Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, available online 16 July 2018
DOI: 10.1177/0954410018788922

The GENUS Aircraft Conceptual Design Environment

H Smith[1], D Sziroczák[1], G E Abbe[2], P Okonkwo[2]
[1]Aircraft Design Group, School of Aerospace, Transport and Manufacturing, Cranfield University, UK.
[2]Air Force Institute of Technology, Kaduna, Nigeria

Abstract

The design of aircraft has evolved over time from the classical design approach to the more modern computer based design method utilising multivariate design optimisation. In recent years aircraft concepts and configurations have become more diverse and complex thus pushing many synthesis packages beyond their capability. Furthermore, many examples of aircraft design software focus on the analysis of one particular concept thus requiring separate packages for each concept. This can lead to complications in comparing concepts and configurations as differences in performance may originate from different prediction toolsets being used. This paper presents the GENUS Aircraft Design Framework developed by Cranfield University's Aircraft Design Group to address these issues. The paper reviews available aircraft design methodologies and describes the challenges faced in their development and application. Following this, the GENUS aircraft design environment is introduced, along with the theoretical background and practical reasoning behind the program architecture. Particular attention is given to the programming, choice of methodology and optimization techniques involved. Subsequently, some applications of the developed methodology, implemented in the framework are presented to illustrate the diversity of the approach. Three special classes of aircraft design concept are presented briefly.

Keywords

GENUS, Aircraft design, conceptual design, knowledge management, design framework, design environment

1. Introduction

The aircraft design process is a field of science in itself; there are various theories and methodologies developed over the years by the practitioners of these methods, sometimes referred to as a "black art". Often methodologies are based on the individuals' or organizations' beliefs of how design should be performed, what has worked in the past, and various interpretations of different design theories. Inevitably, over the relatively short history of modern aviation, there have been various aircraft design theories, schools, methods, books, guides, computer tools and so on. Perhaps, the best known examples of aircraft design theories are Howe's 1, Raymer's 2, Roskam's 3,Torenbeek's 4 and Stinton's 5, Corke's 6 and Jenkinson's 7. Howe's design method considers configuration, propulsion, layout, mass, performance, aerodynamics, cost and optimization. The methods discussed are limited to speeds below Mach 3. Raymer discusses sizing, geometry, aerodynamics, configuration, systems integration, propulsion, weights, stability, performance, and also cost and trade studies. Its Mach range extends up to values just below Mach 4. Roskam's aircraft design integrates sizing, configuration, structural and systems layout, weight estimation, performance and stability evaluation methods with the "softer" aspects of design such as cost, design development, manufacturing and operation parts. The maximum Mach numbers are usually discussed up to M 2.5. Torenbeek includes configuration, systems and propulsion layout, flight deck and structures design, performance, but does not discuss cost aspects or supersonic flight. Stinton considers aspects of airworthiness, configuration, performance, propulsion, stability and weight estimation. His methods are not applied to supersonic designs. Thomas Corke, a professor of engineering at the Notre Dame University, combined aerodynamics, structures, propulsion, stability and control to develop a design philosophy that captures the interactions of different disciplines within the aircraft design process and the compromises resulting from such interactions. Jenkinson deals with aspects of modern civil jet aircraft design.

In addition to the aforementioned few, there are other authors with publications on aircraft design. Crawford's 8 book deals with practical aspects of GA aircraft design. Fielding 9 also gives an excellent overview on aircraft design, with many practical examples.

In addition to the classical printed version of the design methods, there are many computerized options available ranging from simple codes to design environments. These design environments are computer based, with variable fidelity levels and features. Levels of fidelity can be described based on the work of Robinson and Martin (NASA) 10. They identified different levels of fidelity with numbers ranging from 0 to 4, representing simple empirical or analytical equations to full 3D, and knowledge based methods. The method was originally developed for NASA's System Analysis Project (SAP), which aimed to standardize definitions of design fidelity. There are numerous examples of these design environments. Among the first ones were General Dynamic's SYNAC (Synthesis of Aircraft) 1967, Boeing's CPDS (Computer-aided Preliminary Design System) 1972 11, NASA's ACSYNT (Aircraft Synthesis) 1987, Northrop's CDS (Conceptual Design Synthesis) and Delft's Aircraft Design and Analysis System (preceded by the Delft Interfaculty CAD Installation, ICI). From the more modern ones, NASA's Integrated Design and Engineering Analysis (IDEA) 12 environment uses the Adaptive Modelling Language as its underlying framework. It integrates a higher fidelity method to evaluate geometry, configuration, propulsion, aerothermodynamics, trajectory and structural analysis. An earlier effort from NASA was the Advanced Engineering Environment (AEE) 13, part of the Next Generation Launch Technology program. It uses an Oracle database to link the modules together, and uses code developed independently in various NASA centres.

Some successful publicly available codes are so widespread today, that some of them have become the de facto industry standard for many aerospace companies. Piano 14 is perhaps the most commonly known computer aided tool, used by aircraft and engine OEMs worldwide. Its primary area of application is for the analysis of conventional subsonic commercial aircraft in the preliminary design phase, especially powerplant-airframe integration. The code also allows designers to perform competitor evaluations, performance studies and emission estimations among many other aspects of aircraft design. It also has a large and validated database of aircraft designs. NASA's Flight Optimization System (FLOPS) 15, 16 is also widely known amongst aircraft designers. It is a collection of programs from various engineering disciplines that are integrated into an analytical design tool which is capable of producing conceptual and preliminary level designs. It also integrates optimization algorithms, that allow the user to set up variable driven inputs and generate optimized output variables.

VSP and Open VSP 17, 18 are part of NASA's aircraft design software tools. VSP stands for Vehicle Sketch Pad, and it can be used to create parametrized aircraft geometry representations. These 3D models then can be exported into various engineering geometry formats for further analysis. Open VSP is the open source release of the code, which also integrates analysis tools such as VSPAero. Although the software has a comprehensive set of modelling tools, the analysis capability is very limited, due to its open source nature. Results available are for example Centre of Gravity (or volumetric centre), wetted area, and similar geometry derived properties.

The APD design software from Pacelab 19, 20 is also among the well-known aircraft design tools in existence. APD is a conceptual and preliminary level knowledge based aircraft design tool. It was designed to provide well-defined interfaces that allow it to communicate with other software, offering the user the option to use higher fidelity methods or to use the code as a platform for detailed aircraft-level analysis. Pacelab also developed a systems architecture design tool, SysArc 21, 22, and various solutions for more detailed aspects such as flight path optimization.

Multidisciplinary Integrated Conceptual Aircraft Design and Optimization (MICADO) 23 developed by Aachen University, is aimed to conduct fast parameter studies with minimal user input using an .xml interface to couple various software together. It has a modular architecture, allowing different analysis software (FEA, CFD, etc.) to be integrated in the environment, but also uses more empirical methods such as Digital DATCOM 24. The method it uses for the design synthesis is not documented. Another example, DARcorporation's Advanced Aircraft Analysis 25 program, represents a lower complexity procedure based on Roskam's methodology. Raymer also has a design environment, the Conceptual Design Corporation's RDS 26 based on his methodology. The software provides a comprehensive tool for aircraft design, including economic analysis for aircraft and operations and built in optimization tools. Adjustment factors are available when the methods are applied to designs where they are outside the bounds of applicability. CEASIOM 27 is a free engineering environment, developed within the SimPAC project, utilizing various design methodologies, for example its weight estimation module incorporates Howe's, Torenbeek's, Raymer's, USAF, and Cessna methods, from which the user can select the desired method. Extensive use of this environment has been made and a broad list of publications are available demonstrating the capability of the environment 28, 29, 30, 31, 32, 33, 34, 35.

Desktop Aeronautics developed the Program for Aircraft Synthesis Studies (PASS) 36. It is suited for conventional subsonic layouts, but is also capable of analysing supersonic business jets or oblique wings. However, it does not state what methodology is used for its modules.

Focusing more on the MDO aspects, the AGILE development framework 37, 38, 39 is reported to be currently under development by a wide consortium of industry, research agency and academic partners. The EU collaboration project was started in

2015, and it is estimated to last until late 2018. The framework aims to further develop and speed up current MDO methodologies using a distributed approach. There are various examples available demonstrating the effectiveness of the approach, applied for conventional and non-conventional (BWB, box-wing, braced-wing, etc) configurations.
As a more specialist application, Hypersonic Aerospace Sizing Analysis (HASA) 40 was developed by NASA. As the name implies, HASA focuses on hypersonic transports and single stage to orbit vehicles, and for validation, compares the methods to existing designs and concepts. For a longer list, Chudoba 41 lists a summary of 86 aerospace vehicle synthesis systems in his book.
In addition to the more generic design methodologies, there are many methods that are applicable only to a single specific class of aerospace vehicle. A review of these specialized methodologies is outside the scope of the current paper.

2. The GENUS aircraft conceptual design environment

Many design synthesis programmes have been developed to investigate the characteristics of a specific aircraft configuration or concept. The resulting parametric studies or design space investigations, for the specific class of vehicle, being the desired end point. More recently there has developed a strong need to make comparisons between different concepts – particularly with regard to identifying more environmentally benign civil aircraft. Whilst the results from these separate investigations can be compared there is a risk that performance differences may result due to differences in prediction tools rather than the concepts themselves. The GENUS Aircraft Conceptual Design environment is a new software tool, developed by researchers in Cranfield University's Aircraft Design Group, based on the vision developed by the author. The programme seeks to enable the comparison of concepts and configurations on a self-consistent basis by using identical models and toolsets where applicable. The accuracy of the results would, of course, be limited by the fidelity of the toolset but relative differences will be more a function of the configuration and concept and less a function of the use of different tools. Thus, a design synthesis framework has been devised by the authors to permit a wide range of configurations and concepts to be modelled. It is intended that the compatibility of the framework can be extended to further configurations and concepts through the addition of extra modules. To accommodate this flexibility, and to get the best from the code and user, it is assumed that the user is a knowledgeable aircraft designer with some programming skills.
The word 'genus' originates from either the Latin term genus (descent or family) or the Greek 'genos' (race, kin). It is used by biologists as a taxonomic rank to classify organisms; Genus is positioned above the rank of species and below the rank of family in the hierarchy. The choice of this name represents the tool's place in the taxonomy of aerospace vehicles; it is a design environment that is not only focused on a particular type of vehicle (single species) but rather a large and diverse group of aerospace vehicles.
An additional motivating factor behind the development of this tool came from the experience of supervising many post-graduate level researchers in the field of aircraft conceptual design. More often than not researchers, often under severe time constraints, would choose the first and most convenient tools and methods to achieve their specific aircraft design objective. And while the majority of these researchers have done an excellent job with their ad-hoc tools and methods, this often results in a

barrier being created to the continuity and development of their work; a serious knowledge management issue. Why a tool or method can't be reused by future researchers can be due to many reasons; the most common ones encountered being unstructured, monolithic, poorly commented code - lack of documentation - use of proprietary tools that may not be accessible in the future - use of specific versions of proprietary software causing future compatibility issues and the adoption of niche programming languages.

