5-1-2019

DESIGN SYNTHESIS, ANALYSIS, AND REFINEMENT OF A THREE-SURFACE NEXT-GENERATION REGIONAL JET

Thomas Arruda

DESIGN SYNTHESIS, ANALYSIS, AND REFINEMENT OF A THREE-SURFACE NEXT-GENERATION REGIONAL JET

by

THOMAS ARRUDA

Presented to the Faculty of the Honors College of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

HONORS BACHELOR OF SCIENCE IN AEROSPACE ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2019

ACKNOWLEDGMENTS

I would like to dedicate this work to my mother and father. Without their continual belief in my abilities and constant support of my continued education, I would not have been able to write this thesis. As the first generation of my family to be born in the U.S. and the first member to complete a college degree, I am forever in their debt for helping me achieve this goal.

I would also like to thank my Senior Design team for taking part in this process. Their openness to my ideas and acceptance of me as their leader allowed me to be successful in both the project and this Honors College endeavor. I would specifically like to thank Carmen Green and David Frew for always making sure that work was completed to the highest standards. I could not have asked for a better team.

Finally, I want to thank all the people in my personal life who loved and supported me along the way. In particular, I would like to thank my longtime girlfriend Lisset Palma for staying by my side since high school and my younger sister Valerie Arruda for keeping me competitive and never letting me be complacent. This was not an easy undertaking and I will always be grateful for the support and guidance they provided me during my undergraduate program.

May 3, 2019

ABSTRACT

DESIGN SYNTHESIS, ANALYSIS, AND REFINEMENT

OF A THREE-SURFACE NEXT-GENERATION

REGIONAL JET

Thomas Arruda, B.S. in Aerospace Engineering

The University of Texas at Arlington, 2019

Faculty Mentor: Dudley E. Smith

The regional jet is an area of fierce competition among the airline industry, where every advantage in terms of operating cost and weight reduction are incredibly important. The study was concerned with possible advantages and disadvantages of a three-surface aircraft when compared to a conventional aircraft. The trade study focused on varying several aspects of the wing planform to find an optimal configuration and served as an introduction to preliminary aircraft design. The focus of this paper is on the individual contribution of the author to their group. The leadership and planning methods used along with industry software design training will be assessed as to their effectiveness. Effectiveness will be measured by the performance of the team according to a panel of industry members as well as through feedback from a mentor, Dr. Dudley Smith, an industry member with over 40 years of experience in the preliminary design field.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Scope of the Thesis

This thesis will outline the individual contributions of the author to a Senior Design project. A Senior Design project is a two-semester long effort consisting of a group project over a single topic as outlined by the mentor, Dr. Dudley Smith. Dr. Smith is a professor of practice at the University of Texas at Arlington as well as a current industry analyst with over 30 years of experience in the preliminary design field. He teaches Aerospace Vehicle Design I and II, which will be used as the basis for the Senior Design project. The author took on a leadership role as both the group leader and the software design lead. The methods used for these items and how these contributions affected the overall performance of the team will serve as the main topics of this thesis

1.2 Senior Design Project Overview

To begin, a breakdown of the overall project will be described. As necessary, a higher level of detail about the project will be given to aide in understanding of the scope of tasks completed. The project being completed in Dr. Smith's Aerospace Vehicle Design is a design synthesis of a next-generational regional jet. A regional jet is defined as a commercial transport aircraft capable of flying 60 to 100 passengers about 1500 nautical miles (nm), or around half the United States [1]. A design synthesis for an aircraft is the buildup of a design from some starting point that ends with an iteration through certain parameters to arrive at a best solution. The class broke into five different groups, and each group was responsible for a separate configuration of regional jet. The author's group was responsible for the three-surface configuration; a picture of the Piaggio P180, one of the most successful three-surface (albeit much smaller than typical regional jet size) aircraft, is shown in Figure 1 for reference. The other four configurations were the conventional layout, the canard layout, the joined-wing layout, and the blended wing-body layout. Pictures of these layouts can be found in Appendix A for further clarification. These groups were meant to take these configurations and attempt to complete the same mission. Once each group designed a plane for this mission, the groups would come together and choose the best plane. This idea of best was defined as the plane which could complete the mission while satisfying all mission requirements as well as maintaining a low weight, a small engine, and a low production cost with high profit margins. A final measure of the groups' efforts was given through a grading of an oral presentation by a panel of industry members and a grading of a written report by Dr. Smith at the end of each semester.

Figure 1.1: Piaggio P180 in Flight [2]

1.2.1 Project Focus – Semester 2

The project focused heavily on the creation of a Synthesis program to automate the design process. This program would take a set of inputs with initial geometry data, vary certain geometrical aspects of the main wing, conduct a mission performance analysis for each plane with the different geometrical parameter, and then plot the data on a carpet plot. This Synthesis program should have incorporated stepwise refinements of methods used during the first semester of work. An example of this would be the drag buildup model. In order to calculate the performance of a plane, certain aerodynamic quantities such as the drag must be calculated. This is done through a buildup of the drag polars, or characteristic equations which model the drag coefficient for the plane during a certain phase of flight. During semester two, the team created control surfaces such as ailerons and flaps which were then incorporated into this drag polar buildup, offering a more accurate or refined model.

The program was necessary during the second semester because of the overall goal: to conduct trade studies by varying certain aspects of the main wing and analyzing the data to see which combination of alterations led to the best design. The varying of certain aspects led to over 500 unique planes being investigated. It would be impossible for a team of six undergraduate level students to complete the design of 500 different planes by hand while maintaining accuracy and consistency. For this reason, a computer program was developed to complete the design process and analysis.

CHAPTER 2

COMBINING LEADERSHIP AND SOFTWARE DEVELOPMENT

2.1 Honors Thesis Definition

In order to complete the group project option for fulfillment of Honors College requirements, an extra and individual contribution or responsibility must be taken while also completing the normal Senior Design course. The author took the role of team lead as well as the responsibility of software design lead. Seeing as the software portion of the normal class was so important, the items of group leadership and software development were combined. This was accomplished through an implementation of the Agile Software Development process.

2.2 Defining Agile Software Development

Agile Software Development is an umbrella term to classify a method of thinking and planning for creating software products. The Agile manifesto, in its entirety, states "We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value: individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan. While there is value in the items on the right, we value the items on the left more [3]." The first key value speaks to face-to-face interactions and their importance. During the software development process, constant interaction among the team as well as between producer and consumer are very highly valued. The second value outlines the need for a working product at all times. The team should always have a working copy of the product available, and this product is continuously upgraded in order to increase functionality. The third value refers to how the customer should be involved in the design process rather create a set of needs and sit back waiting for a product. The last value essentially says that should the customer change a requirement or add a desired functionality, the team should be able to respond to this and not take issue with deviating from a plan.

There are also 12 principles of Agile Software Development that are shown in Table 1. These principles are slightly more specific to delivering a software package but are still generalized applications of the key values present in the Agile manifesto.

Table 2.1: Agile Development Principles [4]

Number	Principle Statement	Number	Principle Statement
1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	7	Working software is the primary measure of progress.
2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9	Continuous attention to technical excellence and good design enhances agility.
4	Business people and developers must work together daily throughout the project.	10	Simplicity--the art of maximizing the amount of work not done--is essential
5	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	11	The best architectures, requirements, and designs emerge from self-organizing teams.
6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

2.3 Applying Agile Principles to the Senior Design Project

The Agile design approach is an intensive process and not all principles applied to this specific project. For example, Principal 1 talks about delivery of software early and often. For the project, it was important to have a program that could output certain parameters when requested by the customer, but the customer never received a software package to run themselves. This goes in hand with Principal 3 which says to deliver software frequently.

This section will attempt to describe the use of some Agile principles as they apply to this specific project. Principle 1 does not apply to this project through software delivery. Satisfaction of the customer is achieved through delivery of the results output by the software being developed. Principle 2 can apply directly to the project; the main customer, Dr. Smith, may ask for something new at a point late in the semester. These are new requirements which could show up on the grading rubric that the panel of industry members possess when reviewing each groups' presentation at the end of the semester. If this group can provide the new requirements, then we have gained a competitive advantage. Principle 5 is difficult to directly relate to this project. Dr. Smith would be the assumed businessperson, but he has a thorough understanding of the processes that each team is going through and also has many years of experience in this field, so explaining to the business team how their goals are being accomplished is not common. Daily contact is rare due to the strenuous schedules of Dr. Smith and each team member. Setting up online meetings was not an option that was pursued, but historically has been one way to overcome this hurdle. Principals 6, 11 and 12 were utilized through face-to-face team meetings hosted every Monday and Wednesday. This will be expanded upon further later in the thesis.

In order to apply these Agile Development principles and key values, leadership that can hold the team accountable is required. The author filled this leadership role and was the sole organizer of team meetings and customer updates, ensuring compliance with these principles, and educating or helping the team understand the process. The main way that these tasks were completed was through short meetings or scrums hosted every Monday and Wednesday, where team progress was tracked and updates on task progress were given by team members.

CHAPTER 3

REAL APPLICATIONS OF AGILE SOFTWARE DEVELOPMENT IDEALS

The application of this software development methodology will be the main focus of this thesis. This was accomplished through four channels: incremental, short bursts of focused work, efficient communication between the consumer and development team, constant, meaningful contact among the team, and decentralizing work with constant integration.

3.1 Sprints

Agile development emphasizes quick turn around and delivery of a finished product. To help ensure this, a main feature is to pick a few tasks to work on from a larger overall list and finish those tasks. This helps the overall project progress by adding in completed functionalities little by little. A common problem among any large ongoing project is to start a new task, get close to finishing, then start a new task and repeat. Those original tasks never get fully completed and implemented because the team is always trying to work on brand new issues. This is an easy trap to fall into when the customer continuously piles on new tasks or functions to add because starting on these new items shows ambition and eagerness to please the customer. However, if these tasks are never fully finished and implemented, the program is left a jumbled and discontinuous mess with little or no meaning for the customer. Ensuring that a select few tasks are completely worked through and integrated corresponds to consistent, stepwise improvements of the base-line project. When this method of working a task to completion is combined with a timeframe, the idea of a sprint is created. A sprint gets the name from its typically short timeline and the rushed feeling of trying to complete the set of tasks; sprints can last anywhere from one week to one month depending on the workload. Each team member is assigned a handful of items or fewer to complete, submit for peer review, and integrate into the main branch of the program. These tasks are expected to be fully reviewed and integrated by the end of the timeframe to arrive at a more complete and more robust program.

