METHODOLOGY FOR AIRCRAFT SYSTEM ARCHITECTURE SIZING

Cyril de Tenorio, Dimitri Mavris, Elena Garcia, Michael Armstrong
Aerospace System Design Laboratory, Georgia Institute of Technology,

Keywords: *conceptual design, architecture design, functional induction, system sizing, multi-level optimization*

Abstract

This paper introduces a methodology to estimate aircraft performance by sizing the systems constituting its architecture. The proposed approach requires the definition of the architecture structure which is performed using functional induction. This definition allows the formulation of the modeling structure necessary to integrate sizing models at the component level. The sizing of the systems is based on a multi-level optimization method. This approach allows the definition of objective-functions at the component level which are driven by the optimization of performance at the aircraft level.

1. Introduction

Conceptual design is a design phase of great opportunity and danger. During this phase, decisions are seeds which will yield success, technical difficulties or failures. The word conceptual should remind us that the center of focus is on the development of the global concept which will be used to meet some requirements. The terms, "focus" and "global", highlight the challenge faced by conceptual designers and architects. Given the stakes pertaining to this phase, the designers must concentrate their attention on what matters the most, which is the global picture: the aircraft level. On the other hand, the decisions for a system architect are made on local elements (or system level): Should I select a hydraulic or an electro-hydro-static actuator? Should I connect this power load to network A or B? Therefore, decisions must focus on the aircraft level while the level of action remains at the component level. From this situation results the overarching challenge faced by all system architects on how to reconnect their level of action to their level of decision.

The methodology proposed in this paper is part of a research effort undertaken within the Aerospace Systems Design Laboratory (ASDL) to help architects in addressing this challenge. By introducing automated system sizing methods, this paper will present a means to relate the aircraft level objectives to system level decisions.

2. Problem definition

Unlike many industrial products, a commercial aircraft requires a degree of technological sophistication and complexity which differentiates it from most. Because of this, the public perception of aerospace engineers has classified us as people with extraordinary technical capabilities (or at least higher than average). We are, unfortunately, not cleverer than our counter-parts in others industries. As a result, the extraordinary technical challenges offered by an aircraft development are addressed using a sophisticated organizational structure. This organization allows for the subdivision of the complex problem into simple, or at least technically manageable, projects. The more complex the technical problem, the more subdivided it becomes. This subdivision practice is defined, ruled, and managed by systems engineering. As a result, an aircraft development is really a network of developments including engine development, fuselage development, control systems development, cabin development, etc…This breakdown structure allows the engineers to deal with "simpler" problems toward the realization of a technologically complex ensemble. In other words, systems engineering facilitates the design of complex systems by translating them into interdependent manageable designs of “simpler” systems.

From this observation we can see two levels emerging: the complex system level (the aircraft) and the “simpler” system level (pumps, generators, electrical wires, etc…). In this paper the complex system level will be referred to as the “architecture” and the “simpler” system level as “system”.

This system-oriented approach has fostered the emergence of problems often pointed out by research groups. The first problem is the rigidity of architectural concepts. As an organization becomes “comfortable” working around a given architectural framework, changing the structure of an architecture, which implies moving away from this framework, becomes increasingly difficult. The second problem relates to the difficulty in relating system level improvements to architecture optimization. In preliminary phases of design, experience from past developments is applied to the new. Since a new development may imply new technologies which modify the relationships amongst the systems composing the architecture, that approach may become obsolete.

This evolution is clearly illustrated by the emergence of energy or power optimized research programs in all major aerospace organizations. Those programs are the industry expression of their need to evolve from the traditional perspective on system level development which is focused on weight. As advertized by their names (Power Optimized Aircraft, Energy Optimized Aircraft Systems) the orientation of improvement is shifting from weight-reduction toward energy-efficient approaches.

On the academic side, several solutions have been proposed to address technological orientation issues in complex systems settings. References [1] and [2] present examples of the trades that can be used to relate system level technology factors to architecture performance levels and objectives. If we observe the work by Kirby and Biltgen et al., we can see that both are based on the observation of the effect of changing factors at the system-level on the aircraft/architecture level metrics. The Technology Identification Evaluation and Selection method [2] explores technological alternatives by changing system level settings and observing the impact on aircraft level performance. In a similar fashion Biltgen et al. propose probabilistic design exploration of technological settings to observe the distribution (expectancy) of performance. It is to be noticed that all state of the art techniques require an important computational capacity. Their application has, therefore, been limited to problems where automated means of analysis were available to the designer.