Examples of concept/configuration specific multivariate optimisation aircraft design synthesis packages include Djafri 42, Siegers 43, Woodford 44, Watjatrukul 45, Niyomthai 46 and Rajendran 47. Focusing primarily on slightly different aspects, Chudoba 48 investigated conceptual design from a more fundamental stability and control point of view.

One key issue for aircraft design tools is the topic of data format and flow. This key question is common to most research in this area. One of many successful attempts at standardising data flow, DLR has developed a common configuration format referred to as CPACS 49, 50. This format is used as the primary means of data transfer throughout all DLR aircraft design projects and to couple established MDO environments together. The additional great benefit of using the standardised format is that it enables the efficient collaboration between different contributors and their own aircraft design methods during large projects. Essentially CPACS couples the different models and namespaces of the various contributing design environments which enables all players to use their own potentially proprietary tools without modifications. The CPACS standard offers methodology to define conventional aircraft components similarly to most other commonly used design tools available today. Complex/unconventional configurations can be theoretically dealt with by combining the various predefined components.

The GENUS environment in its current format covers a smaller breadth of aircraft design; at the moment it is specifically created to enable the conceptual level design of various, potentially radically different aircraft configurations. With this approach, the "tool bias" can be avoided, where comparing different configurations, such as a BWB and a conventional configuration, one or the other can be shown in a better light, purely because different tools were used to evaluate the design concepts. The GENUS environment is thus aimed to reuse and generalize lower level tools as much as possible, rather to provide seamless link to a large variety of high fidelity tools. Furthermore, because most of the tools are appropriate for conceptual level design, the execution duration is fairly small compared to a higher fidelity (FEA, Navier-Stokes CFD, etc.) tool. This means, that computational overheads such as writing and reading input/output files from the hard disc can amount to significant proportion of calculation time. For this reason, most codes are modified and integrated into the environment, to enable programmatic data transfer as opposed to using these files, speeding up the individual execution in the process. Since GENUS was written in JAVA, it provides convenient interfacing to many other programming languages, and even where no native solution exists, practically all significant languages can be interfaced when going through JNI and a C wrapper. It would be fairly easy to connect the software using the JAVA API to CPACS and similar. However, the original version of the GENUS tool did not consider data and workflow integration into standards. Current and future researchers on the project are investigating options to enable cooperation with such international efforts.

Based on the GENUS concept, the first working version of the tool was developed by the authors. The demonstration of the framework as applied to the design of solar powered unmanned aerial vehicles was completed by Abbe 51. Okonkwo 53 applied the framework to Blended-Wing-Body (BWB) aircraft concepts, and Sziroczák 54 applied it to hypersonic transport and space launcher vehicles, his work also describes the major part of the programming architecture design and implementation. These three references demonstrate the capability of the tool to implement the methodologies developed for various classes of aerospace vehicles. Subsequent to these founding studies GENUS is being applied to supersonic business jets 55 and Unmanned Combat Air Vehicles 56.

The GENUS software tool's main task is to implement the *Design framework*, the structure of which is shown in Figure 1.

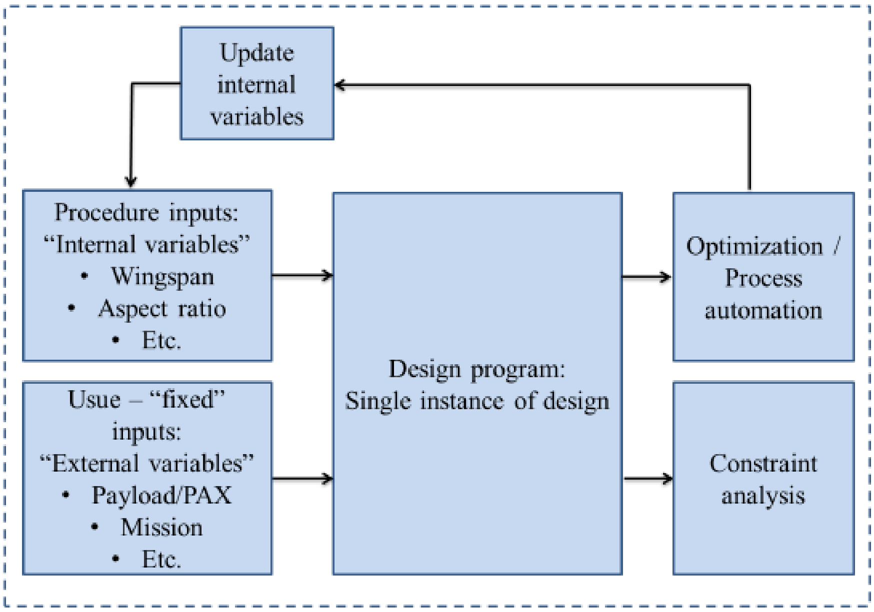

Figure 1: Genus Design Framework

The *Design framework* allows the user to manipulate the tool using a graphical user interface. Behind the graphics interface, the various modules available for the different classes of aircraft design are loaded into the *Design program*. This essential part of the framework is where the developed conceptual design methodologies are implemented; based on the user defined inputs. This part of the code generates a single instance of aircraft conceptual design. This instance is then further processed using the constraint analysis tools. Using the outputs, design responses (objectives, constraints) can be provided for an optimization or exploration loop to drive the design process.

The key element of the *Design framework* is the *Design program*, using a combination of various developed modules to facilitate the production of a single aircraft design. Synthesis of an aircraft design is a complex task which requires a broad knowledge of multi-disciplinary design areas or modules. The modules typically addressed during aircraft design are the following:

- Geometry
- Mission specification
- Propulsion
- Mass estimation
- Aero(thermo)dynamics
- Systems
- Environmental considerations
- Certification
- Flight testing and prediction
- Reliability

- Packaging and CG
- Performance
- Stability and control
- Maintainability
- Manufacturing
- Operational cost

Out of these modules, not all of them are always addressed during conceptual design. Most of the “softer” aspects of design, such as cost, maintainability, manufacturing and so on are usually approximated as some function of aircraft mass at conceptual level. Thus minimizing mass, which is oftentimes the objective, would usually drive towards minimum cost as well (Certain technologies need to be treated more carefully, such as avionics – where software is expensive and yet does not contribute to mass). Consequently, based on experience, and the evaluation of various concepts and design methods, 9 essential modules where defined and implemented in the GENUS design tool as shown in Figure 2.

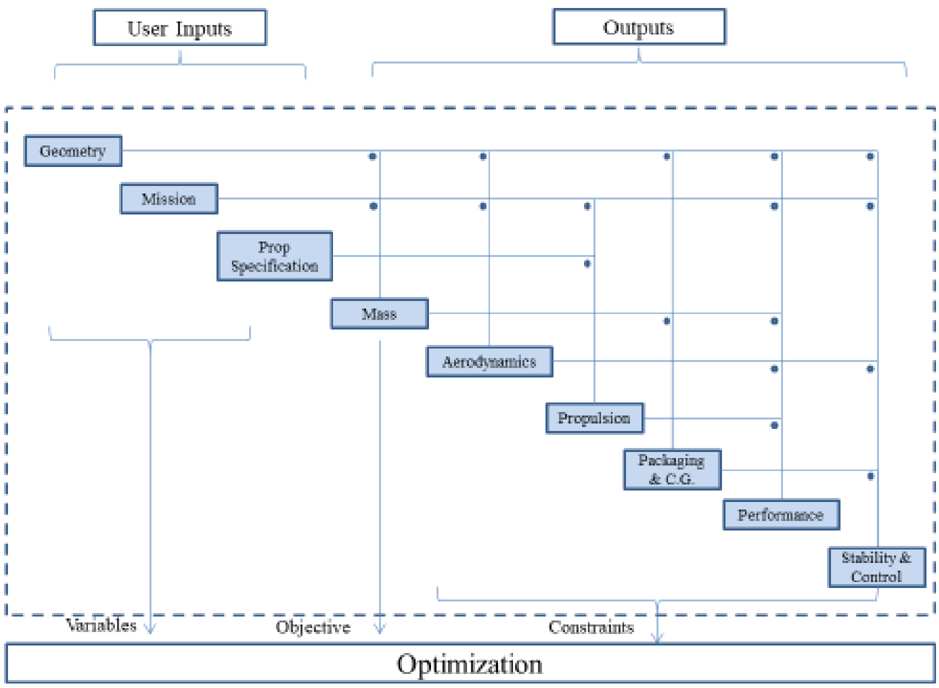

Figure 2: Design program essential structure with example data flow

These modules are the essential elements of any conceptual level aircraft design; and when all these aspects of design are addressed through the application of appropriate constraints, the conceptual design can be deemed complete, self-consistent and fit for purpose.
The *Design program* is required to perform its task, and perform it robustly; generate single instances of design based on any conceivable output. This is essential as modern aircraft design relies on optimization methods to generate feasible, or the “optimum” designs. This is due to the fact that the computerized and human thought processes work in fundamentally different ways. Figure 3 shows the difference between the flow of information in the two thought processes. The human mind relies on its creativity to overcome the challenge of reverse engineering the required inputs

from the desired outputs. This can be done by relying on rigorous, sequential multi-fidelity design processes, but also engineering judgement, “gut feelings”, “rules of thumb”, educated guesses or similar, intuitive approaches. While this comes to us naturally, this is not how a computer behaves, or could behave in the foreseeable future.