3.2 Task Creation and Storage

In order to organize a sprint, tasks need to be chosen. These tasks originate from either errors found in the program or from new requests from the customer. When these items are not being worked on, they are referred to as issues. These issues were kept track of using a project management program known as Trello. Trello is an industry tested website that allows for projects to be divided and organized in a Kanban style board. A Kanban board is a visual analysis tool that showcases what work is currently being worked on, helps to minimize work in progress, an increases efficiency [4]. An example of the overall view of the Kanban-style Trello board for this project is shown in Fig. 3.1.

3.2.1 Explanation of Trello Board Lists

When an issue is found, it should be added to some sort of queue. Once on the queue, this specific team's application of the Agile process can deviate from other established methodologies. After being added to the queue, the item was immediately assigned an importance category. These categories would aide in task assigning during the scrums. There were three groups of lists on the board: tasks being worked on, issues to be worked on, and the modules to which the tasks belonged. Issues would start their lifecycle on the issues to be worked on list, shown in Fig 3.1 as “Code Maintenance/New Features.” Once an issue was assigned, it would move to the “TO DO IMMEDIATELY” list. This list would constantly update with assignees, due dates, and tasks moving onto and off the list. Once a task was complete, it would move from the to-do list into the specific module that it belonged to. This allowed for the progress of a module to be tracked over the course of its existence.

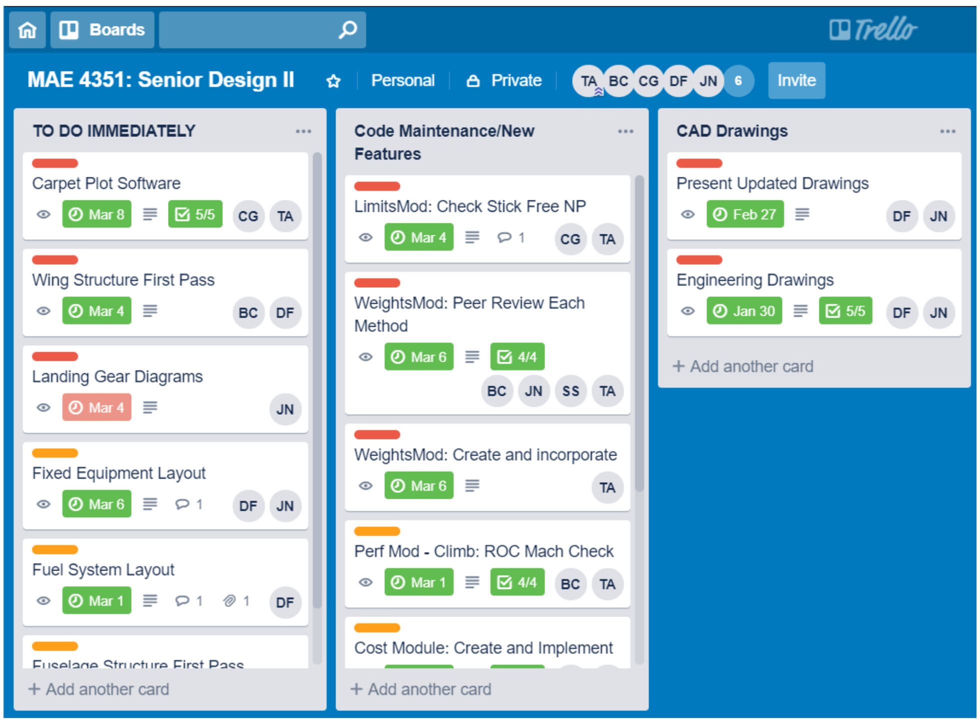


Figure 3.1: Kanban-Style Trello Board

3.2.2 Explanation of Tasks and Issue Lifecycle

A further explanation of the lifecycle of a task is necessary to understand the full process involved in completing an issue. An issue begins from either an error in the program found through user or development team use, or from a new feature being suggested by either the team or customer. The issue would be given a title and brief description during its inception. During a scrum, the team would converse on an issue in the queue and assign it both an importance level and timeframe. The timeframe was simply an estimate of how long it might take to accomplish the task. During a sprint planning meeting, an issue may be moved from the queue onto the in-progress list. Once here, a team member would be assigned. This team member was responsible for developing a plan of action for the task and producing a list that they could check off as progress was made. These lists were the main way of tracking progress on an issue not during a scrum. An example of a completed task card is shown in Figure 3.2. The last steps on the list should always be peer-review submission and finally integration into the baseline. Once these final items were done, the task was marked as complete on the due date and moved into its specific module, with more recent tasks on top.

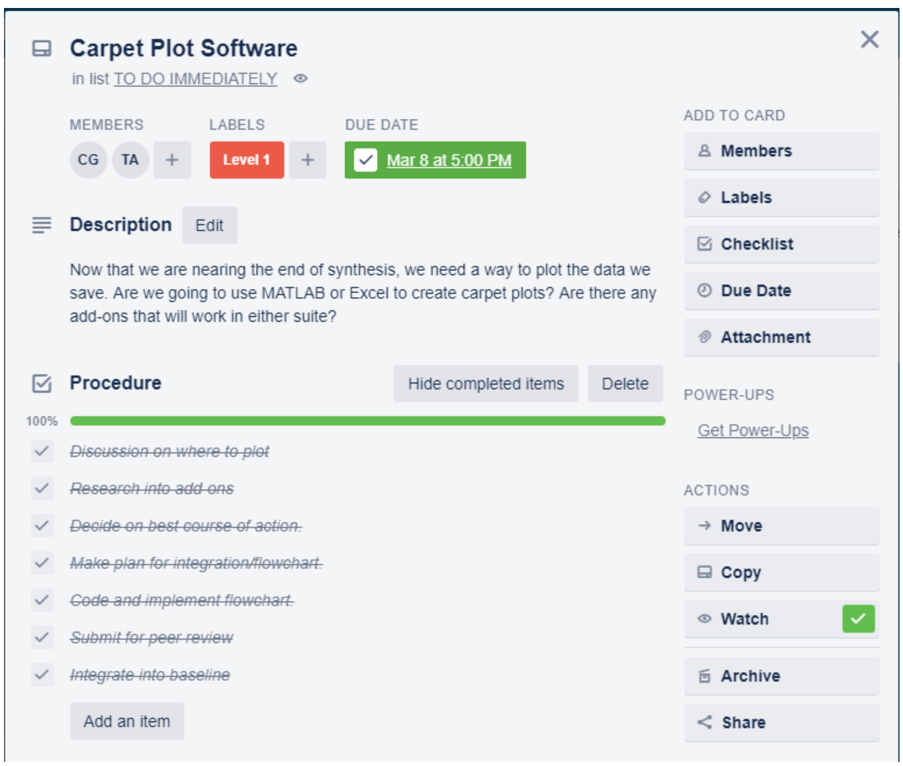


Figure 3.2: Completed Task Card

3.3 Team Meetings: Scrums and Sprint Planning

In keeping with the Agile Manifesto, team meetings were almost exclusively conducted in person and each meeting had a defined outline. Each meeting had a goal to complete and participation from each member was expected. The team leader also made these meetings an open forum for discussion so that all ideas were heard and given consideration.

Scrums are shorter meetings that took place each Monday and Wednesday after the normal meeting time of the class. These usually lasted no more than 15-20 minutes and were used for progress updates on each members' tasks, an update on the overall sprint progress, and a look into where the team stood in relation to the end goal of the semester. During scrums, the main goal was to understand how each member was performing. If a member was falling behind on a task or had an issue completing a task, help was quickly offered to try and help them catch up. This goes back to the idea of an open forum; team members were encouraged to talk about issues they were having or if they were falling behind. This allowed the group and leader a better understanding of whether the sprint would be completed in time. The entire group is working towards the same goal, so if one member is falling behind it behooves the rest of the group to help them catch up and fix the shortcoming.

Sprint planning meetings occurred during the laboratory time for the class. The main goal of these meetings was to organize a new sprint to start work on. The sprint was typically a one and a half week long action plan where the team worked to complete certain tasks for the project. During the meeting, tasks in the to-do queue were assessed. The issues were assessed in terms of age and importance level. As stated, each issue was assigned an importance. There were four unique values to choose from, ranging from Level 3 to Roadblock. These items and explanations are given in Table 3.1. Items of higher importance were looked at first. Any new roadblocks were usually assigned to the new sprint first, and then the level 1 issues being blocked would follow. Next, older level 1 items were investigated. If the item was still pertinent, it would be added to the in-progress list to be worked on. After this, items at level 1 or older level 2 items were usually just filled until each member had two to four tasks to work on.

Table 3.1: Issue Importance Levels

Level	Color	Meaning
ROADBLOCK	Blue	This item is blocking other issues. It must be done before these issues can continue.
Level 1	Red	This item is critical to proper function of the program or customer satisfaction.
Level 2	Yellow	This item does not stop the program or inhibit the customer but would be a welcome addition.
Level 3	Green	This item was not asked for by the customer and is only a quality of life change for the program.

Once each member was assigned their tasks, they went back and created the defining list outlining the task's approach to completion. Then, the time estimate was adjusted based on if the level of effort required was reflected in the original value. Once this process was done for each task, the group reconvened to establish the sprint end date. This date represented a firm date by which all work should be peer reviewed and integrated. This date was semi-flexible based on the needs of the overall team. The first few iterations of this process, the team would underestimate the amount of time required to complete tasks and the deadline would be pushed back a week. After a few tries, the group became more comfortable with estimating workload and this issue was not as common.

However, these sprints were sometimes extended due to workload of the team for outside classes. Each member was also a full-time student and working part-time. Typically, most classes would follow a similar exam schedule, so the team would all have exams for other classes during the same week. This would often cause the sprint to be delayed if it was not planned for. While this usually did not present an issue, the lost time became apparent at the end of the semester as tasks in the to-do queue piled up and time estimations were ending after the last class.

Also, important to note about these sprints is that they are a flexible notion. Principal 2 of Agile Development talks about expecting and welcoming changes late in development. If the customer required a new item and expected the item to be completed quickly, the task was immediately added to the sprint and a large portion of the team was committed to working on the task. This eagerness to adapt to changing requirements that is stressed in the Agile Development process allows for simpler and less painful integration of these new requirements.

3.4 Constant Customer Contact

One of the ways the team tried to employ Agile Development was through constant contact with the consumer and including them in the program development process. This was accomplished through meetings between the team lead and Dr. Smith or other group leads. During regular class time, Dr. Smith would usually provide guidance on where the project should be going and where in the process the group should likely be. Dr. Smith would often call on the class (specifically the team leads) to present their findings and give a progress update for the team. Since this team was usually on top of tasks thanks to the prior implementations of the Agile process as well as a craving for this constant contact for both keeping the customer in the loop and learning about new requirements or shortcomings in the project, the author almost always volunteered to present first.

