TASS: A Toolkit for Aircraft Sizing and Synthesis

AE6343 A – Aircraft Design I

Presented by: Komahan Boopathy

Presented to: Prof. Dimitri Mavris

October 16th 2015

School of Aerospace Engineering
Georgia Institute of Technology

Honor Code Statement

I certify that I have abided by the honor code of the Georgia Institute of Technology and followed the collaboration guidelines as specified in the project description for this assignment.

Signed:

Contents

List of Figures

List of Tables

TASS: A Toolkit for Aircraft Sizing and Synthesis

Komahan Boopathy komahan@gatech.edu

October 16, 2015

Abstract

A Toolkit for Aircraft Sizing and Synthesis (TASS), capable of performing sizing and synthesis calculations in the context of conceptual design of aircraft is developed in Matlab.[5] The TASS implements energy-based constraint and weight-fraction approach for the mission sizing analyses, and provides the user with a design point that is a function of *thrust-weight ratio* and *wing-loading*. The performance of the TASS is benchmarked against the known metrics of a transonic jet fighter aircraft F-86L Sabre (Sabrejet).

1 Introduction and Motivation

Design of aircraft is subject to tens of thousands of constraints that span across a variety of disciplines such as aerodynamics, structures, controls and performance.[3,4,9,10] Examples of such constraints, to name a few are: achieving a long cruise range, minimizing the take-off distance, fuel-efficient engines, stealth capabilities for reconnaissance, lift-constrained-drag-minimization, stress and strain requirements, and so on. These constraints are indeed the ones that drive the design and play a pivotal role in shaping the end-design. There has been an increased interest in exploring more at initial stages of design to identify conventional as well as non-conventional configurations – a strategy that helps one to get a better insight of potential configurations that can meet the design requirements and eliminate retrospective changes that may be due at later stages of the design process, that are known to be prohibitively expensive.

Similar motivations have led to development of tools that ably perform conceptual analysis in various platforms.[8,11] To this end, this work too aims to develop a toolkit in Matlab[5] known as TASS, that performs an analysis of the early phases of aircraft design *i.e.,* the sizing and synthesis. The scope of current work is limited to the conceptual design phase of aircraft design; more advanced phases in aircraft design (e.g. preliminary, detailed design *etc.*) fall beyond the scope of current work. Nevertheless, modularity is a key consideration in the development of the toolkit, which enables further advancements a function of time.

Energy based constraint analyses encompass an attractive way to start with aircraft design. The main advantage of energy-based approach over conventional approaches is that it employs Lagrangian paradigm of mechanics, as opposed to Newton's world of vector mechanics. The key idea is that aircraft is treated as a system that converts energy from one form to another as the mission progresses. For example, combustion in the engines convert the chemical energy of the fuel into thermal energy, which in-turn accelerates the gases to rotate the turbines to provide mechanical energy to sustain motion. It is easier to think in terms of scalars such as potential energy (depends on position) and kinetic energy (depends on rate of change of position), compared to resolving forces (e.g. lift, thrust) along different directions (e.g. vertical, horizontal) and invoking force and moment balances. Due to such advantages, the TASS preferably employs energy based constraint analyses. The mathematical and physical models used in various stages of the mission are explained as and when they are introduced in later sections and are not explained here for brevity.

This report is organized as follows. Section 4 outlines the mathematical models behind the aircraft sizing and synthesis procedure. Section 3.1 outlines the functionality that is expected from the toolkit. Section **??** contains detailed step-by-step instructions on the working of the tool with the help of an example. Section 5 concludes the report highlighting the key aspects of the tool and lays-out scope for further improvements to the toolkit.

2 Installation and Layout

This section is devoted to provide a high-level overview of the of the software. A Graphical User Interface (GUI) is designed to collect all the mission-specific inputs that the user supplies to the software. From an user-standpoint these input parameters are decided based on the complete break-down of mission requirements (e.g. humanitarian, rescue, military missions).

2.1 Installation

The TASS uses native Matlab code for all of its functionality – including initialization, iterative calculations and graphics, therefore a working version of Matlab[5] is sufficient. The software is tested to work on both UNIX and Windows based workstations. The source code can be obtained by using `git clone https://github.com/komahanb/tass.git` from a *terminal* or *shell* program or by downloading `tass.zip` file and extracting it in a working folder of convenience.

2.2 Constraint Analysis Layout

The TASS provides a constraint analysis plot and enables the user to directly identify a design point in terms of thrust to weight ratio and wing loading.

2.3 Mission Summary Layout

2.4 Weight Analysis Layout

This layout provides a mission analysis for the sizing of the system indicating the converged estimated gross takeoff weight and weight contributions from the different weight groups such as payload, fuel and crew.