Computers excel at performing simple operations repeatedly without error. A current PC or laptop can perform about 177 GFLOPS ($177 \cdot 10^9$ Floating Point Operations per Second), and there seems to be little indication that Moore’s law is slowing down. Thus this increasing computing power should be utilized. To fully utilize a computer’s capacity, it is traditionally used in a design process that is fundamentally different to the manual (or human approach). The method of operation is to assume, synthesize, evaluate, and then iterate by changing initial assumptions until a viable, self-consistent and optimum solution is achieved. Whilst this is, in theory, a sound and logical approach it does carry with it a number of limitations. Optimisation processes tend to lose robustness and the capability to locate the optimum solution (local or global) as the design problem becomes more complex resulting high CPU time requirements and, often, failure to converge. Potential solutions will always be constrained, to a greater or lesser extent, by the design space defined by the software and, where a solution is found, little comprehension as to how the solution was driven to the end point.

The best solution from the whole design process point of view is to create a hybrid of both approaches. Both the creative designer and the power of the computer should be used in a way that they both do what they do best. As such the designer needs to use his experience and knowledge, to steer the computer into the required direction and fully utilize its capabilities. This can be achieved by the designer limiting the number and choice of variables that the optimiser can vary as the design develops. Thus the design might start by exploring the effect of thrust and wing loading, then progress to packaging and stability, prior to allowing a full optimisation to be attempted. Where necessary, the designer has the ability to modify/augment specific modules where required. This enables the knowledgeable user to progressively explore the design space, move close to the optimum, maintain awareness of design issues, apply creative approaches to the design and finish with a design that is viable, (to a greater extent) optimised and well understood. Additionally, whilst this, in common with most design processes, does not guarantee the global optimum (as opposed to a local optimum) it does offer further strength in this direction.

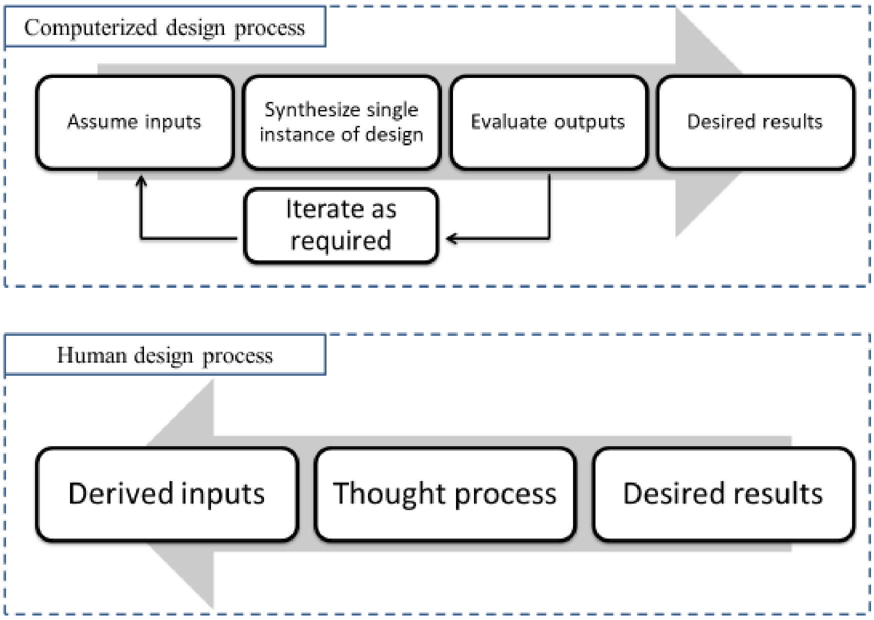

Figure 3: Primary data flow direction in computerized (top) and human design process (bottom)

The GENUS tool was designed to fully utilize this computerized design process. Automatic processes, such as optimization or exploration loops are internal parts of the framework, integrated deep into the flow of data. An example optimization loop can be seen in Figure 4.

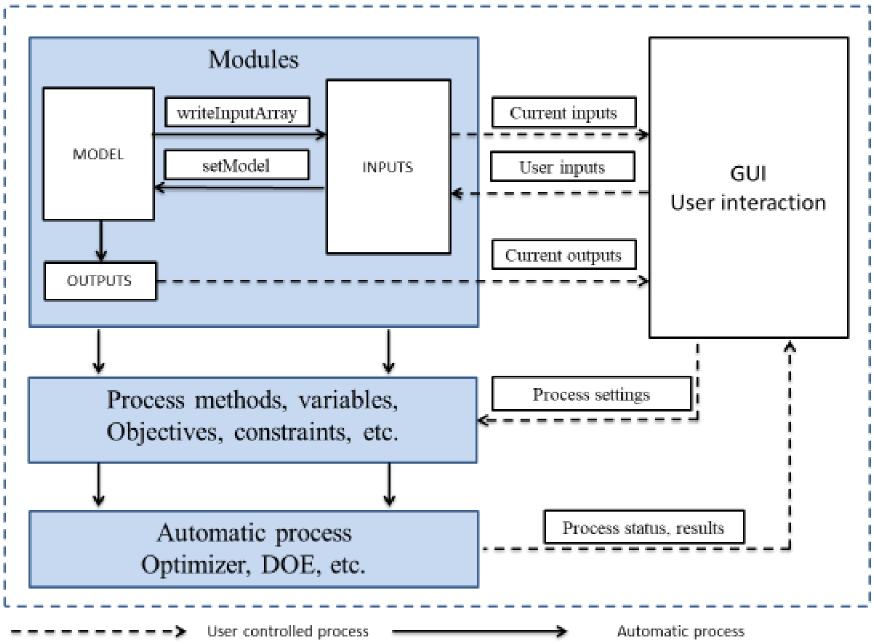

Figure 4: GENUS data flow

The program architecture allows the knowledgeable user to dynamically specify any number of inputs and outputs for the chosen automatic process. With this approach, an initial optimization loop can be run with relatively few variables in order to search for feasible solutions in the design space. Subsequently additional degrees of freedom can be freed up or locked, so optimization loop can be run with less or more variables. This ensures that solution processes such as a gradient based optimization starts from a position where it is feasible to come to a converged optimal solution. Also sensitivity effects of individual parameters to the overall solution can be easily investigated by altering the list of inputs or outputs used in the process.

The core of the GENUS design program is written in Java. Java is one of the most popular programming languages available today (based on IEEE top 10 in 2014 57). It's free to use, easy to learn and there is a very large knowledge base available online in the form of forums, discussion boards, official tutorials, and so on. It is also a general purpose programming language that does not depend on other software. It is platform independent, and being popular, there is an active community and it's likely to stay for a long while. As Java runs on a Java Virtual Machine (JVM) due to this extra layer, it cannot match the speed of a classical compiled language, such as Fortran or C, but it is considerably faster than interpreter languages such as Python or Matlab scripts. At conceptual level design the models used are generally simpler and smaller, so the code is not expected to perform very large matrix inversions or similar, thus even with low processing speed many iterations can be achieved in a short space of time.

What Java also offers is object orientation, and the use of inheritance and polymorphism; the GENUS data architecture extensively relies on these features of the language. A last, but key point of Java, is that it's capable of interfacing with many different programming languages. It has an official Java Native Interface (JNI) to interface with C programs, which then can be used as a wrapper to interface with many other languages, such legacy codes written in Fortran (to use legacy aerospace design codes). In addition there are libraries available to directly communicate with other popular codes, such as Python or Matlab.

There are 2 other key elements of engineering computing, that were considered during the design of the GENUS environment. Since each individual researcher is focusing on a specific class of aerospace vehicle, there are many fields of knowledge that need to be evaluated at potentially different fidelity level. Although it is the design aim of the GENUS environment, to develop common tools, this inevitably still leads to a large suite of numerical codes addressing different aspects of a concept, ranging from simple calculations, to complex, full vehicle flow simulations. Taking into account the level of detail appropriate to and the exploratory nature of conceptual level aircraft design, it is imperative that every part of the design code performs, without unnecessary calculations or activities. For example, writing and reading data from/to a conventional hard disk at random locations, is about 4 magnitudes slower than accessing the computer's memory directly.58 Thus by avoiding writing to disk, and modifying existing codes, not to access the disk, significant increases in performance can be achieved.

The final, but perhaps most important numerical element is the use of optimizers, and other design driver processes. Optimization is a key, integrated part of aircraft design; many optimization methods were actually developed to support aircraft design activities. Also, serious aircraft design activity today is not feasible without these tools, due to the complexity of the task at hand; larger, lighter, more efficient and greener aircraft than ever before.

Most of the optimization problems faced in aircraft design are inherently multidisciplinary, multivariable, and many times multi-objective. While at conceptual level design, minimising mass is often the objective of choice, there are many, often conflicting requirements the designer has to satisfy. Design driver tools, such as the various optimization methods, DoE (Design of Experiments), Pareto analysis, Monte-Carlo simulation, or robust design methods such as Taguchi and Six-sigma analysis are all well known, documented and ready tools for today's designers.

Perhaps the most often used and most widely known design driver is the optimisation process. The classic optimization procedure is designed to minimise (or maximise) objective variables by changing input variables, while adhering to set constraints. Formally, it can be written as:

- Minimize: $f(\bar{x})$
- Subject to:
    - $g_j(\bar{x}) \leq 0 \; for \; j = 1,2, \dots J$
    - $h_k(\bar{x}) = 0 \; for \; k = 1,2, \dots K$
    - $x_i^{(L)} \leq x_i \leq x_i^{(U)} \;\; for \; i = 1,2, \dots N$

Where:

- $\bar{x}$ is the array of optimization input variables, dimension of N
- $f(\bar{x})$ is the objective function, scalar (for multi objective optimization it is an array of dimension M)
- $g_j(\bar{x})$ are the inequality constraints, J in total
- $h_k(\bar{x})$ are the equality constraints, in practical terms $|h_k(\bar{x})| < \varepsilon$, where ε is a small positive number, K in total
- $x_i^{(L)}$ and $x_i^{(U)}$ are the respective lower and upper bounds for variables. Depending on the process, the variables can be unbounded

There are different families of optimizers available today, according to Raymer 59 and Hammond 60. The most commonly used in aerospace vehicle design are the following:

- Calculus based: gradient, finite difference, Lagrange multiplier, sequential quadratic programming, general reduced gradient, implicit function theorem, etc.
- Response surfaces
- Expert systems
- Evolutionary algorithms
- Simulated annealing
- Neural networks
- Monte-Carlo
- Hybrid methods

As it can be seen, there are many different methods available, and a designer intending to utilise the optimisation procedure to its fullest has to be aware of the strengths and weaknesses of each procedure. As Wolpert and Macready 61 famously described, there is "no-free-lunch"; an algorithm that outperforms another solving one type of problem will have inferior performance solving a significantly different problem. Methods such as a gradient based optimizer are very good at efficiently homing in on local maxima for example. As opposed to a multi-island generic algorithm is more exploratory in nature; it is more likely that it will find a global maximum as opposed to getting stuck at local extrema, and it will handle discontinuous design spaces better, but is not as quick or efficient to converge on the solution, as the calculus based gradient method.

