TASS: A Toolkit for Aircraft Sizing and Synthesis

Komahan Boopathy `komahan@gatech.edu`

September 25, 2015

Abstract

A <u>T</u>oolkit for <u>A</u>ircraft <u>S</u>izing and <u>S</u>ynthesis (TASS), capable of performing sizing and synthesis calculations in the context of conceptual design of aircraft is developed in Matlab.[3] TASS implements energy-based constraint and weight fraction approach for the mission sizing analyses. The performance of TASS is benchmarked against the known metrics of a transonic jet fighter aircraft F-86L Sabre (Sabrejet).

1   Introduction

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.[1,2,4,5] 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 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 restrospective changes that may be due at later stages in design process, that are known to be prohibitively expensive.

Similar motivations have led to the development of tools that ably perform conceptual analysis in various platforms.[6] To this end, this work too aims to develop a toolkit in Matlab[3] known shortly as TASS, that performs some of the very early phases of aircraft design *i.e.,* the sizing and synthesis. The scope of the current work is limited to the conceptual design phase of aircraft design; more advanced phases in aircraft design (preliminary, detailed design *etc.*) fall beyond the scope of the current work. Nevertheless, modularity is a key consideration in the development of the toolkit, which enables further advancements straight-forward function of time.

A brief introduction of the concept under study and a summary of the mission and requirements are outlined next. Energy based constraint analyses form 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 kinetic energy (depends on velocity) and potential energy (depends on position), compared to resolving forces (e.g. lift, thrust) in along different directions (e.g. vertical, horizontal). Due to these advantages, TASS preferably employs energy based constraint analyses. The mathematical and physical models used in individual disciplines are explained as and when they are introduced in later sections and are not explained here in the spirit of brevity.

This report is organized as follows. Section 3 outlines the mathematical models behind the aircraft sizing and synthesis procedure. Section 2 outlines the functionality that is expected from the toolkit. Section 4 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 Requirements of the Toolkit

The requirements for this toolkit are outlined in the project description.[?]

3 Theory

This section provides a quick outline of the theory under TASS. A rather comprehensive discussion of these topics can be found in the literature.[1,2,4,5]

4 Installation and Benchmarking

4.1 Getting Started

TASS uses native Matlab code for all of its functionality – including initialization, iterative calculations and graphics, therefore a working version of Matlab[3] 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-master.zip` file and extracting it in a working folder of convenience.

5 Conclusion

This work presented a generic 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] J. P. Fielding. *Introduction to Aircraft Design.* Cambridge University Press, New York, NY, 1st edition, 1999.

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

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

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

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

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

A   Flexibility of the Tool

A.1   Versatility of the Tool

1. The Graphical User Interface (GUI) takes the set of inputs and perform the analyses. As long as the inputs specified are within the validity of the underlying mathematical models, the tool will work for any set of requirements as explained in Section **??**.

2. TASS can operate with both English and SI units

3. TASS employs interpolations and extrapolations whenever it encounters data that are be rather difficult to supply or unavailable (e.g.)

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)