3 Fighter Aircraft Example

A brief introduction of the concept under study and a summary of the mission and requirements are outlined next.[7] The design produced by the TASS will be compared to the design of the actual aircraft.

3.1 Aircraft Requirements

The conceptual design process begins with a defined set of requirements. These requirements layout specific constraints and design goals that the end-system must be capable of achieving. These goals and constraints are often presented the form of Request for Proposal (RFP).

3.1.1 Functional Requirements

The functional requirement of the aircraft is to carryout a military mission, which is:

- to carry a crew member weighing 210 *lbs*
- to carry a payload of 432 *lb*

3.1.2 Mission Profile

Table 1 breaks-down the performance requirements of the aircraft to be designed using the TASS.

3.2 Performance Requirements

The elemets in mission profile are decomposed into performance requirements that are tabulated in Table **??**.

Table 1: Requirements of the fighter aircraft and for different stages in the mission profile

Segment	Phase	Performance Requirements
0-1	Takeoff	Distance 4400 ft Clear obstacle at 50 ft Maximum power with afterburners Sea level runway at 90 F
1-2	Climb	Altitude 35400 ft Full military power without afterburners Maximum rate of climb 90 ft/s
2-3	Cruise climb	Altitude 38700 ft Distance 550 nautical miles Cruise speed 458 knots Normal power
3-4	Loiter	Altitude 38700 ft 10 minutes loiter normal power
4-5	Climb	Altitude 47550 ft
5-6	Combat	Maximum power with afterburners Duration 5 minutes Speed 536 knots
6-7	Descend	Descend to 37000 ft
7-8	Cruise	Altitude 37000 ft Distance 550 nautical miles Normal power
8-9	Loiter	10 minutes Altitude 35000 ft Maximum endurance conditions
9-10	Descend	Descend and line up for landing
10-11	Landing	Distance 5000 ft Without highilift devices 10% fuel reserve

Table 2: Performance requirements of the desired aircraft

Item	Performance Requirements
Takeoff distance	4400 ft at sea-level and 90F temperature
Landing distance	5000 ft at sea-level and 90F temperature
Maximum air speed	1200 ft/s at sea-level
Maximum rate of climb	90 ft/s
Manueverability at Peak Altitude	47550 ft
Turn	2g sustained turn [1]

4 Architecture of the Toolkit

This section provides an outline of the theory, physical and mathematical models used in TASS. A rather comprehensive discussion on these topics can be found in the literature.[3,4,6,9,10] Figure 4.1 provides a generic overview of the elements involved in the sizing and synthesis process. As it can be seen, *constraint analyses* and *weight estimations* are pivotal in producing the design. All the discussions in this section are organized in terms of the benchmarking test case Sabrejet F-86 L described in Section 3.1.