The importance of this eagerness to volunteer is an asset that was often overlooked and regretted by the other teams. The regular class periods were only 50 minutes long, and presentation of the findings often took upwards of 20-35 minutes. Because this team was always first to go, and was usually ahead of the curve, extensive feedback from the customer was acquired. Dr. Smith would often present new ideas or methods to accomplish the work presented or find errors that the team overlooked. His experience with the design process allowed him to find mistakes quickly because of his feel for correct values in various design aspects. This intuition and his feedback were invaluable because without it, the team would likely have never known the values presented were an issue. It also offered a time for other groups to review our work during class time and point out errors or new ways of looking at the data that otherwise would not have been found as quickly if at all. By going first, the team got the bulk of the class time devoted to new forms of peer review and other groups did not receive the same level of attention or time, if they got to present their findings at all.

3.5 Decentralization of Work and Integration

The adage "divide and conquer" is still popular today for good reason. As seen from Section 3.3 over planning, the team divided larger workloads into small, independent chunks. This falls in line with Principal 5 of providing an environment for work to get done independently and trusting that they will provide quality work. The sheer volume of work to be completed does not lend itself to individual completion of the work, or even having one individual with oversight of the entirety. If an individual were relied upon for completion and approval of every task, things would move incredibly slowly, and that trap of many things started with very little finished would likely arise.

To avoid this issue, the group decentralized the workload among each individual, and peer reviews were relied upon to ensure quality work. By splitting up the work and forcing these tasks to be completed, steady progress was achieved. The peer review process usually involved sitting down during a class laboratory period and marching through the work with a fine-toothed comb. Otherwise, the work was uploaded to a central server so each peer reviewer could access the work and check it remotely. This meant that no single member was responsible for oversight of the entire project, and through group alignment for a common goal, quality work was always submitted anyway. Up to this point in the Senior Design process, there have not been many errors that made their way into the code for more than one sprint, and the quality of work has never been an issue when submitted.

CHAPTER 4

SENIOR DESIGN PROJECT RESULTS AND REFLECTION

The results of the senior design project and the apparent effects of the leadership approach taken are a direct measure of the success of the group. This success or failure, when compared with other groups the author has led and with the other groups of the Senior Design class, is a strong indicator of whether this extra work involved in Agile Software Development was worth the extra effort.

4.1 Senior Design Results: Semester 1

During the first semester of the project, the author's leadership methods and use of Agile Software Development were not as well defined. Although the practices were in place, not all methods or principles of the Agile system were followed as closely. Scrums were not as often and did not have the level of communication or honesty needed for successful communication about the project timeline. However, there was more time being devoted to the cause as most of the members had a smaller workload, so sufficient progress was still being made. Also, some of the main ideas such as decentralization of work and constant contact with the customer were still being widely used.

4.1.1 Semester 1 Work Output

During Semester 1, the design process was conducted at a level one detail level for preliminary design. This process closely followed the work of Dr. Jan Roskam's Design Series, Volume 1 [5]. The process started with the whole class working towards a similar goal because this item was not dependent on specific layouts and instead assumed a conventional layout for every item. Through a mission definition from historical research of planes with a similar mission, a performance analysis could be attempted. This mission definition is shown in Figure 4.1. This began first with finding weight estimates for the plane using historical research as well as historical statistical data in the form of linear regressions based on aircraft empty weights.

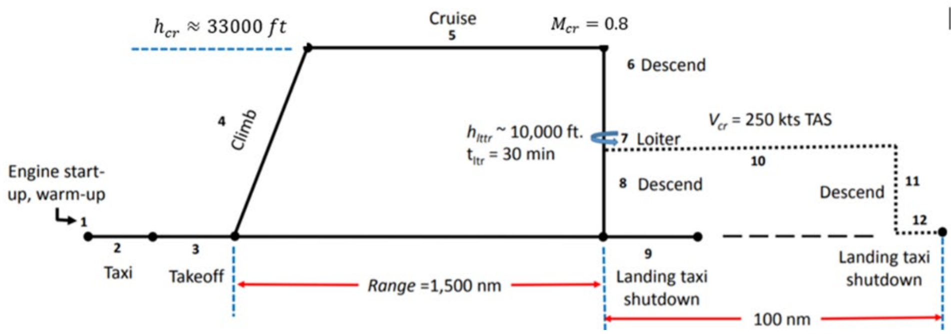


Figure 4.1: Flight Mission Definition

Next, a generalized constraint analysis was conducted to define a design space. This process involved looking into FAR 25 requirements (a set of requirements developed to determine airworthy turbojet powered aircraft) and trying to satisfy them through wing loading equations or thrust-to-weight ratio equations. By satisfying all of the requirements and plotting them against each other, a design space was created. Shown in Figure 4.1 is the preliminary design space the group arrived at. Because the aircraft was said to be next-generational, the design space was then pushed to the right into the darker blue area. Typically, the higher the wing loading and the lower the thrust-to-weight ratio, the lower the weight of the aircraft, so the green dot on Figure 4.2 was chosen as the starting point for the next stage of the preliminary design process.

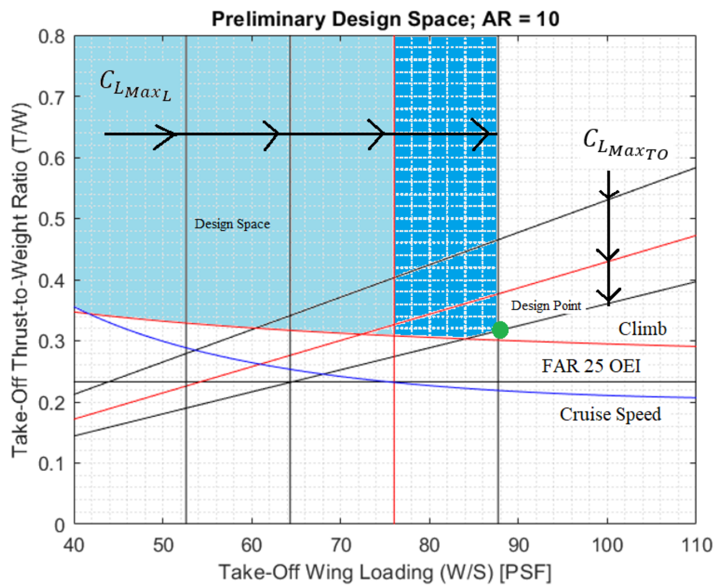


Figure 4.2: Preliminary Design Space

The next step of the design process involved a mission performance analysis to verify the original weights estimations. The main portion of this step to figure out how much fuel was required to complete the mission. By finding this weight of fuel required, the difference between this and the original estimate could be added to the total weight of the plane, and the analysis could be redone. Once the fuel weight guess and the mission fuel weight converged, the aircraft design for those main wing planform values was considered done. The planform values for this design as well as computer-aided designs of the aircraft are shown in Table 4.1 and Figure 4.3, respectively. In Table 4.1, the items are listed as follows: wing loading in units of lbs/ft^2, thrust-to-weight ratio in units of lbs-thrust/lbs-m, taper ratio which is unitless, sweep of the quarter chord of the main wing in degrees, aspect ratio of the wing which is also unitless, and takeoff weight in lbs-m.

Table 4.1: Preliminary Design Variables

W/S	T/W	λ	$\Lambda_{c/4}$	AR	W_{TO}
88	0.311	0.3	25	10	97820

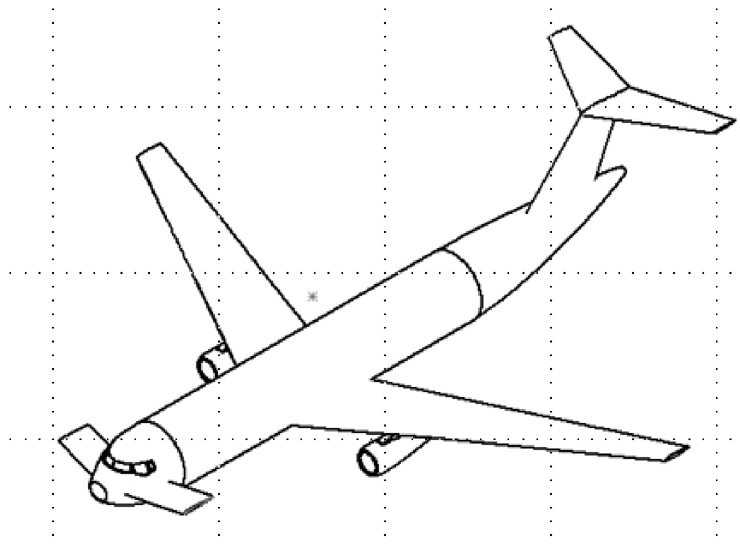


Figure 4.3: Preliminary Design Isometric View

4.1.2 Semester 1 Results

The team finished in second place after completing its analysis of the aircraft, submitting the design report, and presenting the results to the panel of industry members, they finished in second place in all aspects. The team had the second highest grade on the written report and the second highest grade for the oral presentation. The deviations between the grades for the oral presentations were very small among the five teams, varying only by five points between each. The reports had a higher degree of separation and the top two reports were clearly the top two according to the grades Dr. Smith assigned.

It is difficult to say if the success of the team was due to the management style or due to some other external reason based on the grades alone. The author believes that this played a small role in the success, and the team members agreed that the management towards the end of the semester was a large motivator for quality work to be done. The foundation that was laid for the second semester through the Agile Manifesto ideals being followed was very strong. Leading in to the second semester, the team knew exactly what to focus on and was very confident in where they needed to go. The Agile Development practices were refined and the change in objectives during the second semester led to a project that could more easily follow Agile Development practices.

4.2 Senior Design Results: Semester 2

During the second semester, the author and most of the team had less times to work on the project due to new jobs and higher class-loads, but the management styles were more heavily defined, and the processes more closely followed. This led to a definite change in the outlook of the team and a general feeling that the objectives were being completed more often.

4.2.1 Semester 2 Work Output

The objectives for semester two included using the starting point from semester one and varying the first four entries in Table 4.1 through ranges to conduct trade studies. Once the trade studies were completed, the data was plotted to visually analyze trends and identify the best solution. This best solution was then altered to use a fixed or off-the-shelf engine rather than a specific engine for this layout, and various structural analyses were then conducted.