The initial version of the GENUS design environment had 2 optimizers integrated; a calculus based, and an evolutionary method. The calculus based is the LSGRG2, which is a Large-Scale Generalized Reduced Gradient method. It is a very robust and well performing optimizer, it is also probably the most widespread, as it is the default optimisation process used by Microsoft Excel's Solver plug-in. The second optimizer, PDGenetic was written by Sziroczák 54, to provide a tool that can be tuned to mitigate the LSGRG2's known weaknesses, such as exploring the design space and finding a feasible initial configuration, even in discontinuous design spaces. Subsequent additions to the software include additional widely used execution options, such as DOE, and potential future aircraft design methods.

3. Application examples

The basic functionalities of the GENUS Aircraft Design Environment were validated by performing conceptual level designs of conventional and unconventional aircraft configurations. Example conventional aircraft include the A320 and Boeing 737 family, and similar classic turbofan driven airliners. Some aspects of this validation process were already presented at conferences 62.

Okonkwo 53 has developed a conceptual design methodology appropriate for Blended Wing Body (BWB) aircraft configurations. The BWB aircraft concept arose from the desire to create an environmentally friendly aircraft that is aerodynamically efficient and capable of conveying large numbers of passengers over long ranges at a reduced direct operating cost. The design offers immense aerodynamics and environmental benefits and is suitable for the integration of advanced systems and concepts such as laminar flow technology, jet flaps and distributed propulsion amongst others. However, despite these benefits, the BWB is yet to be developed for commercial air transport. This is due to several challenges resulting from the highly integrated nature of the configuration, the attendant inter-disciplinary couplings as well as the absence of well-developed design tools incorporating physics based disciplinary models, and other design issues 63.

The GENUS BWB framework integrates a deterministic geometry sizing model, BWB–tailored mass breakdown models and a physics-based Athena Vortex Lattice aerodynamics analysis tool to create an enhanced conceptual design methodology that enables the synthesis and exploration of the design space of this novel class of aircraft. To illustrate the capabilities of the GENUS environment to use a common framework to minimize differences resulting from the applied tools, Figure 5 shows the modules used to perform such design studies. It can be seen that the only BWB specific methods are in the mass breakdown and stability and control modules; the rest are reusable between conventional and BWB configurations.

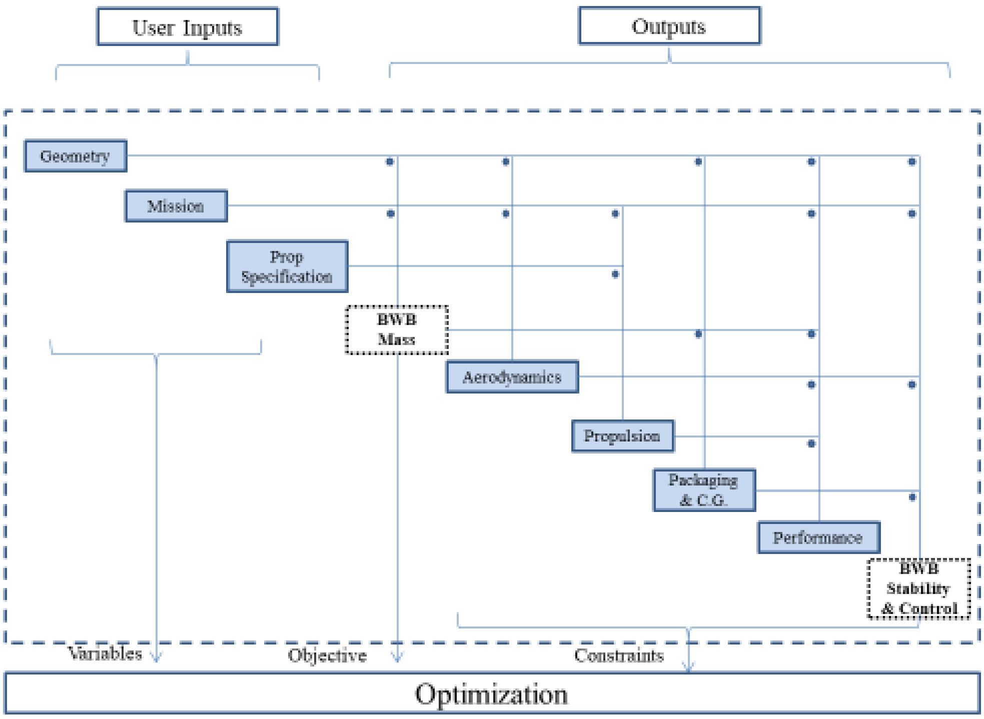

Figure 5: BWB Design essential modules - BWB specific modules highlighted

Other highlights of the developed tool includes the implementation of a performance model using real time theoretically derived estimation of masses at different phases of flight; the integration of a Class Shape Transform packaging module 64 that ensures all items are well enclosed within the geometry to prevent expensive redesign later in the preliminary design stage; and a stability module encompassing longitudinal static stability, trim, turbulence sensitivity and a structural parameter. Trim and turbulence sensitivity are introduced to improve handling quality and enhance passenger's comfort and ride quality. The structural parameter is provided to resolve any potential conflicts between the aerodynamic and structural requirements in the design.

To demonstrate the capability to handle different classes of aircraft in the same environment using a consistent set of tools, a conventional airliner configuration was compared to the Cranfield FW12 flying wing concept. The requirements for both concept were the same, as specified in Table 1.

Table 1: Conventional and FW12 configuration design requirements

Requirement	Value
Range at full payload	12000 km
Cruise Mach number	0.82
Cruise Altitude	10.7 km
Maximum manoeuvre normal load factor, n_z	2.5
Number of passengers	248, 1 class configuration
Passenger and baggage mass	83 + 30 kg

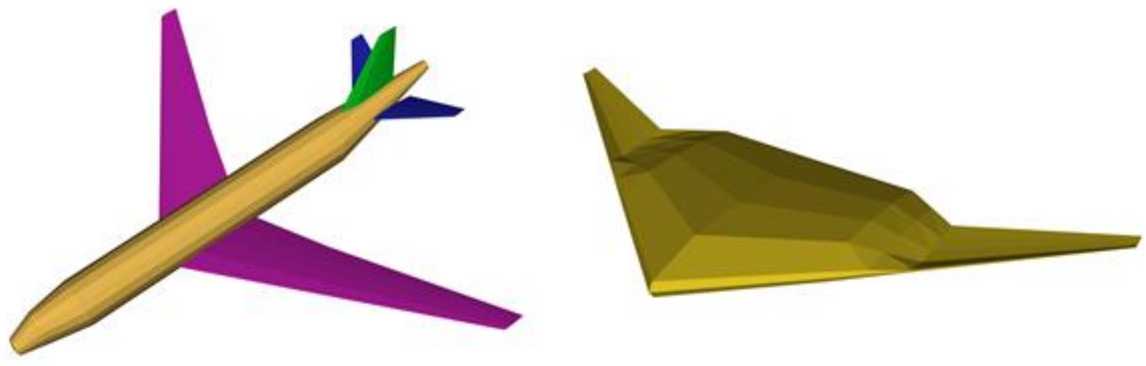

Figure 6: Conventional airliner (left), Cranfield FW12 (right)

Figure 6 shows the graphical representation of the two concepts, generated by the GENUS design environment. The mass breakdown of the two concepts, shown in Figure 7, demonstrates the benefit of the flying wing configuration; to perform the same mission, the aircraft MTOM is reduced from 180.8t to 161.0t, a 10.9% decrease.

Figure 7: Conventional airliner and Cranfield FW12 mass breakdown comparison

The main factor for this difference is a significant, 26% reduction in fuel mass. The OEM decrease is only about 2%. Due to the aerodynamic improvements resulting from the large lifting surface, the C_L required to sustain cruise (based on respective S_{gross} areas) is significantly smaller, 0.21308 for the flying wing, compared to 0.52090 for the conventional airliner. Figure 8 shows the comparison of the drag polars of the two configurations. It can be seen that to sustain cruise CD is larger for the conventional configuration, 0.02962 compared to 0.00957 for the FW12.