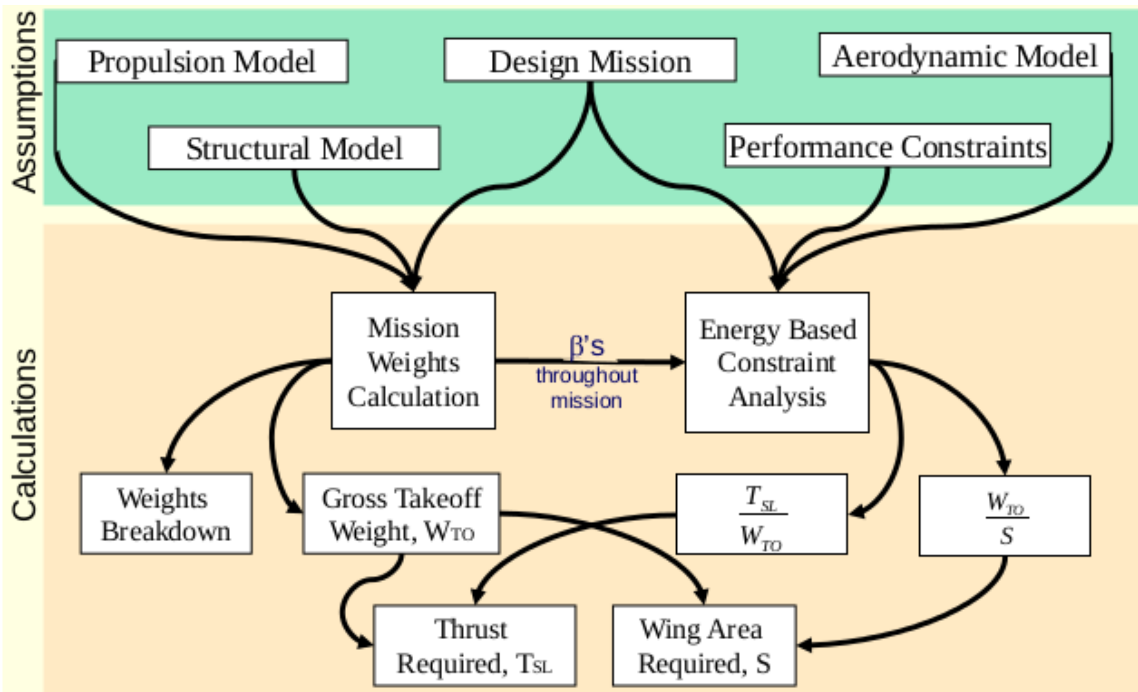

Figure 4.1: A top level overview of sizing and synthesis process implemented in TASS.[7]

Standard Atmosphere Model: The toolkit automatically computes the standard atmosphere conditions based on the U.S. Standard Atmosphere model.[13,14] The standard atmosphere model provides the temperature, speed of sound (Mach number), pressure and density as a function of height in the calculations.

Mattingly Energy Equation: The performance constraints and requirements are set as functions of *thrust loading* (thrust-to-weight ratio) and *wing loading*, the two sizing parameters that contain key information about the top level characteristics of the system.[6] The Mattingly Master Enerygy equation is:

$$\frac{T_{SL}}{W_{TO}} = \frac{\beta}{\alpha} \left\{ \frac{qS}{\beta W_{TO}} \left[K_1 \left(\frac{n\beta W_{TO}}{qS} \right)^2 + K_2 \left(\frac{n\beta W_{TO}}{qS} \right) + C_{D_0} + \frac{R}{qS} \right] + \frac{1}{V} \frac{d}{dt} \left(h + \frac{V^2}{2g_0} \right) \right\} \tag{4.1}$$

Table 3 contains a short definition of the parameters in Eq. 4.1.

Table 3: Parameters in Mattingly Energy Equation

Parameter	Description	English Unit
T_{SL}	sea-level thrust	lb
W_{TO}	take-off gross weight	lb
q	dynamic pressure	lb/ft^2
S	wing planform area	ft^2
C_{D_0}	drag at zero lift	no unit
K_1	Parasitic Drag constant	no unit
K_2	Interference drag	no unit
q	load factor	no unit
R	Resistance Drag	lb
V	velocity(speed)	ft/s
h	altitude from sea level	ft
g_0	gravity at sea level	ft/s^2
α	thrust lapse correction	no unit
β	fuel burn correction	no unit

4.1 Constraint Analysis

By manipulating Eq. 4.1 various constraints pertaining to the mission can be derived. Detailed derivation and treatment of such constraints can be found in Mattingly *et al.*[6]

4.2 Weight Estimation

The takeoff gross weight W_{TO} can be estimated using

$$W_{TO} = W_C + W_P + W_E + W_F \tag{4.2}$$

where W_C is the crew weight, W_P is the payload weight, W_E is the empty weight and W_F is the fuel weight.

Payload and Crew Weights: The design is done to carry a crew member who weighs 210 *lbs* and a payload of 432 *lbs*. Though this is an expendable payload as per the requirement which is dropped-off at some point in the mission, it is assumed that in the event of aborting the mission, the airplane has to carry the payload back to the base, and thus it is treated as a non-expendable payload. User input panels that facilitate the inputs of these weights is shown in Figure 4.2.

Empty Weight Estimation: It is rather difficult to estimate the empty and fuel weight compared to the other counterparts. The empty weight is found using empirical correlations of historical data. In the toolkit both, Mattingly[6] (Eq. 3.52) and Roskam[12] text based models are implemented and the user has the flexibility to choose between the two. The user only needs to input the regression constants A and B in the `weight analysis layout`, under *empty weight panel*. The panel uses the following default values: $A = 0.5091$ and $B = 0.9505$, which pertains to the current design under study and is adopted from Roskam[12] (Eq. 2.16). The input panel is shown in Figure 4.3:

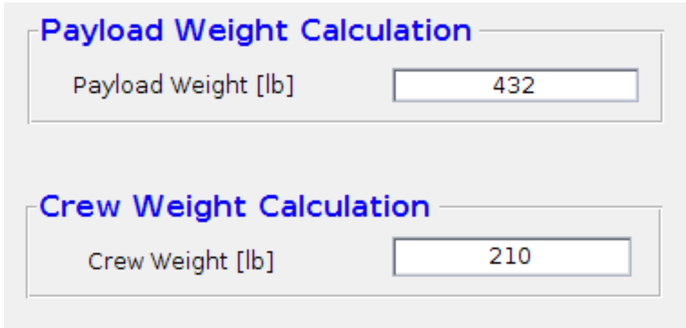

Figure 4.2: Panel designed for inputting the payload and crew weights.