To start this process, the new items required by the customer were evaluated and methods for completing these tasks were laid out. This included a longitudinal and lateral dynamics buildup with handling qualities included and a cost estimation of the life cycle cost with its individual pieces. The longitudinal and lateral dynamics buildup came from Roskam's Design series, volume VII [6] as well as his airplane flight dynamics series [7]. The handling qualities during cruise for this specific category of plane are defined by FAR 25, however some items were taken from military specifications to round out the analysis. Next, the cost estimation module was defined. The methods for finding the life cycle cost are outlined in Roskam's Design series, volume VIII [8]. The main goal was calculating the life cycle cost of the program, which allowed for several other important values to fall out. The equation for total life cycle cost is shown as Eqn. (1), and is comprised of research and development with testing, acquisition, operational, and disposal costs.

$$LCC = C_{RDTE} + C_{ACQ} + C_{OPS} + C_{DISP} \tag{1}$$

4.2.1.1 Software Program Design

Once these items were described and planned, the overall design of the program was completed. The first step of this process was to complete the mission analysis by "hand," or walk through the process with the aide of spreadsheets or some scripting language. Once done, the process was written down and transferred to a flowchart. This idea of flowcharting and planning goes with both the principles of Agile Software Development and Dr. Smith's own recommendations. Dr. Smith emphasized a top-down design approach and a bottom-up implementation. A top-down design approach involves looking at the problem and solution process from a 30000-foot level; the most important items along with logic flow are shown. Then, these items or modules are broken into smaller and smaller pieces until the entirety of the program is essentially defined. With this done, the actual coding of the program can begin. The code should be created from the bottom up; the coding process should start with the smallest pieces defined during the planning portion and these pieces should be integrated according to the flowchart. This extra work done during the planning phase allows for the entire group to understand what the final goal is and how their individual work fits into the grand scheme of the program. The extra planning also allows for far less time to be spent later when debugging, or the fixing of issues that arise after running. The flowchart of the top level of the program is shown in Figure 4.4.

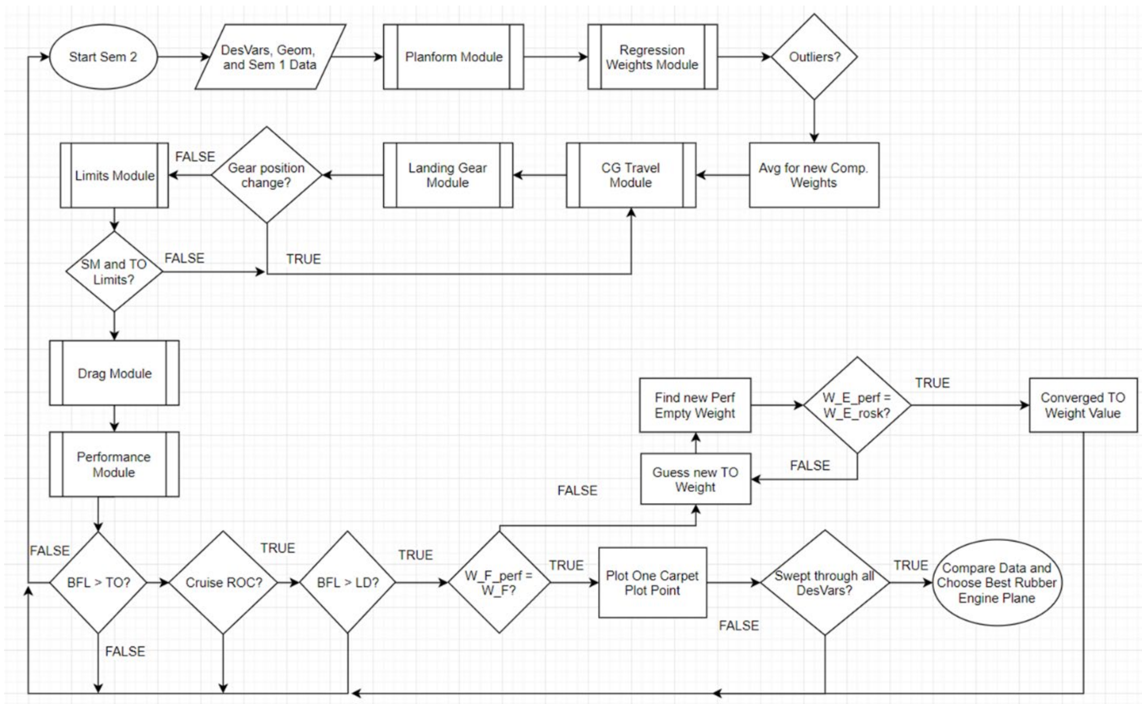


Figure 4.4: Top Level Flowchart of Synthesis Program

4.2.1.2 Data Representation and Analysis

Once the program was implemented and run with all of the new requirements and refinements in place, data for over 500 different plane designs was output. This data needed to be collected and represented visually in order to check the analysis that was done by the program. The program should automatically find the lowest weight, or best, solution but it is always important to double check and cross reference that the program is producing the correct results. This was done by visually analyzing the data via carpet plots. Carpet plots are graphs which can represent up to seven different variables at once. The program varied aspect ratio, taper ratio, wing loading, and main wing quarter chord sweep, and these variables were all represented on a single carpet plot. Plots of takeoff weight, sea level installed thrust required, and cost per aircraft were all created. An example of each of these plots for a main wing quarter chord sweep angle of 25° are shown in Figures 4.5-4.7 respectively. All three of these plots for all other design conditions, as well as a table showing the values that were swept through, can be found in Appendix B.

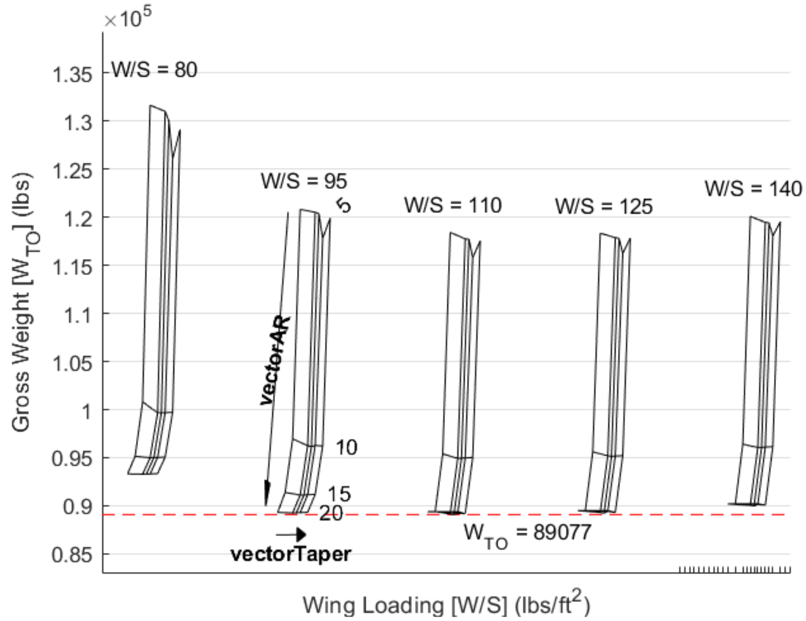


Figure 4.5: Weight Carpet for $\Lambda_{c/4} = 25°$

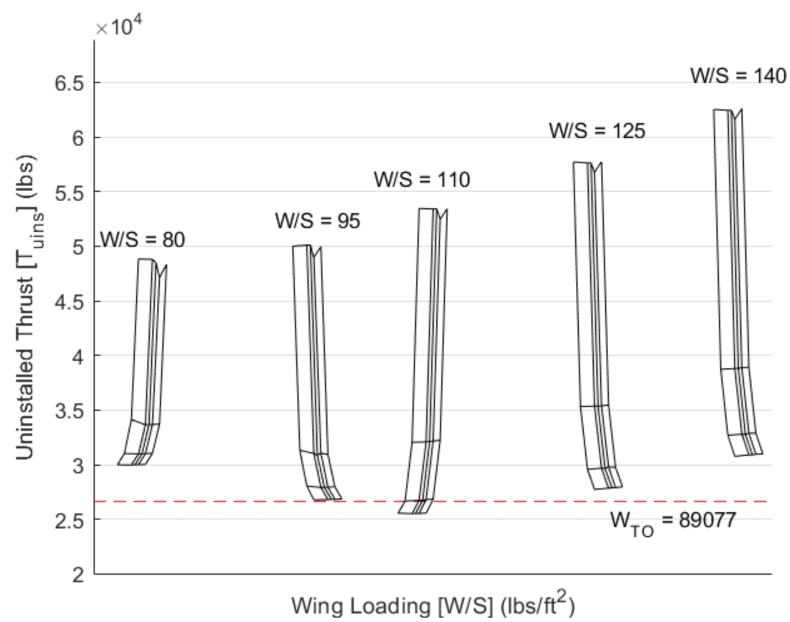


Figure 4.6: Thrust Carpet for $\Lambda_{c/4} = 25°$

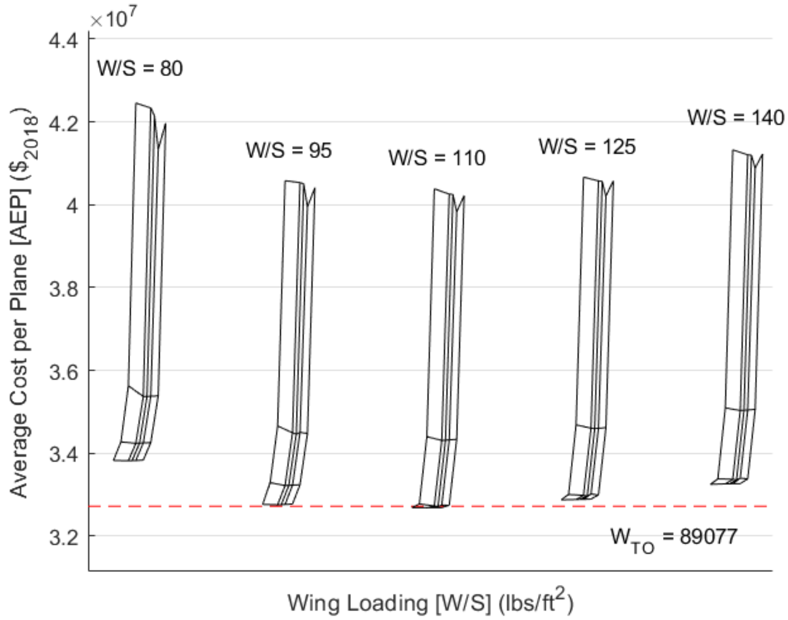


Figure 4.7: Cost Carpet for $\Lambda_{c/4} = 25°$

4.2.1.3 Best Solution Analysis

With the best aircraft now defined, analysis of certain aspects could begin. The first order of business was to swap out the plane-specific engine created by the program to an off-the-shelf engine. This was done to reduce program cost because engine development programs are typically very expensive and beyond the scope of commercial transport jet designs. Once all geometrical properties were known, a computer-aided design model could be created. This model included new lifting surface sizing and positions, new control surface sizing, positions of several internal equipment pieces, several system layouts like air conditioning and fuel, and structural analysis of the main wing and fuselage. The design variables for this converged solution are shown in Table 4.2.