The application of the state of the art has thus been limited to exercises where the design space could be covered by a single model with parametric inputs. This type of situation is rare in Aircraft System Architecture (ASA) design. Several factors can explain this fact.

The first factor is the discontinuous structure of the ASA design space. As an architecture changes, the relationships between systems also change. This fact can be illustrated by the comparison of “more-electric” and “bleed” architectures. The bleed architecture refers to the fact that the Air Conditioning (AC) unit is powered by pneumatic energy “bled” from the engine compressor. The “more-electric” concept powers the AC unit through electric energy originating from mechanical transformation on the engine shafts. The system relationships structuring these architectures are different. As a consequence, the sizing of the systems composing the architectures will follow different physical rules and trends. It is therefore very difficult to capture those trends within a single numerical model.

The second factor results from organizational realities. Given the scale and technical complexity of an ASA, the design expertise of its systems is systematically scattered amongst multiple design groups. Each of these groups has its own design and sizing models. Integrated models are rare and difficult to setup.

Consequently, most studies within an industrial setting are based on semi-automated studies. Often within centres of excellence, numerical modelling is used to support internal studies, but the exchange of information at the architecture level remains “manual” (i.e. table sent via email, excel spreadsheets). Those practices are incompatible with state of the art systems engineering methods which draw their conclusions from the recursive sizing and analysis of the different concepts considered.

The goal of this paper is, therefore, to propose a means to integrate system-level numerical analyses within a modelling structure which will estimate the size of the systems and deduce the performance at the aircraft level.

3. Architecture definition through functional induction

In order to size the systems composing an architecture, one must know what the given system is supposed to do within this architecture. Based on this observation, two perspectives are emerging: Functional and physical. In order to know what a system is “supposed to do” one must consider its function. However, the function itself is an action which can be quantified but not sized. The system performing the function is sized, which introduces the physical perspective. Based on this observation, the method of functional induction was proposed by ASDL[3].

The mission definition of an aircraft defines a set of functionalities. To illustrate this approach let us consider two of those functionalities: “propel the airframe” and “actuate the rudder”. The reason why an architecture is required comes from the fact that we do not have one simple solution to satisfy all functionalities. For each function, a physical solution is selected: a turbofan for propulsion and a hydraulic actuator for controlling the rudder. By selecting these solutions (or systems) new functions are induced: provide jet fuel and provide hydraulic energy. For each of those functionalities new systems are selected: fuel tank, and a hydraulic pump. Once again a new function is induced by the pump: provide mechanical energy. We can therefore assume that this functionality will also be fulfilled by the engine. By looking at this functional induction chain, (see Figure 1) an architecture concept is constructed. This concept is defined by the physical-functional structure which can now be used for sizing.

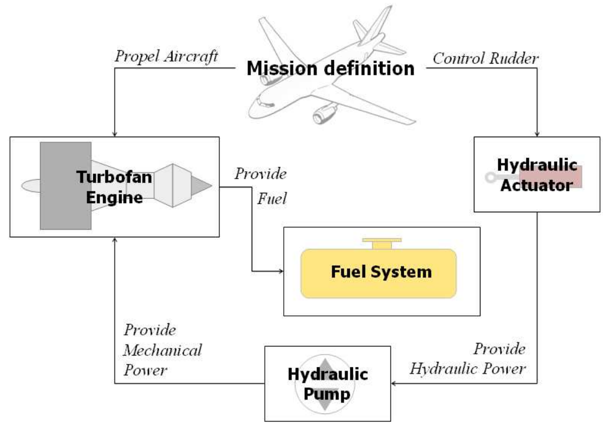

Figure 1: Example of a Functional/Physical Structure

If we now consider each function, the relationship it implies can be modeled by a flow of variables. The flow of variables corresponding to the functions in the example is listed in Table 1.

Table 1: Example of data flow characterizing functional relationships

Function Name	Function requesting side	Flow of variables (directionality)	Function provider side
Propel Aircraft		Thrust required [lbf] Flight conditions: OAT [°C] Speed [kn] Altitude [ft] →	
Provide fuel		Fuel mass required [lbm] →	
Actuate Rudder		Power required [W] Force required [Nm] Stroke length [m] →	
Provide Hydraulic Power		Flow required [gal/min] → Pressure provided [psi] ←	
Provide Mechanical Power		Power required [hp] → Speed provided [RPM] ←	

Using this flow of variables and applying it to the physical/functional structure defined previously, a model structure can be defined. Using sizing models for each system, this model