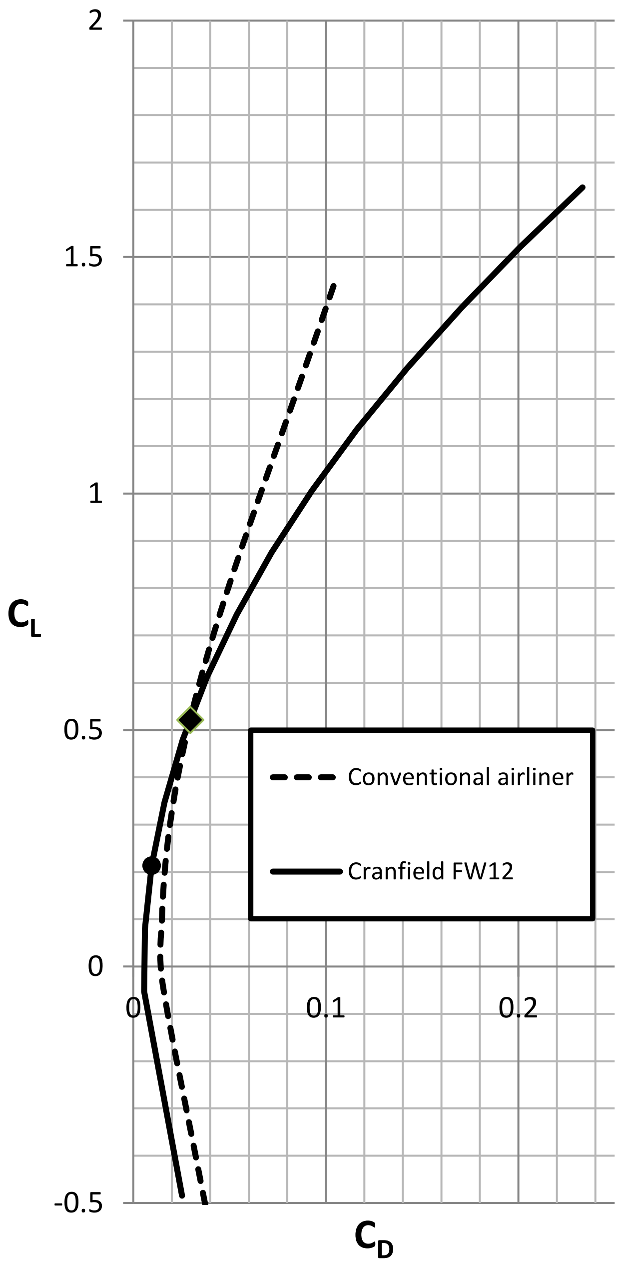

Figure 8: Conventional airliner and Cranfield FW12 drag polar comparison at Mach 0.82

This reduced drag results in significantly reduced fuel consumption. Figure 9 shows that the achievable range of the flying wing aircraft can be extended significantly compared to the similar conventional configuration.

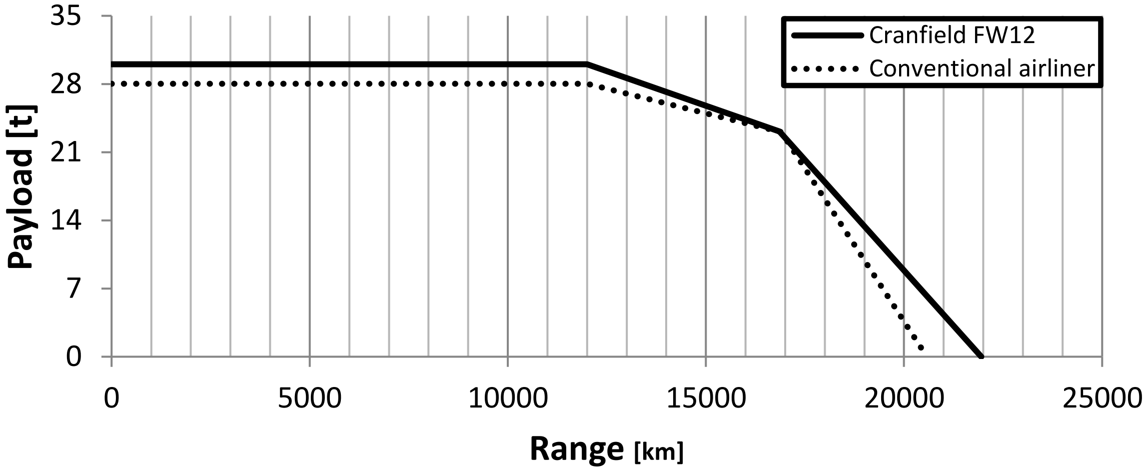

Figure 9: Conventional airliner and Cranfield FW12 payload-range comparison

The short summary of the conceptual level comparison of a conventional airliner and the FW12 concept shows that the GENUS environment can generate conceptual level results for aircraft design using the same framework and as many common tools as possible.

The geometrical representation of concept aircraft is an important element of the conceptual design process. The examples shown here were performed using a simplified representation of aircraft geometry, to enable robust methods and runs thus facilitating the more complete exploration of the design space of the various aircraft. 51 describes some of the issues with this approach, relating to fuselage, nacelle and simple wing geometry generation. The geometry module in GENUS enables the user to define arbitrary aircraft geometry with unlimited number of surfaces and bodies, in arbitrary configuration, and define as many cross-sections as desired. There does not have to be a direct definition for every cross-section for example, each individual module can use its own rule- or knowledge-based geometry definition, such as the Class Shape Transform already mentioned. But regardless of how the user defines it, the geometry is stored in the common format for the rest of the program to use. From the common stored geometry format, GENUS is capable of outputting various geometry formats, such as simple surfaces for external and internal components visualised in VTK, or the NASA LAWGS format, CATIA surface parts, Digital DATCOM equivalent inputs or PANAIR input mesh format, and so on.

An ultra-long endurance solar powered UAV, including perpetual flight capable aircraft design methodology was researched by Abbe 52. The continued concerns and interest over rapid climate changes and global energy security has spurred scientific efforts towards alternative energy technology. In the aerospace community, this has involved research into alternative energy sources, propulsion solutions and perpetual flight missions; the basic aim has not only been the environmental benefits, but also the capability of expanding the mission range and duration optimally. Research and development of high altitude long endurance, and even perpetual flight aircraft systems have become a significant theme in the search for low cost communications and surveillance solutions.

The design space for solar-powered aircraft has evolved over the years, rendering it the subject of several design studies. Since the first solar powered aircraft attempt in 1974, advancements in the capacity and efficiency of energy storage technologies, solar cells, electric motors, and sensors, as well as improvements in robust light weight structural material technologies have further expanded the possible applications of this class of aircraft.
Solar-powered aircraft design is a combination of several multi-disciplinary aspects which include aircraft structural design, propulsion system design, electrical system design, and power and control system design. Key points of the research activity and their critical appraisal are given by Abbe and Smith 65, and their utilization in conceptual level aircraft design.
An extensive study was performed on the design mathematical modelling of the subparts, generating a large database of empirical information. Equations derived from the multivariate regression analyses of this data were then used in the design syntheses to perform a minimum power flight performance analysis, hence obtaining a vehicle weight under minimum energy conditions. A relationship between the calculated parameters was then derived.

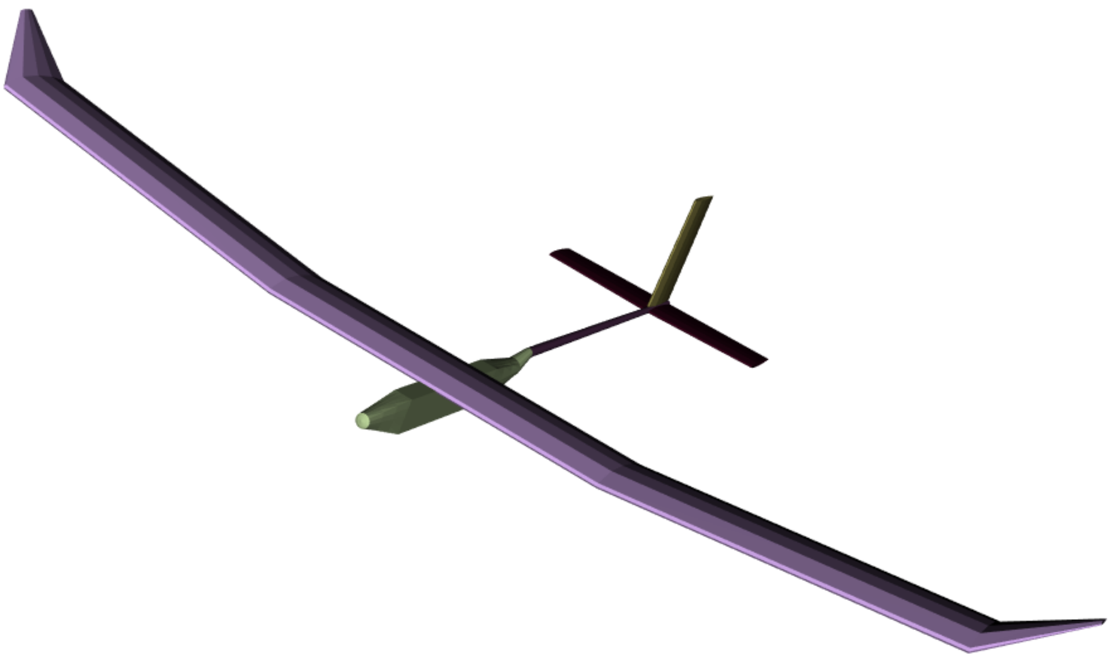

Figure 10: **Solar powered small scale UAV geometry in GENUS 51**

The following example shows the use of the GENUS environment to generate a conceptual level design of a small-scale Solar powered UAV, as shown in Figure 10. The aircraft design is based on Noth's Sky-sailor concept65. Table 2 shows the requirements that were used as the input for the design process.

Table 2: Solar powered UAV design requirements

Requirement	Value
Endurance with full payload	Perpetual

Cruise speed	8.03 m/s
Day of year, Latitude, Altitude	172th day, 45º, 3km
Maximum manoeuvre normal load factor, n_z	3.8
Payload mass	0.250 kg
Payload power consumption	0.5 W

The mass breakdown generated using the methodology in GENUS was compared to Noth's method and the measured mass values from the Sky-sailor. Figure 11 shows the comparison, demonstrating the excellent agreement between the different values.