Table 4.2: Best Solution Design Variables

W/S	T/W	λ	$\Lambda_{c/4}$	AR	W_{TO}
110	0.309	0.35	25	15	88077

The new three-dimensional rendering of the plane is shown in Figure 4.7 however the control surfaces are not included. Appendix C will include cutouts, engineering drawings, and systems layouts of this solution as well. By looking into the takeoff weights from Tables 4.1 and 4.2, it is seen that the new design is lighter by about 10000 lbs. This new weight, new geometry of the lifting surfaces, and new engine were all then used for a structural analysis of the fuselage and wing structures. The wing stress analysis was done using Ansys by applying a loading distribution found through a V-n (velocity-loading) diagram as well as the lift distribution along the wing during cruise flight. A picture of the V-n diagram is shown as Figure 4.8.

4.2.2 Semester 2 Results

The team felt increasingly confident in the quality of work accomplished during the semester. After presenting the work to the panel of industry members, the team once again came in second place with a higher score than last semester, and a smaller gap between first and second place. Nearly all items asked of by the customer were completed, and they were completed to a very high standard. When the team was asked to reflect on the difference between semester one and semester two, it was unanimous that the organization and effort that went into using the Agile process was worth it. The team was much more open with communication and very few sprints were delayed. Most of the peer reviews were very short due to few errors and the integration into the baseline was usually seamless.

Also worth mentioning is the interactions of the author with other teams or team leads. Other group leads began to express their shortcomings within their groups and that team morale was much lower at the end of this semester than last semester. They worried that the team had lost sight of the overall goal and were spending too long addressing errors and trying to incorporate changes into the overall project was an arduous and difficult task. The exact opposite was noticed in the author's group. The team morale was high going into the final presentation and the group feels confident that the work completed will please the customers.

Figure 4.8: Best Solution 3-D Rendering

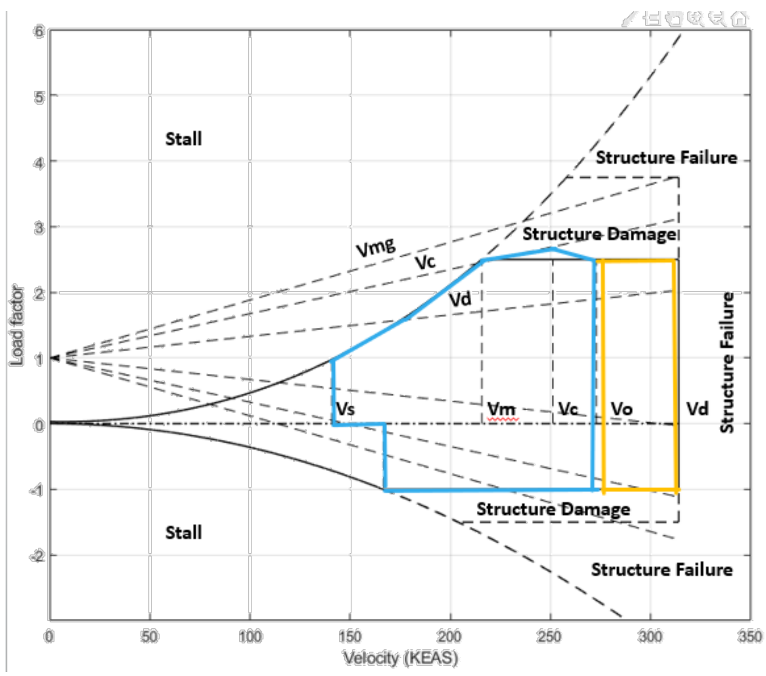


Figure 4.9: V-n Diagram for Best Solution

CHAPTER 5

CONCLUSION

The goal of this thesis was to analyze whether the extra effort involved in the Agile Software Development process was worth the extra effort. This extra effort included extensive planning of tasks before work could begin, constant meeting among the team and with the customer Dr. Smith, extended planning sessions to understand the overall goal of the work being completed, and several other tasks typically outside the scope of the Senior Design class.

The Agile process emphasizes constant and meaningful contact among both the team and with the customer. This was done through the use of sprint planning meetings, scrums, and volunteering to show work first even when incomplete during class time for feedback. Through the scrums and sprint planning, the team was kept aware of the overall objective of the project and kept motivated by seeing progress constantly be updated on the Kanban-style Trello board. By always ensuring the team got extra feedback from Dr. Smith and other groups during class time, the group was able to identify and rectify more errors more quickly than other teams.

The process also details the emphasis on decentralization of work. By allowing each member to be responsible for their own set of tasks, peer reviewing each piece of work, and allowing open communication among the team, this decentralization of work led to higher work output with no loss in work quality. Removing the dependence on a single person to verify all work was correct and not micromanaging each member, the team had more freedom to complete tasks while remaining more motivated to produce quality work.

The team was able to produce more high-quality work while decreasing time spent reworking tasks due to the Agile process. The first semester of the class had this team finishing in second place among all others while these practices were still being developed. During the second semester, the application of Agile principles was much more refined and followed more closely. This led to a large difference in team morale and the perception of success among the team. Although quantitative results of the team's performance during the second semester have not been received yet, the general consensus among most teams is that this group will continue to be a high performance work environment. The application of Agile Software Development is a powerful system and when used correctly leads to teams of motivated individuals producing even high-quality work, and teams that were already high performing can produce even more quality work through more freedom and more motivation.

APPENDIX A

FIGURES OF ALL OTHER LAYOUTS

Figure A.1: Conventional Configuration Example [9]

Figure A.2: Canard Configuration Example [10]

Figure A.3: Joined-Wing Configuration Example [11]

Figure A.4: Blended Wing-Body Configuration Example [12]

APPENDIX B

DESIGN SOFTWARE CARPET PLOTS

Table B.1: Synthesis Program Sweep Values

Variable	Ranges	Units
Wing Load, W/S	80, 95, 110, 125, 140	lbs/ft^2
Taper Ratio, λ	0.1, 0.3, 0.35, 0.4, 0.5	-
Aspect Ratio, AR	5, 10, 15, 20	-
Quarter Chord Sweep, $\Lambda_{c/4}$	5, 10, 20, 25, 35	degrees