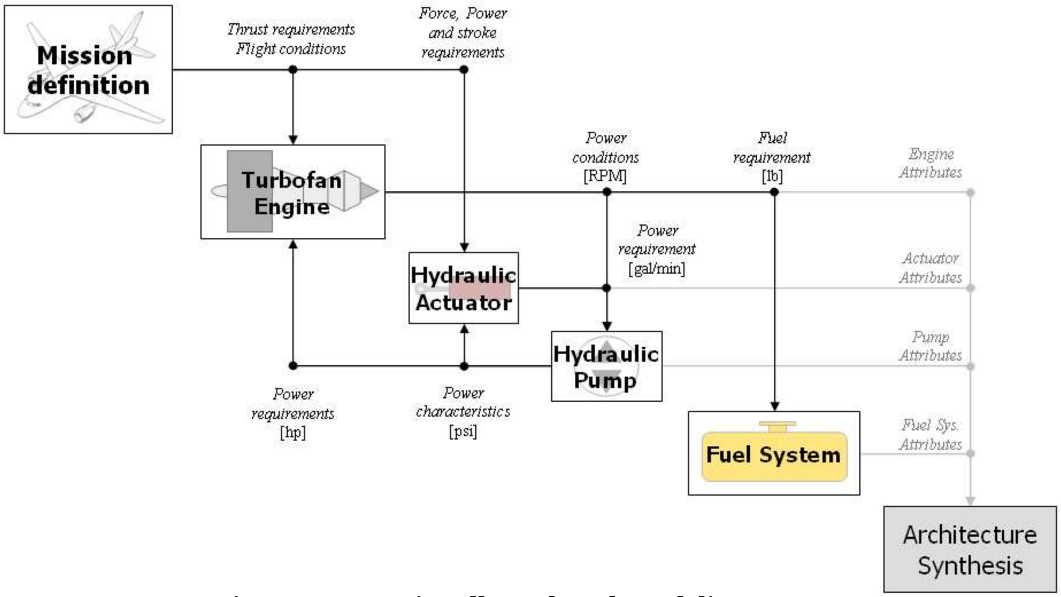

Figure 2: Functionally-Induced Modeling Structure

structure provides a means to estimate the size and performance of the architecture using a bottom-up approach. The model structure for the example is presented in Figure 2.

4. Operational space definition and exploration

The sizing of the systems must satisfy all potential requirements through-out the mission. In most cases, the functionalities of the systems composing the ASA change during the mission. In order to capture the variability in functionality, the mission envelope is broken down into scenarios. In the context of this paper, the mission envelope is referred to as the operational space. The scenarios are defined using a fragmentation of the operational space along three “dimensions”. The fragmentation of the design space is represented in Figure 3. The first dimension is the flight phase (taxi, take-off, climb, cruise, descent, landing etc…). The second dimension corresponds to flight conditions. It captures variability in weather (ΔISA, etc…) or flight environment (presence of icing conditions, dust, etc…). The third dimension captures system failures and will consider the failed architecture configurations. In the context of the example presented in the previous section, an alternate architecture is needed when the hydraulic pump is out of order

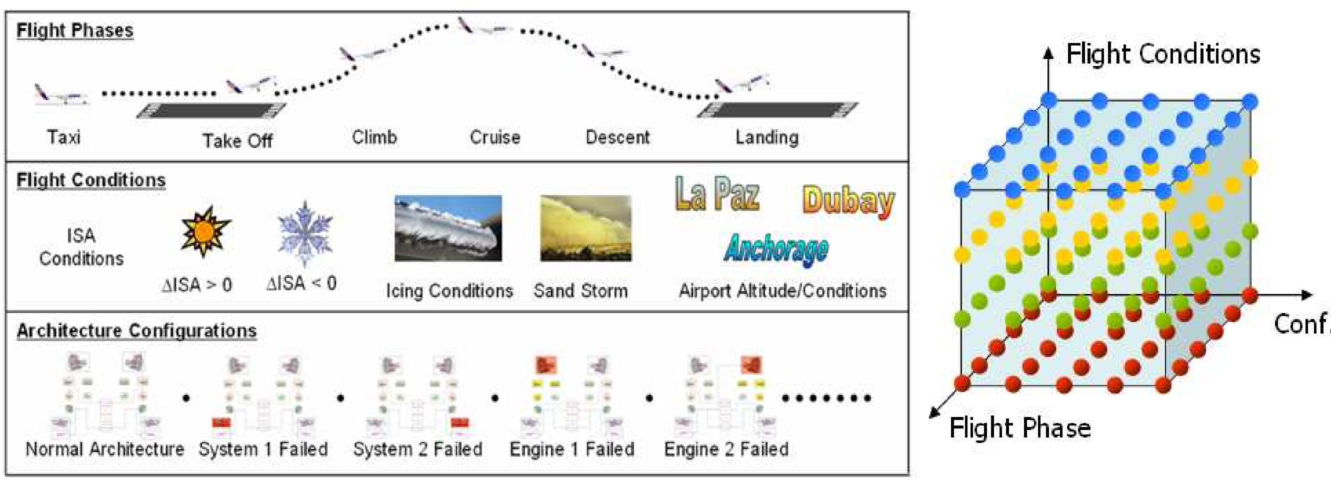

Figure 3: Parameterization/representation of the operational space

and a ram air turbine driven pump is deployed to provide hydraulic power.

