed cruise condition. In this case, the aircraft is trimmed with straight and level flight, but must be capable of rejecting gusts and performing maneuvers. The fourth and final case is a high speed maneuver. This case would be representative of a dog fight with the aircraft required to pull the maximum 9g pitch up maneuvers. For each test case, the Control Power Required (CPR) or the control moments required to meet these requirements are calculated and compared to the moments the vehicle is capable of producing through control deflections, or the Control Power Available (CPA).

C. Optimization Implementation and Results

A Particle Swarm Optimization (PSO) algorithm was selected for this work due to the difficulty generating gradients. Eleven parameters were selected for optimization. Of the parameters described above, sweep was fixed and neglected and the remaining parameters (wing area and engine scaling) were linked to the conceptual mass properties with wing loading and thrust-to-weight ratios matching conceptual examples. The bounds were chosen partially through known physical limitations (i.e. wing and tail overlap), and typical ranges listed in design textbooks[14]. The algorithm was initiated with 100 particles in the swarm and allowed to perform a total of 50 iterations. Due to current limitations of computation time, the aerodynamic analysis was performed with AVL only. A simple cost function was created that attempted to minimize the ratio of CPA to CPR and gross takeoff weight. With this number of particles and iterations, repeating the algorithm consistently led to similar, but slightly different results. This suggests that the algorithm is not consistently able to find the global minima. More particles or more iterations may be required in the final optimization scheme. However, all results were similar in planform to the F-16. Example output parameters are shown in Table 1, along with the selected bounds and the reference values used to generate the notional F-16 vehicle. In Figure 22, the planform is compared to the F-16.

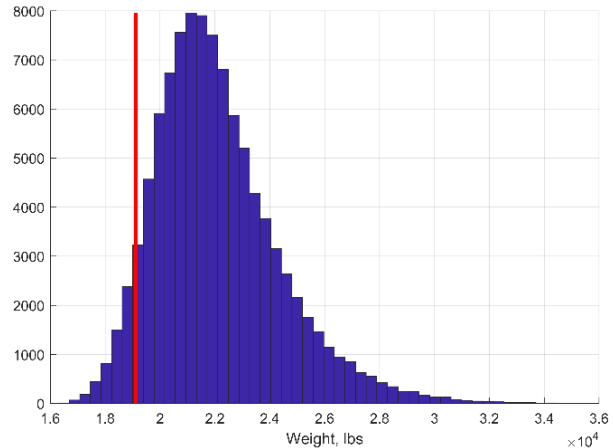


Figure 21. Monte Carlo Simulation results for computed gross weight with 100,000 random parameter variations.

Table 1. MDAO design parameters, bounds, reference values, and optimal values.

#	Parameter	Lower Bound	Reference Value	Upper Bound	Optimal Value
1	Wing Aspect Ratio	2.5	3	5	2.5
2	Wing Taper Ratio	0	0.23	0.75	0.48
3	Wing Leading Edge Location (in)	175	213	300	175
4	Horizontal Tail Area (in ²)	5000	7056	10000	5039
5	Horizontal Tail Aspect Ratio	2.3	2.6	5.8	2.91
6	Horizontal Tail Taper Ratio	0.16	0.3	1.0	1.0
7	Horizontal Tail Leading Edge Location (in)	300	460	1000	393
8	Vertical Tail Area (in ²)	5000	7884	1000	6500
9	Vertical Tail Aspect Ratio	0.4	1.3	2.0	1.40
10	Vertical Tail Taper Ratio	0.19	0.5	0.57	0.39
11	Vertical Tail Leading Edge Location (in)	300	375	1000	300

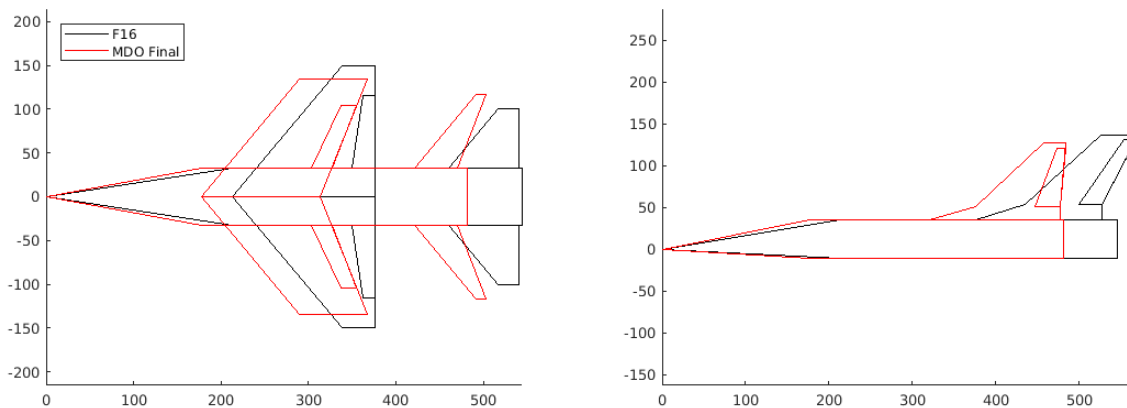


Figure 22. Comparison of baseline F-16 (black) and MDAO optimization result planforms (red).