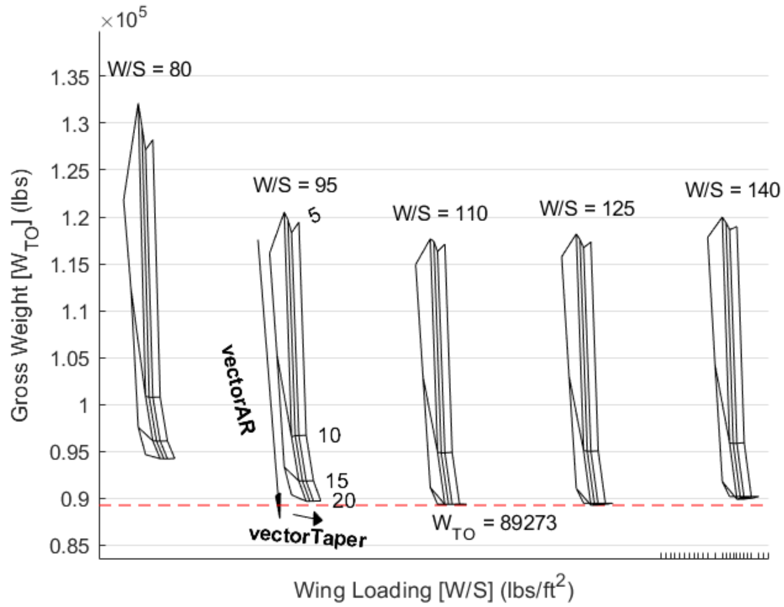


Figure B.1: Weight Carpet for $\Lambda_{c/4} = 5°$

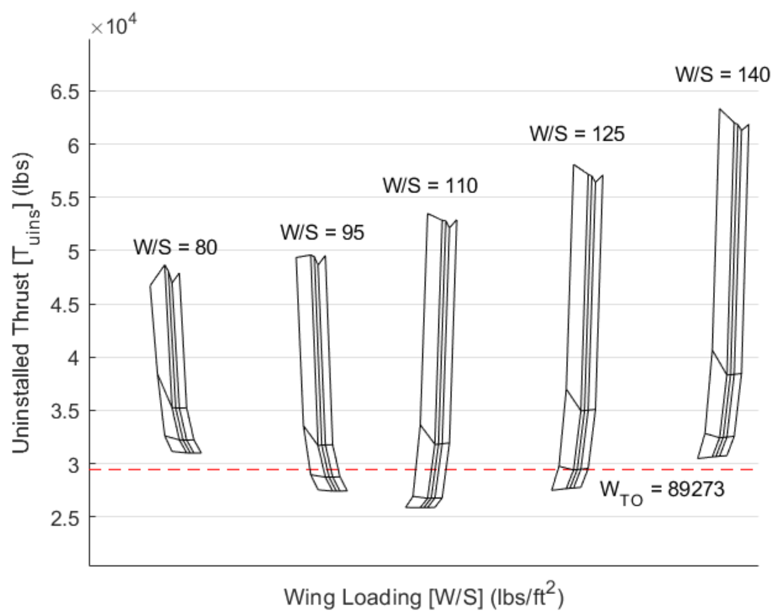


Figure B.2: Thrust Carpet for $\Lambda_{c/4} = 5°$

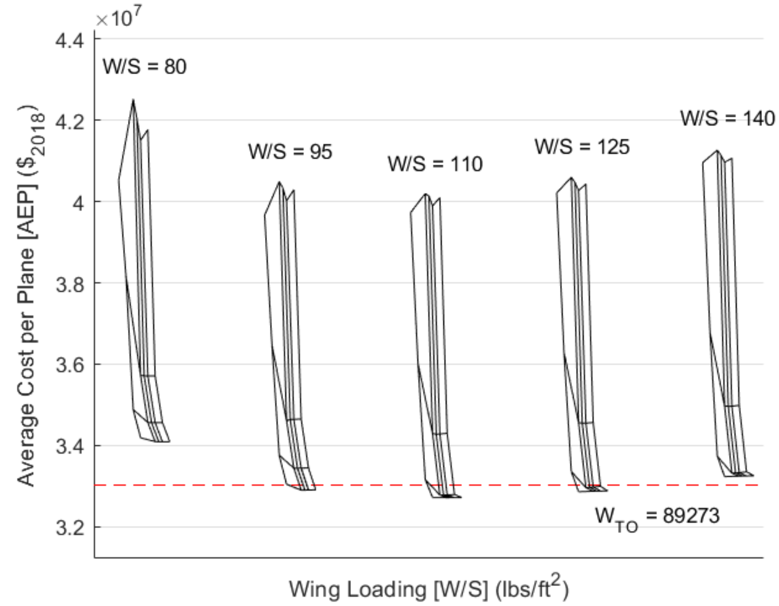


Figure B.3: Cost Carpet for $\Lambda_{c/4} = 5°$

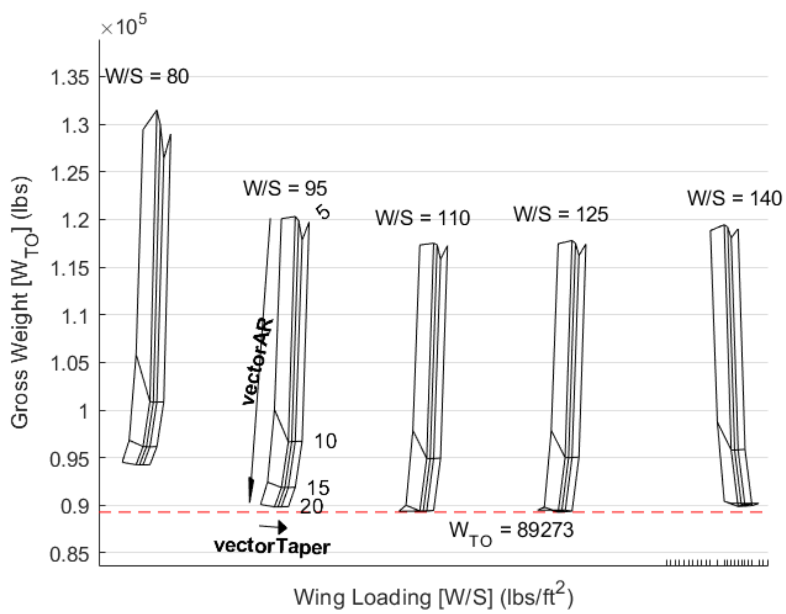


Figure B.4: Weight Carpet for $\Lambda_{c/4} = 10°$

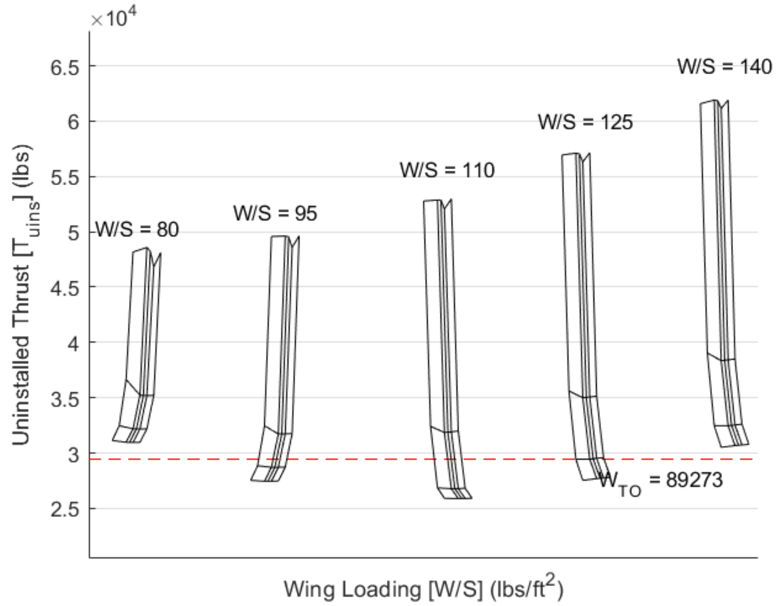


Figure B.5: Thrust Carpet for $\Lambda_{c/4} = 10°$

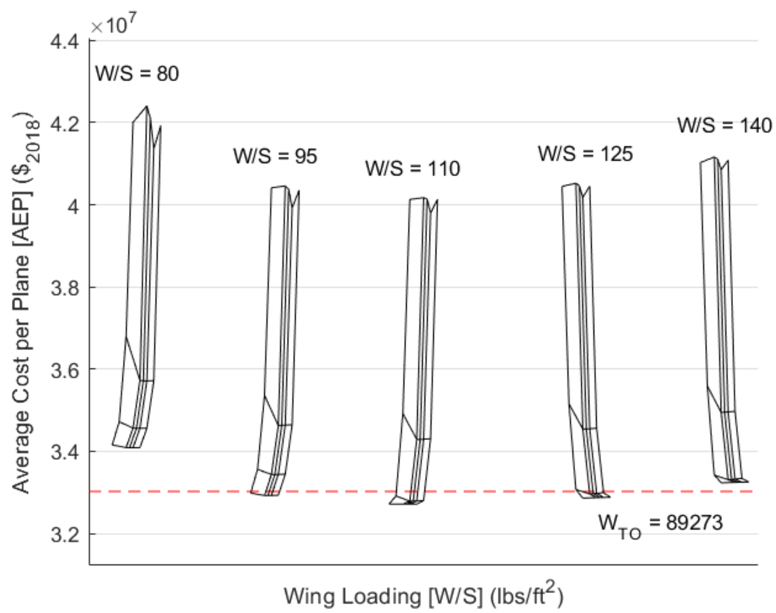


Figure B.6: Cost Carpet for $\Lambda_{c/4} = 10°$

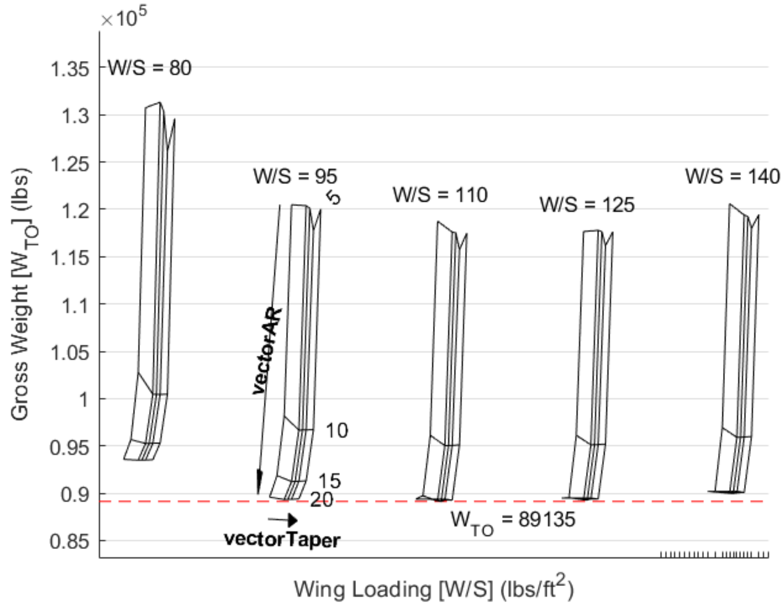


Figure B.7: Weight Carpet for $\Lambda_{c/4} = 20°$

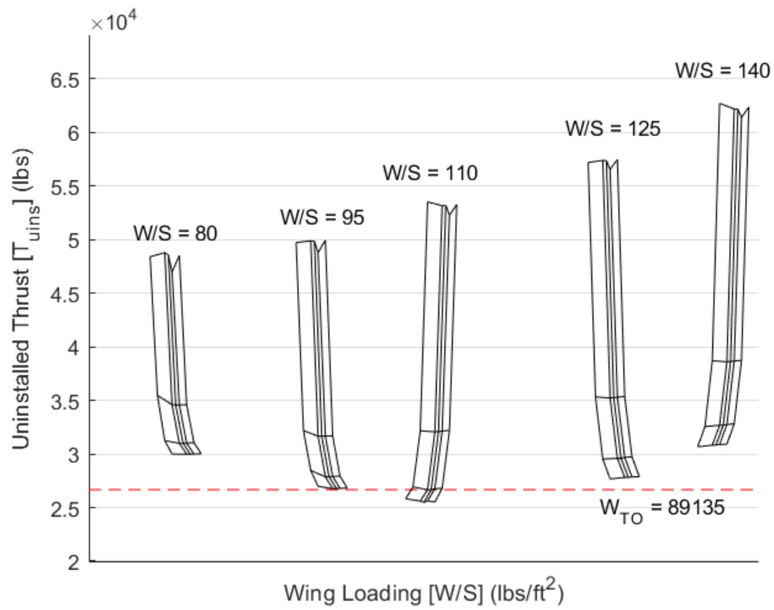


Figure B.8: Thrust Carpet for $\Lambda_{c/4} = 20°$

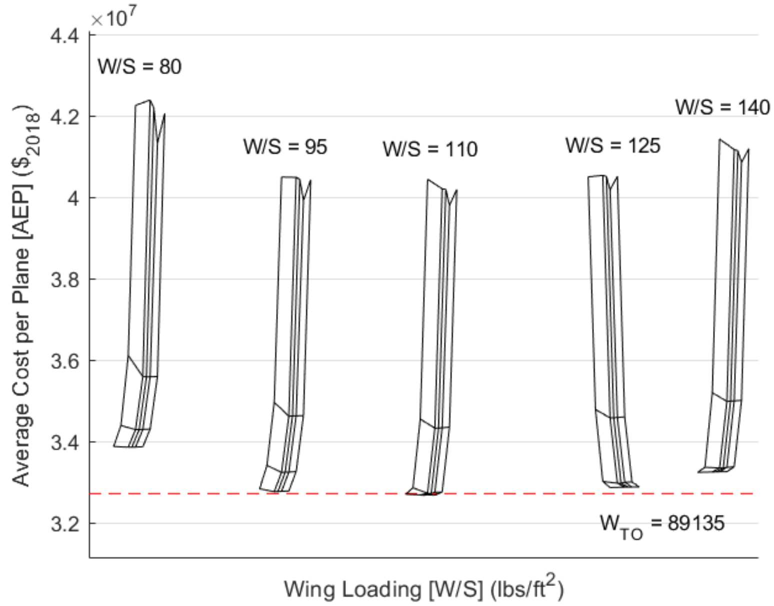