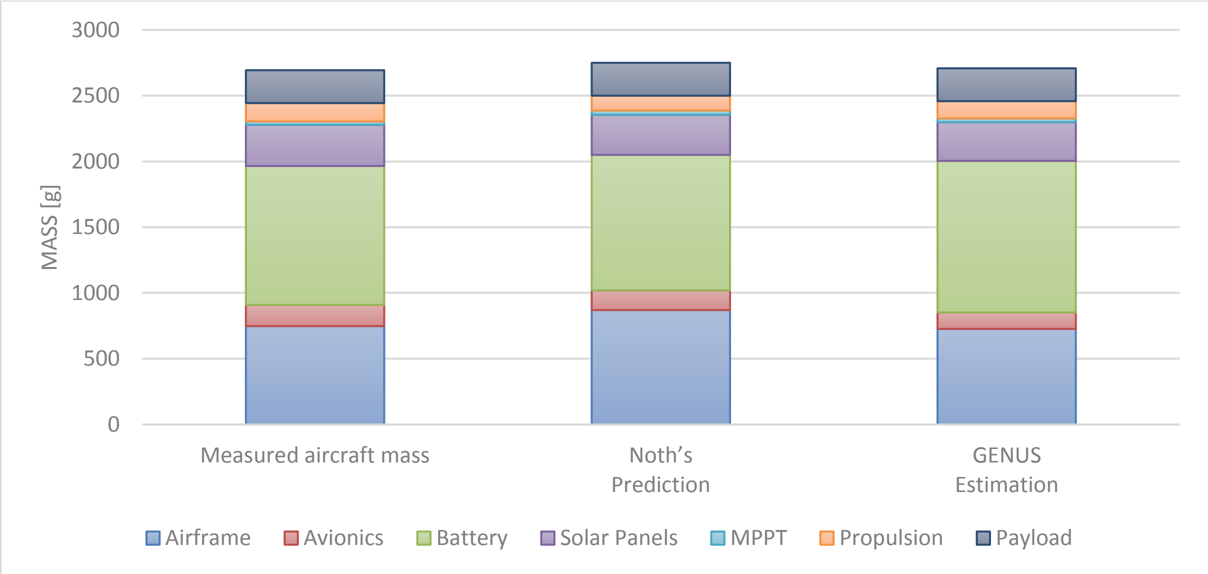

Figure 11: Solar powered UAV mass breakdown comparison

The drag polar of the solar powered UAV is shown in Figure 12. It should be noted that the graphs were generated at significantly different flight conditions; typical cruising speed of an airliner is about Mach 0.82, while a typical solar UAV cruises at Mach 0.025.

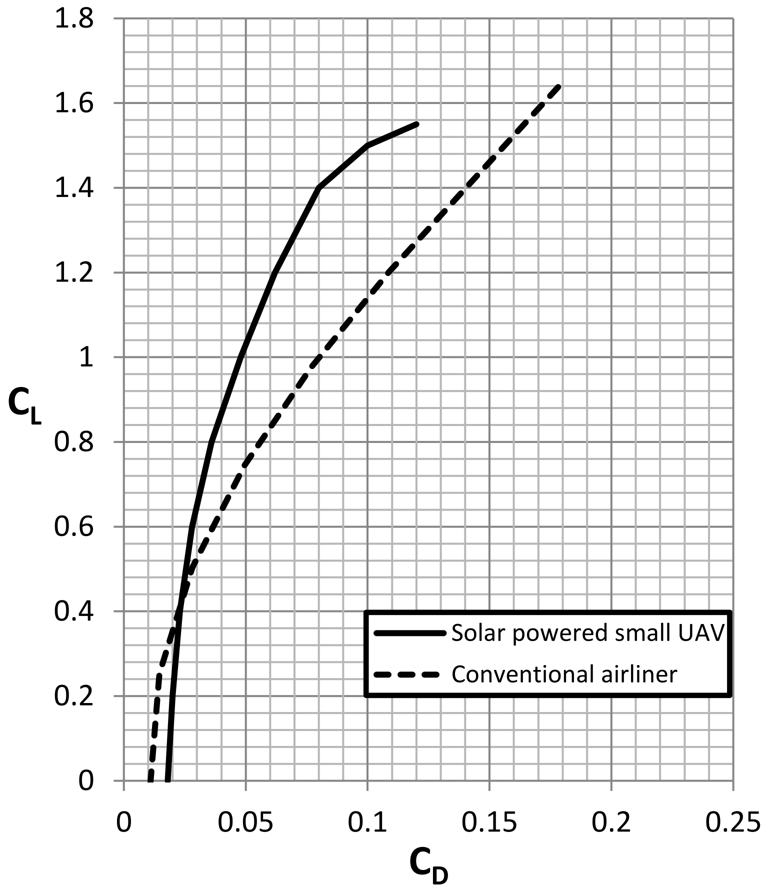

Figure 12: Drag polar of solar UAV compared to conventional airliner

Sziroczák54 investigated aerospace vehicles capable of operating in the hypersonic speed regime; Space Launchers and Hypersonic Transports. The idea of hypersonic flight has been around since the late 1930s, and there has been varying, but constant interest ever since. Both classes of aircraft provide capabilities beyond the state of the art; Hypersonic Transports drastically reduce travelling times, which is a critical feature in applications such as emergency response, military applications or important business trips. Advanced Space Launchers on the other hand have the capability to reduce the cost and increase the reliability of contemporary space access and return capability. Also, they enable repeated space access due to reduced turnaround times, which enables for example rapid deployment of satellites; a critical strategic advantage. Despite these substantial benefits, there are still a range of design issues to be addressed until hypersonic flight becomes more widespread. 67
The developed methodology enables the user to generate designs of both Space Launchers and Hypersonic Transports within the GENUS Aircraft Design Environment. The full range of common powerplant choices are available in the tool; turbojets, ramjets, scramjets, liquid rocket engines, with a range of fuels and fuel/oxidizer combinations. Aerodynamic estimation methods ranging from the subsonic regime up to high hypersonic (Mach 25+) speeds are available. The methodology relies on both empirical, such as Digital Datcom, and computational methods, such as the Supersonic/Hypersonic Arbitrary Body Program. 68 All essential design areas identified at conceptual level are addressed in the research

work. Figure 13 shows example hypersonic vehicles, the Space Shuttle and the Skylon concept, analysed using the GENUS environment.

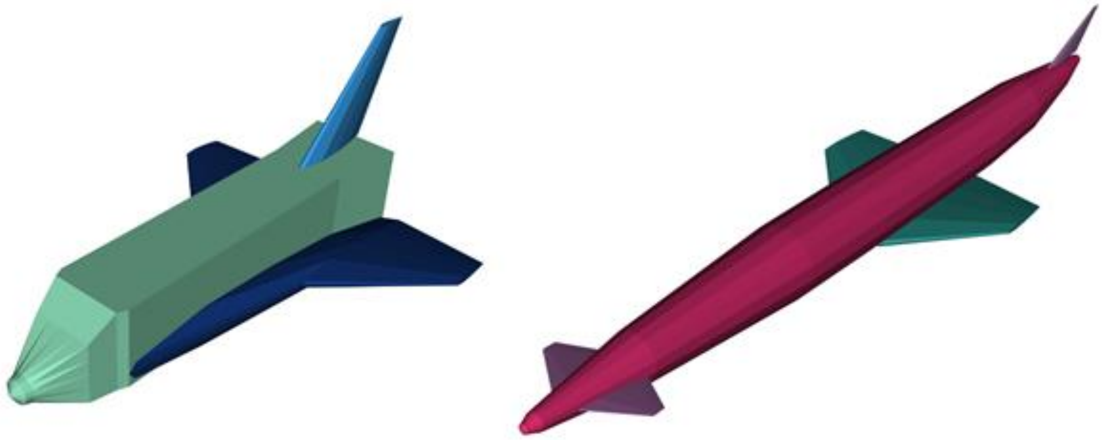

Figure 13: Example hypersonic concept vehicles generated from GENUS; Left: Space Shuttle simplified concept, Right: Skylon fuselage without engines 54

Table 3 shows an example list of requirements that were used to generate a SSTO Space Launcher concept. Compared to more conventional configurations the critical question, in the case of a space launcher, is not necessary optimum, but feasibility.

Table 3: Hypersonic space launcher design requirements

Requirement	Value
Design mission duration	5 days
Orbital ΔV capacity	300 m/s
Target orbit	400 km, circular, 0º inclination
Maximum manoeuvre load factor normal/axial, n_z / n_x	2 / 7
Design payload	3000 kg
Launch site	Equator, 0º Lattitude

An advanced technology Space Shuttle derivative space launcher vehicle with double delta wings, single vertical tail and approximate fuselage was designed using the GENUS methodology. Apart from the specific mass breakdown estimation and the complex performance calculations, the conceptual design was performed using the same 7 essential GENUS modules that were used in the case of a conventional airliner concept. The mass breakdown of the space launcher vehicle is shown in Figure 14. Note that for clarity, propellant is not included in the mass breakdown. This class of aerospace vehicle requires very high fuel fractions, for this specific concept;84.2%, so mass breakdown includes an additional 28t LH2 and 140t LOX.