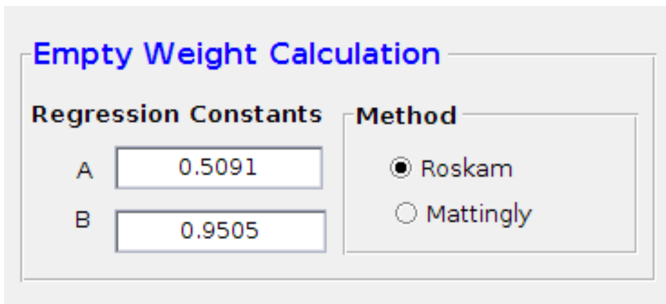

Figure 4.3: Input panel facilitating the selection of different models for empirical estimation of empty weights.

Fuel Weight Estimation: Fuel consumed throughout the mission is modeled by breaking down the mission into various segments as shown in Table 1. The fuel consumed during each segment is modeled based on the values provided in Raymer.[10] For the cruise segments alone, the fuel burn ordinary differential equation is integrated using the values and procedure given in Section (3.4.1) of Mattingly.[6] The the purpose of reusability for similar missions, important inputs have been externalized in the panel, as shown in Figure 4.4. The default values are the ones that are pertaining to the current mission and are assembled from RPF and sources cited above. The user can use the panel for analyzing similar configurations that follow the same mission with different values.

Weights Breakdown: At the end of the mission analysis segment-by-segment, the toolkit provides the weights breakdown in the form of a pie-chart and table (see Figure 4.5).

4.3 Mission Summary

4.4 Benchmarking the design

The comparison of the design obtained using the tool with known configuration of the aircraft follows next. Table 4 presents the performance parameters that are calculated using the tool with that of the existing similar design.[1,2] For this benchmark F86-Sabre-D, the forerunner of F86-Sabre-L and also the RFP data for the J47-GE-33 engine are used. An overall good agreement can be seen between the values except for the thrust loading, which is due to the maximum-speed

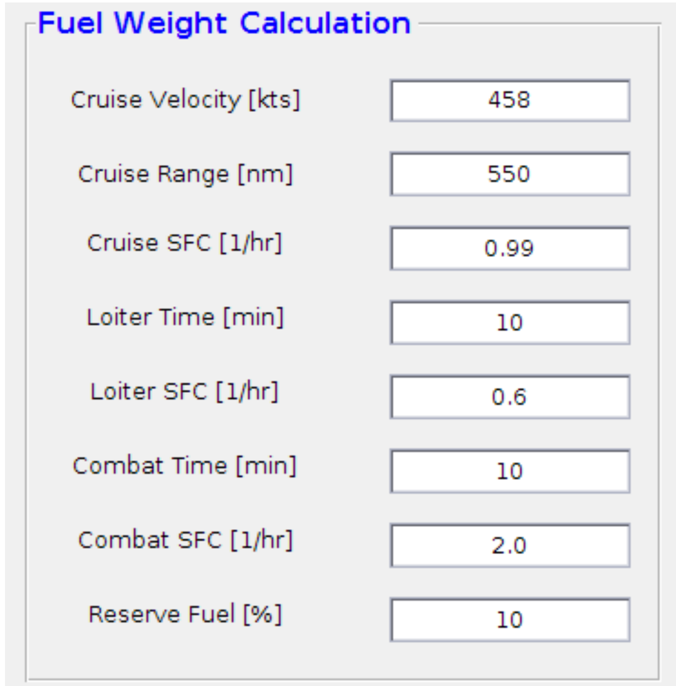

Figure 4.4: Input panel for fuel weight calculations.

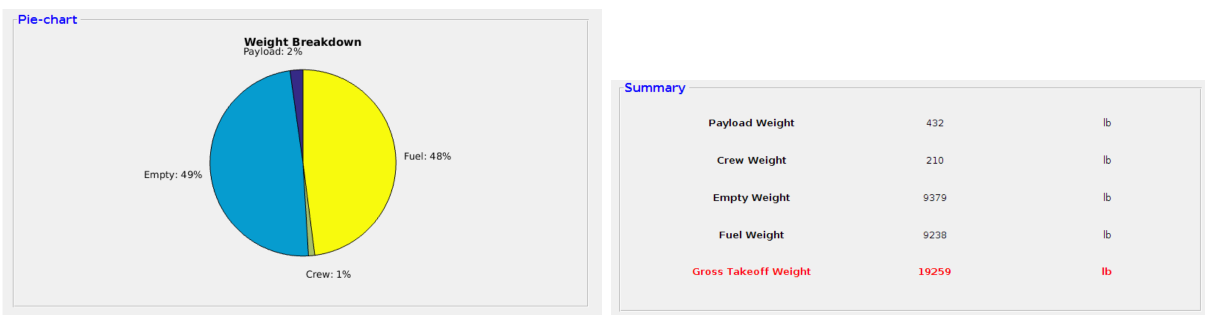

Figure 4.5: A pie-chart showing the weight fractions (left) and the table summarizing the actual weights.