Figure B.9: Cost Carpet for $\Lambda_{c/4} = 20°$

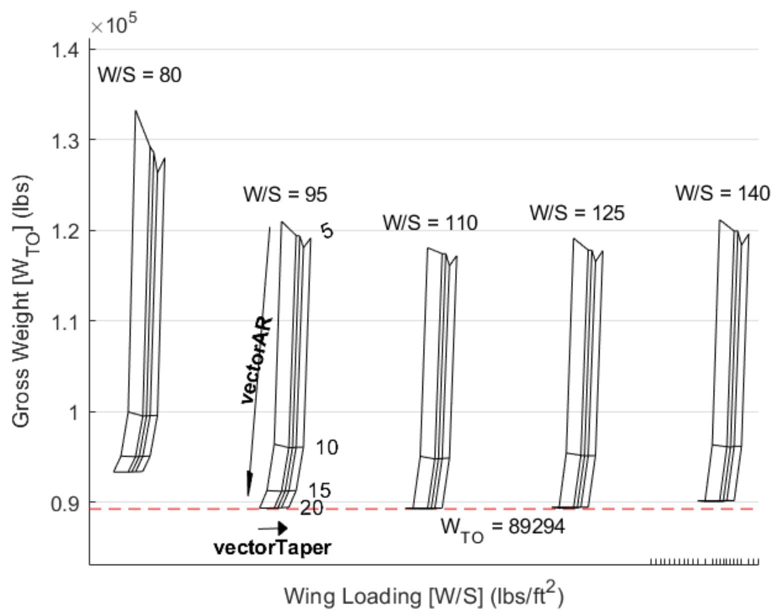


Figure B.10: Weight Carpet for $\Lambda_{c/4} = 35°$

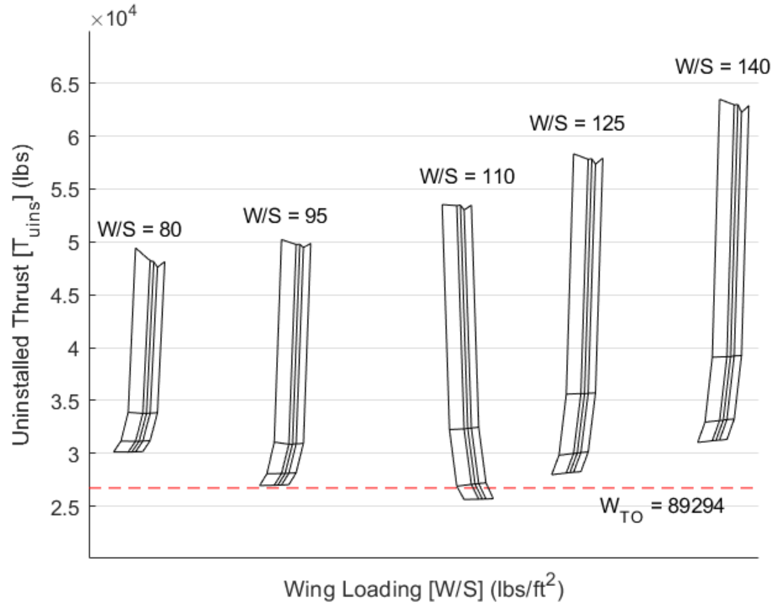


Figure B.11: Thrust Carpet for $\Lambda_{c/4} = 35°$

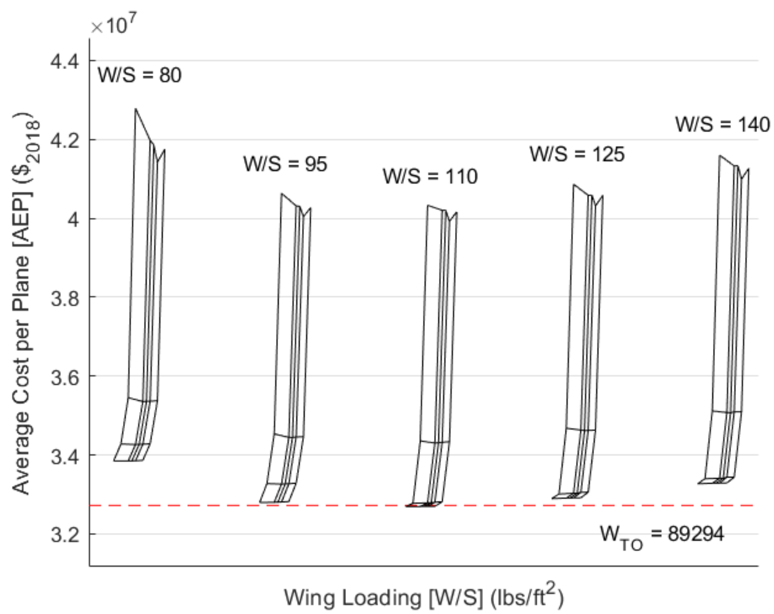
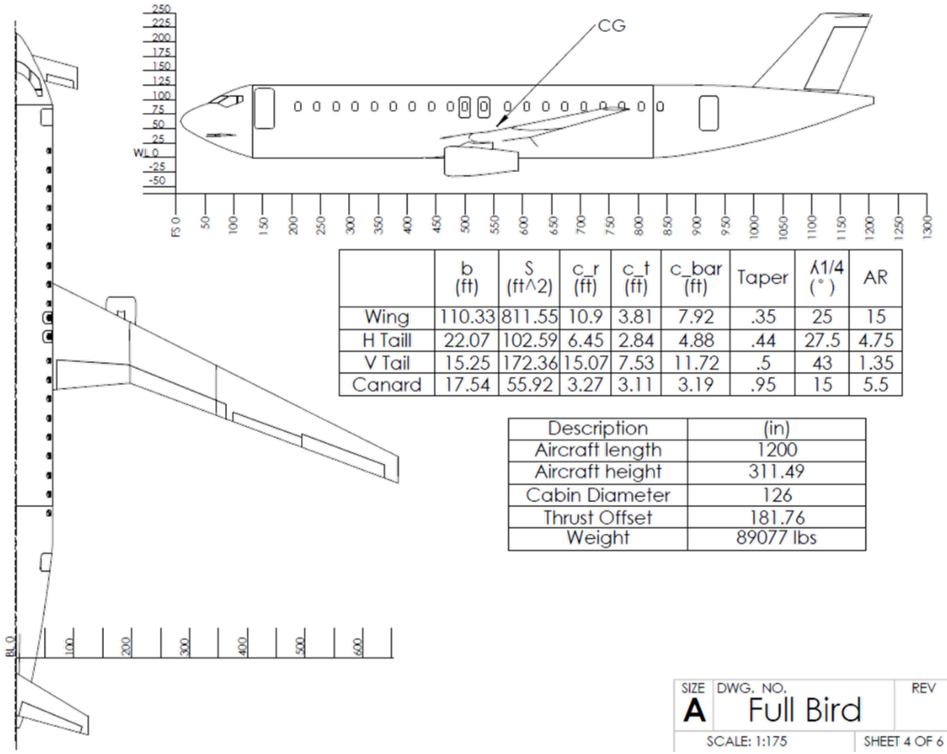


Figure B.12: Cost Carpet for $\Lambda_{c/4} = 35°$

APPENDIX C

BEST SOLUTION DRAWINGS AND LAYOUTS



	b (ft)	S (ft^2)	c_r (ft)	c_t (ft)	c_bar (ft)	Taper	Λ1/4 (°)	AR
Wing	110.33	811.55	10.9	3.81	7.92	.35	25	15
H Taill	22.07	102.59	6.45	2.84	4.88	.44	27.5	4.75
V Tail	15.25	172.36	15.07	7.53	11.72	.5	43	1.35
Canard	17.54	55.92	3.27	3.11	3.19	.95	15	5.5

Description	(in)
Aircraft length	1200
Aircraft height	311.49
Cabin Diameter	126
Thrust Offset	181.76
Weight	89077 lbs