Using this representation of the operational space at the architecture level helps identify the sizing scenario for each component in the architecture. The identification of the "sizing scenarios" is critical to the sizing process. This approach also allows for the fact that as the architecture changes, the sizing scenarios also change.

Even for experts, the identification of the sizing scenario for a given system within a new architecture can be difficult and may only be identified during commercial operation (after entry in service). As new architectures are defined, careful exploration of the operational space is necessary to ensure such a situation is avoided.

The importance of considering the entire operational space highlights the value of the functional induction techniques. One of the dimensions of the space (architecture configuration) requires different functional relationships. These changes in configuration impose very different information flows. The ability to adapt automatically the input/output structure by defining alternative functional flows corresponding to each configuration greatly facilitates the exploration of the operational space.

5. Formulation of a sizing process

In this paper, "system sizing" refers to the estimation of the physical attributes of the system. "Architecture sizing" will refer to the integrated attributes of the systems constituting the architecture. Given this definition of architecture sizing, a bottom-up approach (from system to architecture) is chosen.

Often in conceptual design, sizing processes at the system-level are based on the assumption that the system can be sized using parametric models based on its functionality. For example, in conceptual design, an electric generator with a capacity of X kVA (functionality) will have an estimated weight of Y kg (estimated "size"), with a mathematical expression linking the two values in the background. This approach which evaluates the physical attributes of a system based on its functionality will be referred to as functional sizing.

The advantage of functional is its simplicity from a computational point of view. In most cases, however, system experts do not think that way, and will refuse to commit to models of this type. Going back to the example of an electric generator, when an expert considers the design of a generator, he will consider a set of design variables (diameter of the rotor, number of windings, etc…) or, even more simply, will consider a set of Off-The-Shelf (OTS) generators (if he/she does not have the option to redesign it). From the resulting design space or available OTS set, the system expert will be selecting the feasible alternatives. In other words, the expert will select the designs which are able to fulfill the functional requirements imposed on the system. From these feasible alternatives, the expert will then select the "most appropriate" design and will propose it as the estimated sized system.

Using this design approach at the system-level is significantly more complex than using a functional sizing as it requires a decision-making process. But since the final design of a system is not only driven by its functionality, using purely functional-sizing models ignores system-level trades which may influence the physical attributes of the system.

Unlike architecture-level studies, system-level designs are often equipped in terms of numerical analysis. These tools are traditionally design oriented. In a case of an electric generator, given a rotor radius/geometry, and a winding setup, the power capacity, characteristics and generator attributes can be evaluated by the design model. In order to capitalize on this existing body of modeling capability, a methodology is proposed to transform those design-oriented models into sizing models driven from the architecture level.

The basis of the method is constructed on the formulation of a typical optimization process. A general optimization problem statement is presented in Equation 1 (from reference [4]).

$$
\begin{aligned}
&\underset{\overline{X}}{\text{Minimize}}: F\left(\overline{X}\right) \\
&\text{Subject to}: \; g_j\left(\overline{X}\right) \leq 0 \qquad j = 1, m \\
&\qquad\qquad h_k\left(\overline{X}\right) = 0 \qquad k = 1, l \\
&\qquad\qquad X_i^l \leq X_i \leq X_i^u \qquad i = 1, n \\
&\text{where} \quad \overline{X} = \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_n \end{Bmatrix}
\end{aligned} \tag{1}
$$

$\overline{X}$: Design variables

$F\left(\overline{X}\right)$: Objective - function

$g_j\left(\overline{X}\right)$: Inequality constraints

$h_k\left(\overline{X}\right)$: Equality constraints

$X_i^l \leq X_i \leq X_i^u$: Side constraints

Let us consider a system design model. This design model defines the attributes of a system based on design and operating variables. An illustration of such a model for a compressor design problem is provided in Figure 4. Functionally, the compressor is there to “provide compressed air” and in turn induces the function “provide mechanical power”. One of the attributes defined by the model is the quantified capability of the compressor to perform its function. In Figure 5, this capability is visually represented by the space delimited by the red line in the compressor map. The red line represents the maximum pressure ratio, mass flow and speed at which the compressor can operate. This capability space is scaled by the design variables (roughly speaking the throat area scales the map horizontally and the number of blades scales it vertically).