constraint shown in the constraint plot. Perturbing this constraint (e.g. altitude of 40000 instead of sea-level, maximum speed of 1000 ft/s instead of 1200 ft/s will provide the user with a design that has a lower thrust loading.

Table 4: Validation of the F-86L Sabre design produced by the TASS

Design	Wing Loading $[lb/ft^2]$	Thrust Loading	Gross Weight $[lb]$	Planform Area $[ft^2]$
TASS	62	0.80	19259	311
Actual	68	0.38	19975	290

5 Conclusion

This work presented a toolkit that performs sizing and synthesis of the aircraft configurations at the conceptual design level. The toolkit is provided with a graphical user-interface that helps non-expert users to simulate different requirements at the same time. It is validated against a known configuration of F-86 Sabrejet.

References

[1] North american f-86 sabre. $http://www.fighter-planes.com/info/f86sabre.htm$.

[2] The north american f-86 sabre jet. $http://www.456fis.org/F-86_SABER.htm$.

[3] J. P. Fielding. *Introduction to Aircraft Design.* Cambridge University Press, New York, NY, 1st edition, 1999.

[4] D. Howe. *Aircraft conceptual design synthesis.* Professional Engineering Publishing, London, 1st edition, 2000.

[5] MATLAB and Statistics Toolbox Release 2015a. *version 8.5.0 (R2015a).* The MathWorks Inc., Natick, Massachusetts, 2015.

[6] J. D. Mattingly, W. H. Heiser, and D. T. Pratt. *Aircraft Engine Design.* American Institute of Aeronautics and Astronautics, Reston, VA, 2nd edition, 2002.

[7] D. Mavris. *Fixed Wing Aircraft Design I.* Lecture Notes, Georgia Institute of Technology, 2015.

[8] N. E. Antoine and I. M. Kroo. Framework for Aircraft Conceptual Design and Environmental Performance Studies. *AIAA Journal*, 43, No. 10, 2005.

[9] L. M. Nicolai. *Fundamentals of Aircraft Design.* E. P. Domicone Printing Services, Fairborn, OH, 1st edition, 1975.

[10] D. P. Raymer. *Aircraft Design: A Conceptual Approach.* American Institute of Aeronautics and Astronautics, Reston, VA, 3rd edition, 1999.

[11] D. P. Raymer. Modern Use of Spreadsheet Methods for Aircraft Design, Sizing, and Performance Analysis. AIAA Paper, 2004-0534, 2004.

[12] J. Roskam. *Airplane Design: Part I, Preliminary Sizing of Airplanes.* Roskam Aviation and Engineering Corporation, Ottawa, TX, 1985.

[13] S. Sartorius. STDATMO: Standard atmosphere function. MATLAB Central, 2010.

[14] 1976 U.S. standard atmosphere. *United States Committee on Extension to the Standard Atmosphere.* U.S. Government Printing Office, Washington, D.C., 1976.

A Flexibility of the Tool

A.1 Versatility of the Tool

1. The Graphical User Interface (GUI) enables easy inputting of mission parameters to perform the analyses

2. The user can input any "physically sound" set of values as inputs to the three layouts that are the core-part of the TASS: (a) constraint analysis (b) mission analysis (c) mission summary

3. The plots and tables are co-located with the inputs to help study the effect of different parameter perturbations with ease

4. Clear/Reset push-buttons are provided in the tool that helps the user to start-off again with default values

5. TASS employs interpolations and extrapolations whenever it encounters data that are be rather difficult to supply or unavailable:

    - it models the drag polar plots presented in Mattingly[6] Section (2.3.1)
    - standard atmosphere model function[13] used in the code that eliminated the need for treating them as constants at different stages in the mission

A.2 Challenges and Cons

1. Since it is extremely laborious to implement sanity checks on the user-supplied values, it is recommended that the users proactively consider the physical meaning of the values supplied (e.g. range, rate of climb)

2. The current version of TASS can only be used for designing airplanes that follow the "same mission profile" given in the RFP