Figure C.1: Planform Views and Characteristic Data for Best Solution

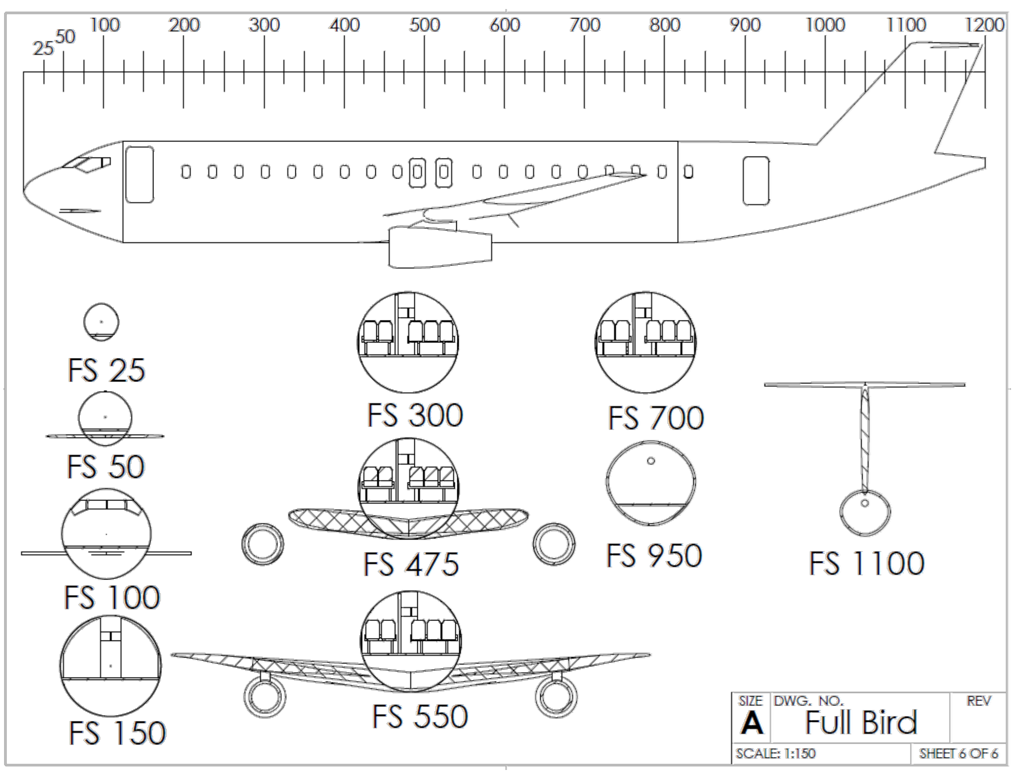


Figure C.2: Cutouts and Section Views for Best Solution

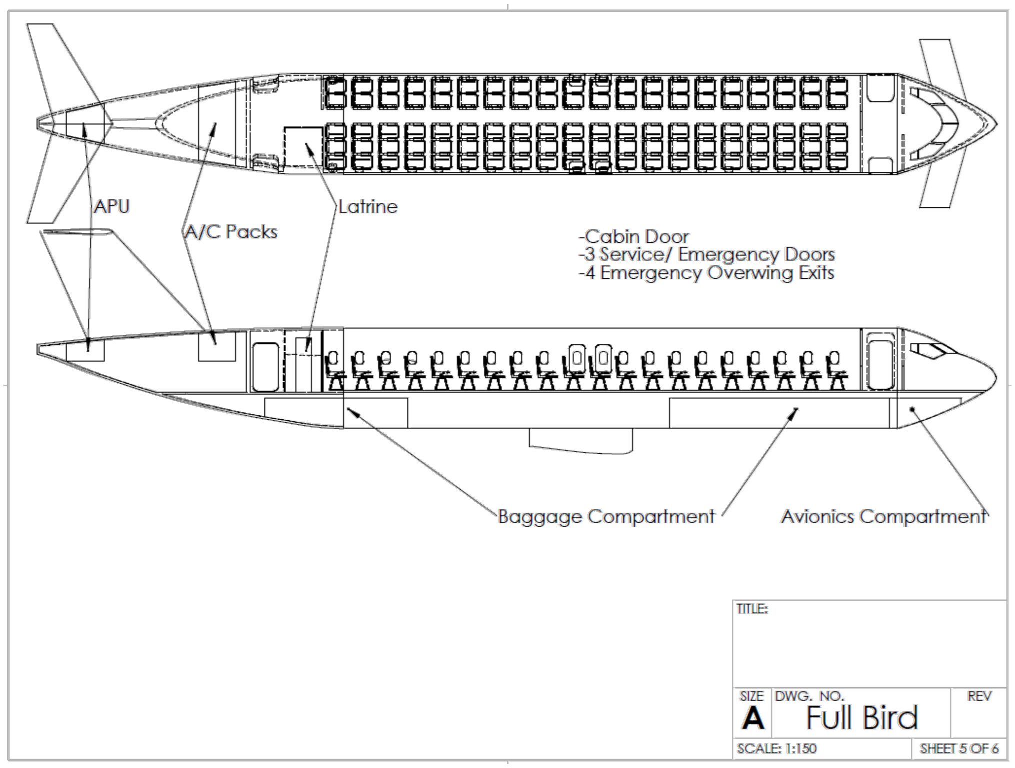


Figure C.3: Seat Layout and Extraneous Items on Best Solution

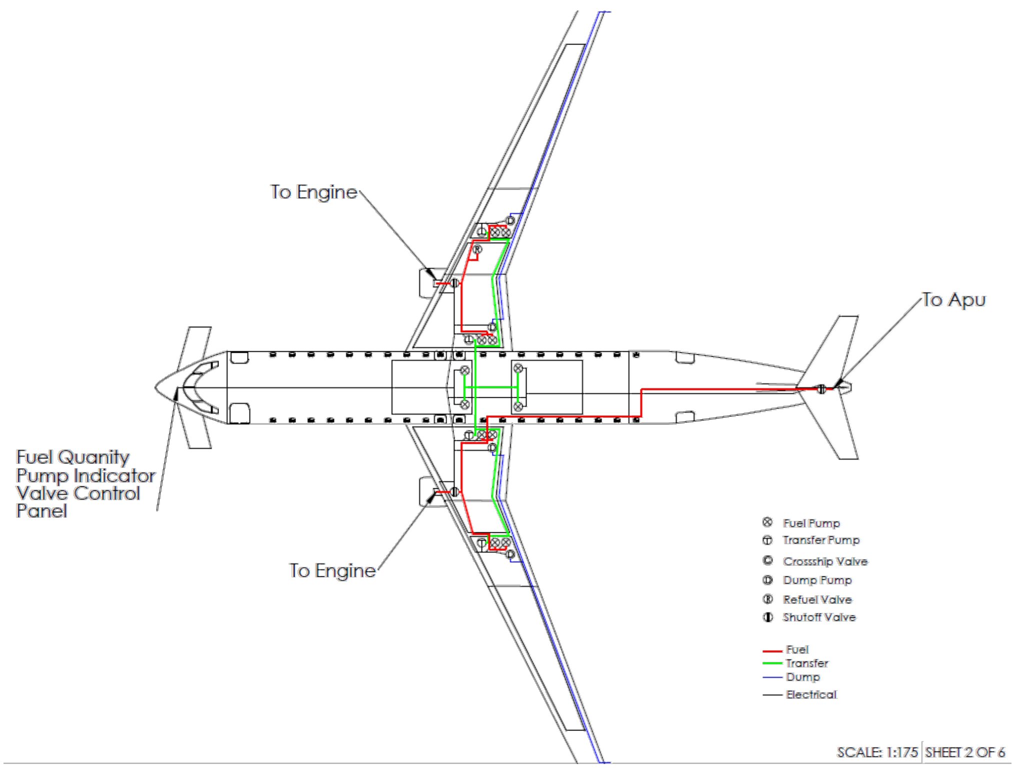


Figure C.4: Fuel System Layout on Best Solution

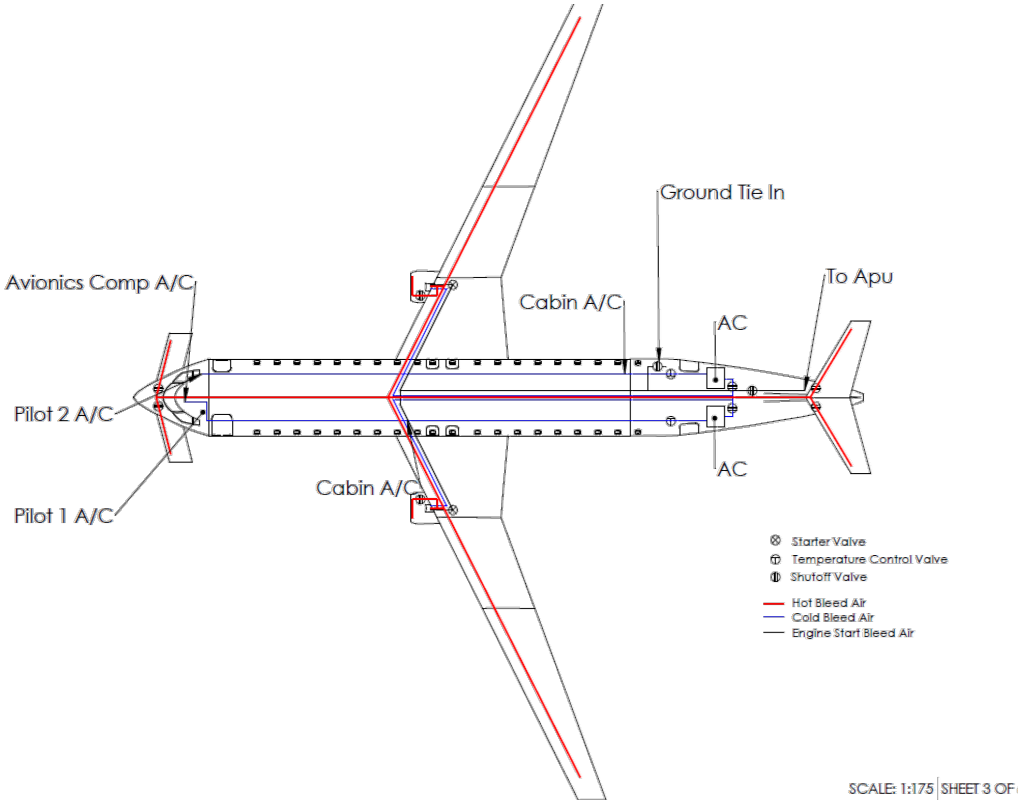


Figure C.5: Bleed Air Systems on Best Solution

REFERENCES

[1] Napolitano, M.R., *Aircraft Dynamics from Modeling to Simulation*, John Wiley & Sons, Inc., Hoboken, NJ, 2012.

[2] Tibboh, "Piaggio P180 Rennes 2010," French Wikipedia, September 2010, URL: https://commons.wikimedia.org/wiki/File:Piaggio_P-180_Avanti_Rennes_2010.jpg [cited 1 May 2019].

[3] Beck, K., Beedle, M., van Bennekm, A., Cockburn, A., Cunningham, W., Fowler, M., Martin, R. C., Mellor, S., Thomas, D., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Scwhaber, K., and Sutherland, J., "Manifesto for Agile Software Development," *Independent Conference*, Vol. 1, Agile Alliance, Snowbird, Utah, 2001.

[4] "12 Principles Behind the Agile Manifesto," *Agile Manifesto*, Agile Alliance, Snowbird, Utah, September 2001, URL: https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/ [cited 1 May 2019].

[5] Roskam, J., *Airplane Design Part I: Preliminary Sizing of Airplanes*, Roskam Aviation and Engineering Corporation, 1st Ed., Ottawa, Kansas, 1985.

[6] Roskam, J., *Airplane Design Part VII: Determination of Stability, Control, and Performance Characteristics: FAR and Military Requirements*, Roskam Aviation and Engineering Corporation, 1st Ed., Ottawa, Kansas, 1985.

[7] Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls*, Part I and II, Design, Analysis, and Research Corporation, 1st Ed., Lawrence, Kansas, 2001.

[8] Roskam, J., *Airplane Design Part VIII: Airplane Cost Estimation: Design, Development, Manufacturing and Operating*, Roskam Aviation and Engineering Corporation, 1st Ed., Ottawa, Kansas, 1985.

[9] Argerich, L., "United Airlines Boeing 767-322ER," February 2009, URL: https://commons.wikimedia.org/wiki/File:United_Airlines_Boeing_767-322ER.jpg [cited 1 May 2019].

[10] Weber, R., "Rutan VariEze - AirExpo Muret 2007 0086 2007-05-12," 12 May 2007, URL: https://commons.wikimedia.org/wiki/File:Rutan_VariEze_-_AirExpo_Muret_2007_0086_2007-05-12.jpg [cited 1 May 2019].

[11] "Outside the Box, Sort of," Lockheed Martin/NASA, 27 January 2012, URL: https://www.nasa.gov/content/outside-the-box-sort-of [cited 1 May 2019].

[12] Landis, T., "X-48B Demonstrates its Unique Triangular Lines," 24 October 2006, URL: https://www.nasa.gov/centers/dryden/multimedia/imagegallery/X-48B/ED06-0198-62.html [cited 1 May 2019].

BIOGRAPHICAL INFORMATION

Thomas Arruda is an Aerospace Engineering major with minors in Mechanical Engineering and Business Administration. He has been a member of the Honors College at the University of Texas at Arlington since the Fall of 2015, and will graduate during the Spring of 2019 *Summa Cum Laude*.

Thomas started an internship with Lockheed Martin Missiles and Fire Control during the Fall of 2018. He plans on entering the workforce with Lockheed Martin immediately after graduation. After a one year sabbatical from academia, he plans to return to UT Arlington to earn a Master’s in Software Engineering.

Thomas has always been fascinated with leadership and motivating other people. He took on leadership roles in at least two projects every semester while completing his undergraduate degree. He plans on applying to a Leadership Development Program with Lockheed Martin which would enable him to enter the realm of project management and functional management for his professional career.