V. Conclusions and Future Work

A parameterized, multi-fidelity fighter aircraft geometry has been presented, along with analysis results consisting of both AVL and Cart3D data, and initial MDAO results. Although CAPS allows for the automatic generation of both AVL and Cart3D data as shown, the MDAO results presented were generated solely in AVL. As mentioned above, the computation time for using Cart3D in the loop is currently prohibitive. The current analysis used 5,000 function calls evaluating AVL, each of which contained all relevant data. Using Cart3D would require at least 6 times as many runs for finite differencing angle of attack, sideslip, and all three control effectors and still require the same number of AVL function calls to generate dynamic derivatives. For comparison, AVL takes approximately 0.25 seconds and Cart3D takes 45 minutes to perform one analysis run on the computers used. In order to perform the same analysis with Cart3D, it would take approximately 2.6 years to run on the same computer. Moving forward, this work will focus on mitigating this through distributed computing, surrogate modeling, and ultimately limiting assumptions as required.

Acknowledgements

This effort was sponsored by the Air Force Research Laboratory's Multidisciplinary Science and Technology Center (MSTC), under the "Technology Research, Integration, and Demonstration (TRIAD)" Program, Task Order 9: "Design Analysis and Testing of Multi-Disciplinary Airframe Technologies (DAT MAT)" contract to the University of Dayton Research Institute (UDRI). This paper has been approved for public release, case number: 88ABW-2018-5908.

References

- [1] J. Roskam, *Airplane Design Part I: Preliminary Sizing of Airplanes*. Lawrence, KA: Design Analysis Research Corporation, 2005.
- [2] N. R. Council, *Pre-Milestone A and Early-Phase Systems Engineering: A Retrospective Review and Benefits for Future Air Force Systems Acquisition*. The National Academies Press, 2008.
- [3] E. J. Alyanak and R. M. Kolonay, "Efficient supersonic air vehicle structural modeling for conceptual design," in *12th AIAA Aviation Technology, Integration and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012.
- [4] J. A. Bowlus, D. Multhopp, and S. S. Banda, "Challenges and Opportunities in Tailless Aircraft Stability and Control," in *AIAA Guidance, Navigation, and Control Conference*, 1997.
- [5] M. Drela and H. Youngren, *AVL 3.30 User Primer*. 2010.

- [6] E. Alyanak, R. Durscher, and R. Haimes, “Multi-fidelity Geometry-centric Multi-disciplinary Analysis for Design,” in *AIAA Modeling and Simulation Technologies Conference*, 2016.
- [7] R. Haimes and J. F. Dannenhoffer, “The Engineering Sketch Pad: A Solid-Modeling, Feature-Based, Web-Enabled System for Building Parametric Geometry,” in *21st AIAA Computational Fluid Dynamics Conference*, 2013.
- [8] N. Bhagat, D. L. Allison, and E. Alyanak, “Geometry Driven High Fidelity Stability Derivatives Obtained Using Automated CFD Analysis Process,” in *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2015.
- [9] D. E. Bryson, M. P. Rumpfkeil, and R. J. Durscher, “Framework for Multifidelity Aeroelastic Vehicle Design Optimization,” in *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017.
- [10] C. M. Heath, J. W. Slater, and S. K. Rallabhandi, “Inlet Trade Study for a Low-Boom Aircraft Demonstrator,” *Journal of Aircraft*, vol. 54, no. 4, pp. 1283–1293, 2017.
- [11] L. T. Nguyen, M. E. Ogburn, W. P. Gilbert, K. S. Kibler, P. W. Brown, and P. L. Deal, “Simulator Study of Stall/Post-Stall Characteristics of a Fighter Airplane With Relaxed Longitudinal Static Stability,” Langley Research Center, 1979.
- [12] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation, 2nd Edition*. Hoboken, NJ: John Wiley & Sons, Inc., 2003.
- [13] D. P. Raymer, *Aircraft Design: A Conceptual Approach, Fifth Edition*. American Institute of Aeronautics and Astronautics, Inc., 2012.
- [14] J. Roskam, *Airplane Design Part II: Preliminary Configuration Design and Integration of the Propulsion System*. Lawrence, KA: Design Analysis Research Corporation, 2004.