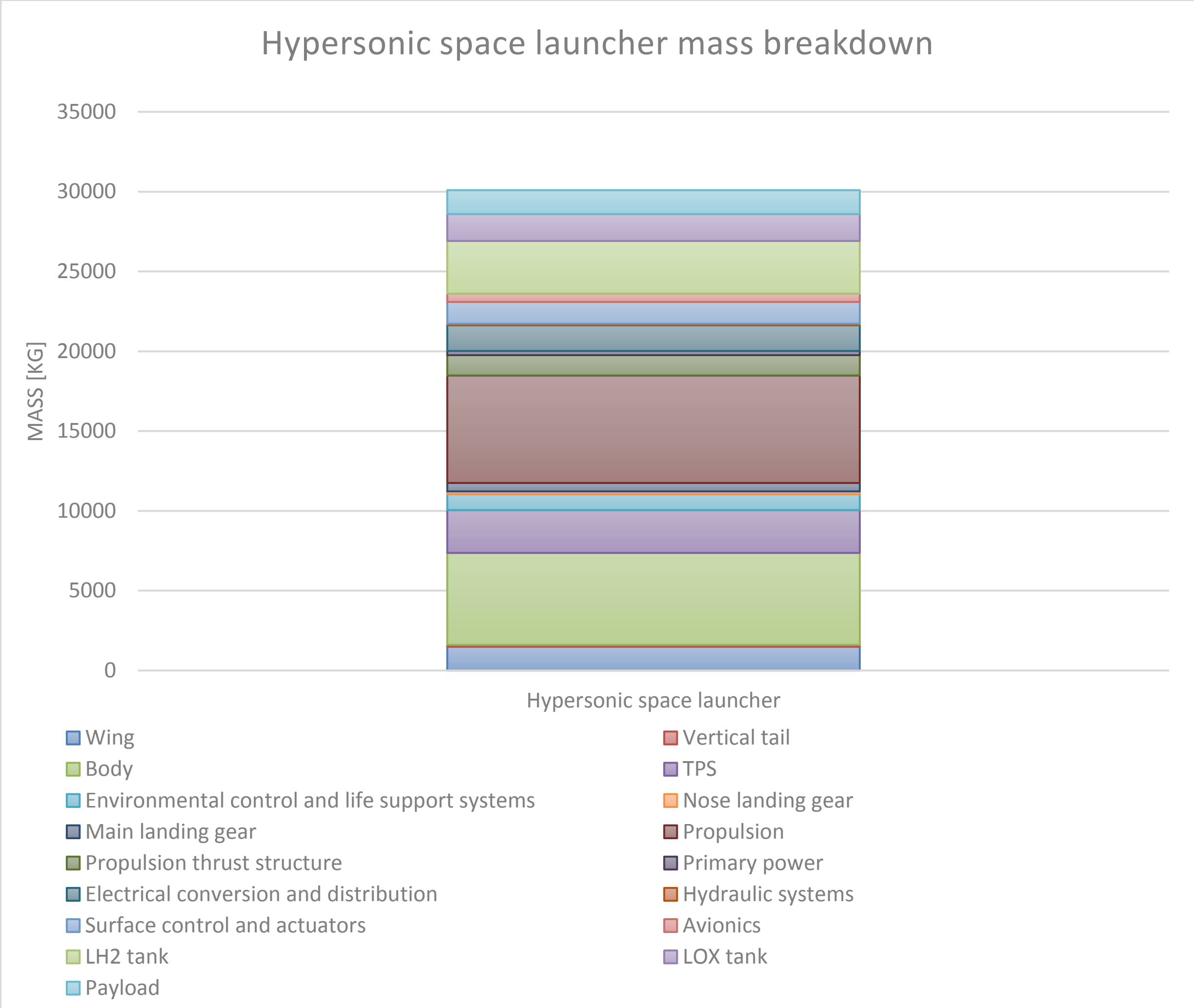

Figure 14: Hypersonic space launcher mass breakdown, excluding propellant

When aerodynamic performance is estimated, hypersonic vehicles need to be evaluated in a wider operating range than conventional configurations. Typical operating speeds of a space launcher, for example the Space Shuttle range from about Mach 0.2 up to Mach 25 and potentially above. This requires methodologies to estimate subsonic, transonic, supersonic and hypersonic aerodynamic performance. In the GENUS environment, a common framework is used so that the different tools can produce consistent results when evaluating a single concept, and that multiple concepts can be consistently analysed with the same set of tools also consistently. Figure 15 shows a comparison of the concept space launcher drag polars at various Mach numbers as generated in GENUS. An example drag polar for a conventional airliner is added to the graph for reference.

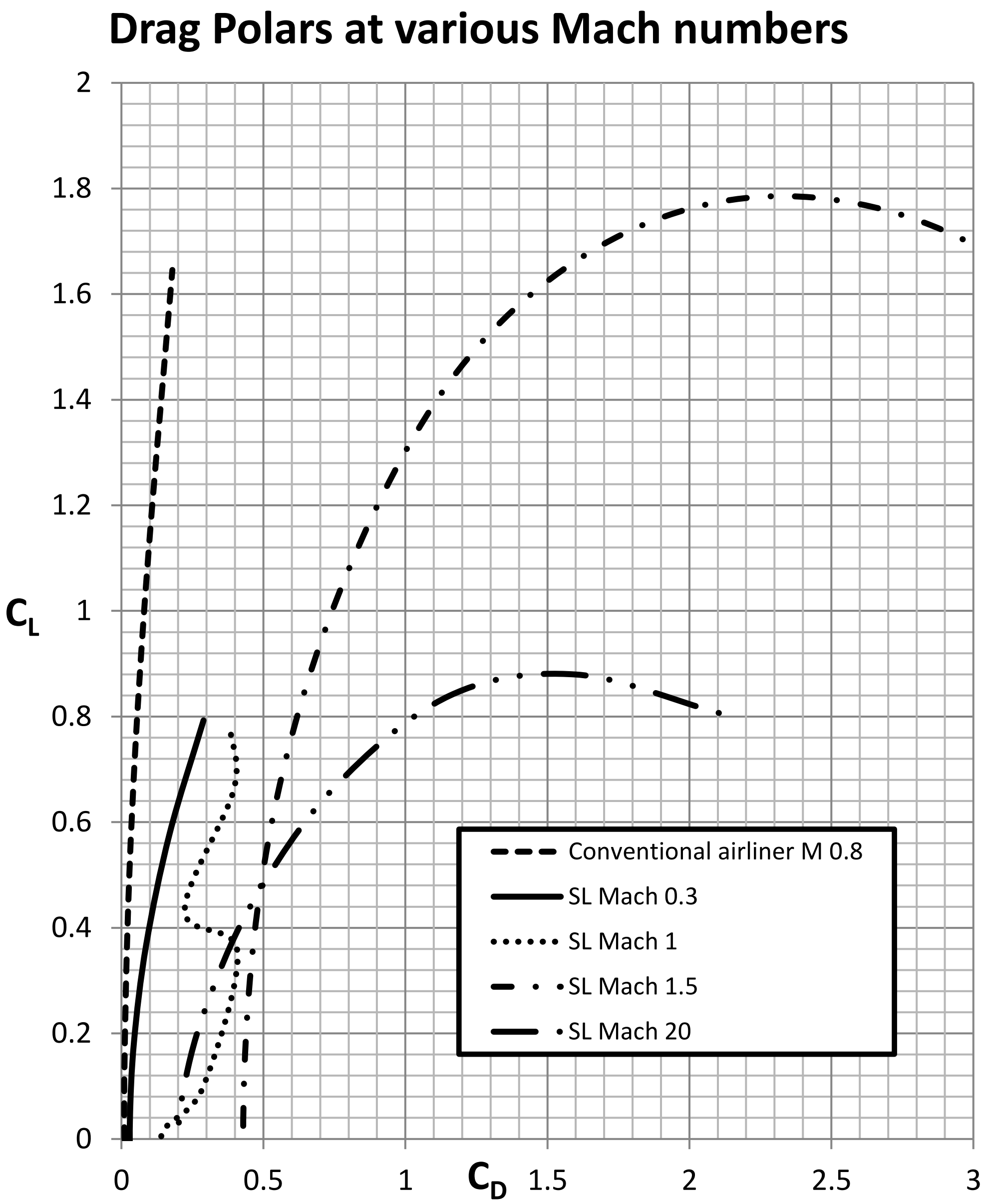

Figure 15: Drag polar comparison, conventional airliner and space launcher (SL) at various Mach numbers

Ongoing development is currently adding further capability to the GENUS toolset. This work currently included stealth characteristics and sonic boom modelling. Whilst the methods currently include static stability may explore dynamic stability, as discussed in 69 and the incorporation of more high fidelity tools such as FE analysis as discussed in 70.

4. Conclusion

A new approach to air vehicle design synthesis is presented that aims to better support the knowledgeable aircraft designer and gets the best performance from both user and computer. The GENUS Aircraft Design Environment is an implementation of this new approach to the development of the tools used to perform conceptual level aircraft design. Methodologies appropriate to contemporary, or advanced aircraft concepts can be implemented into the Design Environment. The study has not only focused on the discipline specific modules but the overall architecture that permits the

knowledgeable user to interact with it, control it and further develop it to handle additional concepts and variable fidelity models. The modular approach also ensures that development effort is effectively used by avoiding the need to repeatedly create common elements.
In addition to the analysis of individual classes of vehicle, the capability will better enable comparisons between conventional and advanced concepts and configurations to be made.
The design environment has been demonstrated to work for a range of concepts and its capability continues to be expanded.

5. References

1. Howe D. *Aircraft conceptual design synthesis*, Professional Engineering Publishing, London, 2000
2. Raymer D. P. Aircraft *design: a conceptual approach 4th edition*, American Institute of Aeronautics and Astronautics, 2006
3. Roskam J. *Airplane Design 2nd edition*, Darcorporation, December 31, 2003
4. Torenbeek E. *Synthesis of subsonic airplane design*, Delft University Press, 1976
5. Stinton D. The Design of the Aeroplane 2nd edition, Blackwell Publishing, 2001
6. Corke T. C. Design of Aircraft, 1st edition, Pearson, 1 November 2002
7. Jenkinson L. R. *Civil Jet Aircraft Design* AIAA/Butterworth-Heinemann, 1999
8. Crafword D. R. *A Practical Guide to Airplane Performance and Design,* Crawford Aviation, Third Edition, 1981
9. Fielding J. P. *Introduction to Aircraft Design,* Cambridge University Press, 1999
10. Robinson J. S., Martin J. G. *An overview of NASA's Integrated Design and Engineering Analysis (IDEA) Environment*, Joint Army-Navy-NASA-Air Force (JANNAF) 6th Modeling and Simulation / 4th Liquid Propulsion / 3rd Spacecraft Propulsion Joint Subcommittee Meeting, Orlando, Florida, December 8-12, 2008
11. Wallace R. E. *A computerized system for the preliminary design of commercial airplanes,* AIAA paper 72-793, August 1972
12. Robinson J. S. *An Overview of NASA's Integrated Design and Engineering Analysis (IDEA) Environment,* AIAA 2011-2392, 17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, San Francisco, California, April 11-14, 2011
13. Monell D., Matihas D., Reuther J. *Multi-Disciplinary Analysis for Future Launch Systems Using NASA's Advanced Engineering Environment (AEE)*, 16th AIAA Computational Fluid Dynamics Conference, Orland, Florida, June 23-26, 2003
14. Lissys Ltd, Piano aero, source: http://www.piano.aero/ last accessed: 28 September 2017
15. McCullers, L.A.: Aircraft Configuration Optimization Including Optimized Flight Profiles, Multidisciplinary Analysis and Optimization Part 1. NASA CP-2327, 1984.
16. Lavelle T M., and. Curlett B P., Graphical User Interface for the NASA FLOPS Aircraft Performance and Sizing Code, NASA TM-I06649, Lewis Research Center, Oct. 1994.
17. Litherland B., Rieth K., VSP Aircraft Analysis User Manual: Modeling and Analyzing Aircraft Designs Using Parametric Geometry Tools and Vortex Lattice Software , NASA July 24th, 2014

18. Laughlin T., STRUCTURAL MODELING AND OPENVSP, OpenVSP Workshop 2015, Hampton, Virginia, August 11-13, 2015
19. PACE GmbH, “PACELAB APD 3.1 User Documentation”, PACELAB User Documentation, 2013.
20. Riggins B K., Development of a Multi-disciplinary Design Optimization Framework for a Strut-Braced Wing Transport Aircraft in PACELAB APD 3.1, MSc Thesis, Virginia Polytechnic Institute and State University, 2/23/2015.
21. “Pacelab SysArc, PACE Aerospace Engineering and Information Technology GmbH,” Online: http://www.pace.de/products/preliminary-design/pacelab-sysarc.html.
22. Chakraborty I.. Mavris D N., A System and Mission Level Analysis of Electrically Actuated Flight Control Surfaces using Pacelab SysArc, AIAA SciTech Forum, 52nd Aerospace Sciences Meeting, 13-17 January 2014, National Harbor, Maryland
23. Risse K., Eckhard A., Lammering T., Franz K., Hoernschemeyer R. An Integrated Environment for Preliminary Aircraft Design and Optimization, 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, Hawaii, 23-26 April, 2012
24. United States Air Force, *USAF Stability and Control Data Compendium, Digital Datcom,* computer program, source: http://www.pdas.com/datcom.html last accessed: July 20, 2012
25. DARcorporation, Advanced Aircraft Analysis, computer program, source: http://www.darcorp.com/Software/AAA/ last accessed: July 20, 2012
26. Raymer D. RDS: A PC-Based Aircraft Design, Sizing and Performance System, AIAA Paper 92-4226, August 1992
27. CEASIOM, Computerised Environment for Aircraft Synthesis and Integrated Optimisation Methods, source: http://www.ceasiom.com/index.php last accessed: July 21, 2012
28. Rizzi A., Modeling and Simulating Aircraft Stability & Control – the SimSAC Project, AIAA 2010-8238, AIAA Atmospheric Flight Mechanics Conference, 2-5 August 2010, Toronto, Ontario, Canada.
29. Tomac M., Rizzi A., Nangia R K., Mendenhall M R., Perkins S C., Engineering Methods on the SACCON configuration – Some Design Considerations, AIAA 2010-4398, 28th AIAA Applied Aerodynamics Conference, 28th Jume-1 July 2010, Chicago, Illinois.
30. Eliasson P., Vos J B., Da Ronch A., Zhang M., Rizzi A., Virtual Aircraft Design of TransCruiser – Computing Break Points in Pitch Moment Curve, AIAA 2010-4366, 28th AIAA Applied Aerodynamics Conference, 28th Jume-1 July 2010, Chicago, Illinois,
31. Khrabrov A., Sidoryuk M., Non-Symmetrical General Aviation Aircraft and its Flight Control Law Design Using CEASIOM Software, AIAA 2010-284, 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Aerospace Sciences Meetings, 4-7 January 2010, Orlando, Florida.
32. Rizzi, Zhang, M., Nage B., Boehnke D. and Saquet P. Towards a Unified Framework using CPACS for Geometry Management in Aircraft Design, AIAA 2012-0549, 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 09 - 12 January 2012, Nashville, Tennessee
33. Bohnke D., Nagel B., Zhang M., and Rizzi A., Towards a Collaborative and Integrated Set of Open Tools for Aircraft Design, AIAA 2013-0222, 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition 07 - 10 January 2013, Grapevine (Dallas/Ft. Worth Region), Texas

34. Zhang M., Rizzi A., Nicolosi F., De Marco A., Collaborative Aircraft Design Methodology using ADAS Linked to CEASIOM, 32nd AIAA Applied Aerodynamics Conference, 16-20 June 2014, Atlanta, GA AIAA AVIATION Forum
35. Goetzendorf-Grabowski T., Multi-disciplinary optimization in aeronautical engineering, Proc IMechE Part G: J Aerospace Engineering 2017, Vol. 231(12) 2305–2313
36. Desktop Aeronautics, Program for Aircraft Synthesis Studies (PASS), source: http://www.desktop.aero/about.php last accessed: July 21, 2012
37. Ciampa P D., Nagel B., The AGILE Paradigm: the next generation of collaborative MDO, 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA AVIATION Forum, (AIAA 2017-4137)
38. van Gent I., Ciampa P D., Aigner B., Jepsen J., La Rocca G., Schut J., Knowledge architecture supporting collaborative MDO in the AGILE paradigm, 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 5-9 June 2017, Denver, Colorado, USA
39. Lefebvre, T., Bartoli, N., Dubreuil, S., Lombardi, R., Panzeri, M., Lammen, W., Zhang, M., van Gent, I., Ciampa, P. D. *Overview Of MDO Enhancement In The AGILE Project: A Clustered And Surrogate-Based MDA Use Case,* Aerospace Europe 6th CEAS Conference, 2017
40. Harloff G. J., Berkowitz B. M. HASA – Hypersonic Aerospace Sizing Analysis for the Preliminary Design of Aerospace Vehicles, Lewis Research Center, November 1988
41. Chudoba B. Stability and Control of Conventional and Unconventional Aircraft Configurations: A Generic Approach, Books of Demand, 2001
42. Djafri D. Y. Multi-disciplinary Design of Wings for Transport Aircraft Operating at High Subsonic Speed, Doctoral Thesis, Cranfield University, College of Aeronautics, 1995
43. Siegers F. Conceptual design synthesis and optimization for new generations of combat aircraft, Doctoral Thesis, Cranfield University, March 1996
44. Woodford S. The Minimisation of Combat Aircraft Life Cycle Cost through Conceptual Design Optimisation, Doctoral Thesis, Cranfield College of Aeronautics, May 1999
45. Watjatrukul B. Effect of aerodynamic constraints on the design and optimisation of UCAV concepts, Doctoral Thesis, Cranfield University, 2002
46. Niyomthai N. Packaging and configuration design aspects of UCAV concept synthesis and optimization, Cranfield University, 2002
47. Rajendran P. Investigation into the Design Space associated with Electrically Powered Flight Vehicles, Doctoral Thesis, Cranfield University, 2012
48. Chudoba B. Development of a generic stability and control methodology for the conceptual design of conventional and unconventional aircraft configurations, Doctoral Thesis, Cranfield University, 2001
49. Nagel, B., Böhnke, D., Gollnick, V., Schmollgruber, P., Rizzi, A., La Rocca, G., Alonso, J. J.; 2012; „Communication in Aircraft Design: Can We Establish a Common Language?"; 28th Congress of the International Council of the Aeronautical Sciences (ICAS); Brisbane, Australia, 23-28 September
50. Walther J N., Ciampa P D., Knowledge-based automatic Airframe Design using CPACS, 6th CEAS AIR & SPACE CONFERENCE AEROSPACE EUROPE 2017, CEAS 2017, 16-20October 2017, Bucharest, Romania
51. Mieloszyk J., Goetzendorf-Grabowski T., Mieszalski D., Rapid geometry definition for multidisciplinary design and analysis of an aircraft, Aviation, Vol. 20, no. 2, June 2016, pp. 60-64, doi:10.3846/16487788.2016.1195066
52. Abbe G. E. Conceptual Design Methodologies for Small Solar Powered Unmanned Aerial Vehicle, PhD Thesis, Aircraft Design Group, Cranfield University, 2015

53. Okonkwo P. P. C. *Conceptual Design Methodology for blended wing body aircraft,* PhD Thesis, Aircraft Design Group, Cranfield University, 2016
54. Sziroczak D. Conceptual Design Methodologies Appropriate to Hypersonic Space and Global Transportation Systems, PhD Thesis, Aircraft Design Group, Cranfield University, 2015
55. Sun Y., Smith H. *Review and prospect of supersonic business jet design,* Progress in Aerospace Sciences, Volume 90, April 2017, Pages 12-38
56. Sepulveda E., Smith H. Technology challenges of stealth unmanned combat aerial vehicles. The Aeronautical Journal. 2017; 121(1243): 1261–1295. Available at: DOI:10.1017/aer.2017.53
57. Cass S. Top 10 Programming Languages, Spectrum's 2014 Ranking, posted on 19th July, 2014 source: http://spectrum.ieee.org/computing/software/top-10-programming-languages
58. Jacobs A. *The Pathologies of Big Data,* ACMQUEUE, Volume 7, Issue 6, July 6, 2009
59. Raymer D. P. Enhancing Aircraft Conceptual Design Using Multidisciplinary Optimization, Doctoral Thesis, Kungliga Tekniska Högskolan, May, 2002
60. Hammond W. E. Design Methodologies for Space Transportation Systems, AIAA Education Series, 2001
61. Wolpert D.H. Macready W.G. *No Free Lunch Theorems for Optimization,* IEEE Transactions on Evolutionary Computation, Vol. 1., No. 1, April 1997
62. Sun Y., Smith H. *Turbofan Airliner Conceptual Design in Multidisciplinary Design Analysis Optimization Environment,* 1st International Conference in Aerospace for Young Scientists, Beijing, People's Republic of China, November 2016
63. Okonkwo P., Smith H. *Review of evolving trends in blended wing body aircraft design,* Progress in Aerospace Sciences, Volume 82, April 2016, Pages 1-23
64. Okonkwo P., Smith H. *Packaging in a Multivariate Conceptual Design Synthesis of a BWB Aircraft,* World Academy of Science, Engineering and Technology, International Journal of Mechanical, Aerospace, Manufacturing, Industrial Science and Engineering, Vol. 8 No. 6 2014
65. Abbe G. E., Smith H. *Technological development trends in Solar-powered Aircraft Systems,* Renewable and Sustainable Energy Reviews, Volume 60, July 2016, Pages 770-783
66. Noth A. *Design of Solar Powered Airplanes for Continuous Flight,* PhD Thesis, ETH Zurich, 2008
67. Sziroczak D., Smith H. *A review of design issues specific to hypersonic flight vehicles,* Progress in Aerospace Sciences 84, July 2016, Pages 1-28
68. Gentry A. E., Smyth D. N., Oliver W. R. *The Mark IV Supersonic-Hypersonic Arbitrary-Body Program*, Air Force Flight Dynamics Laboratory, November 1973
69. Mieloszyk J., Goetzendorf-Grabowski T., Introduction of full flight dynamic stability constraints in aircraft multidisciplinary optimization, Aerospace Science and Technology Volume 68, September 2017, Pages 252-260
70. Goetzendorf-Grabowski T., Mieloszyk J., (2017) "Common computational model for coupling panel method with finite element method", Aircraft Engineering and Aerospace Technology, Vol. 89 Issue: 5, pp.654-662