The previous section introduced the fact that a functional requirement for a system should be based on an ensemble of scenarios constituting the operational space. A hypothetical operational space is represented in Figure 5. Note that each scenario at the aircraft-level (represented by colored dot in Figure 3) will generate a different operation point on the compressor operation space in Figure 5.

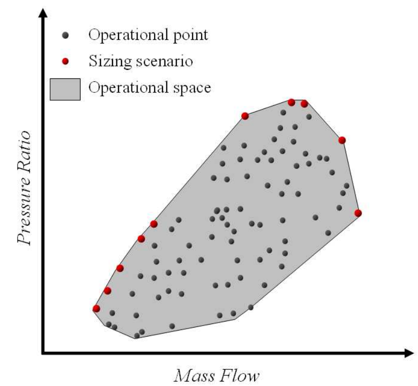

Figure 5: Operational space example for a compressor

If we follow the typical expert system decision making process the first step in sizing is to identify the design alternatives which are able to fulfill the functional requirements. From the visual representation of the capability and operational spaces, the operational space must be enclosed in the capability space to meet the requirement (see Figure 6). If we translate this

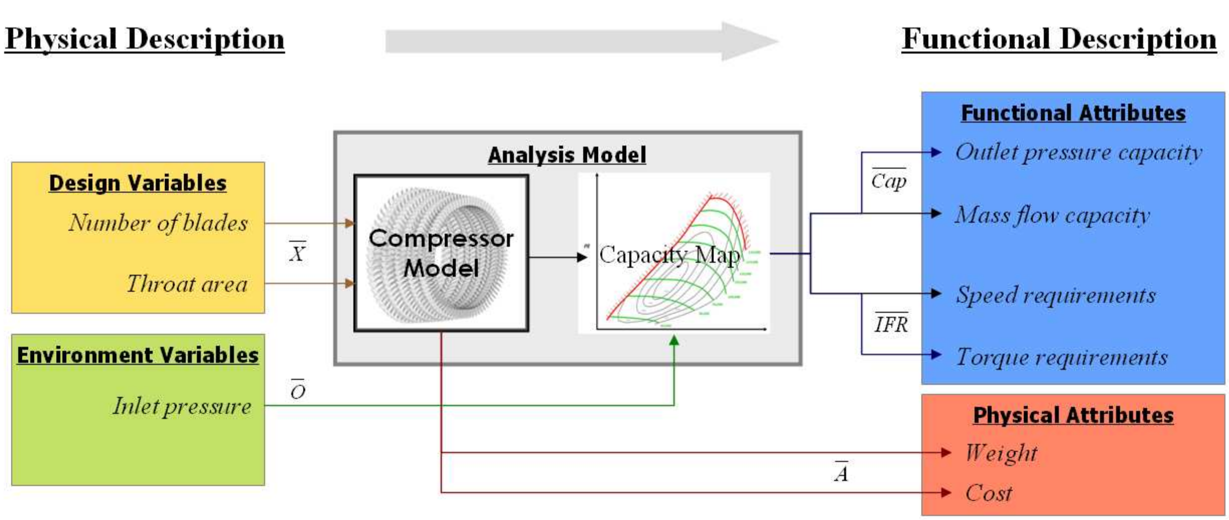

Figure 4: Example of an analysis model for a compressor

requirement into optimization settings we can say that functional compliance corresponds to the fulfillment of an inequality constraint.

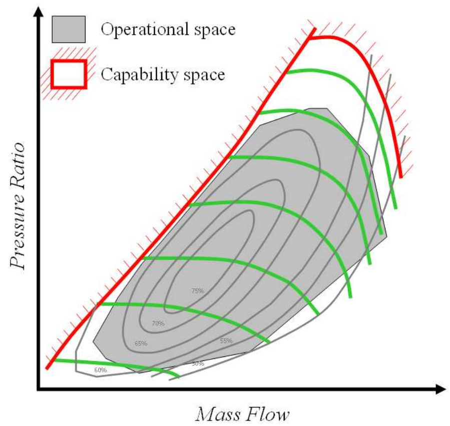

Figure 6: Enclosure of operational space within the capability space

Since we are trying to estimate the size of the system (compressor in the example), using only constraints is not sufficient as it allows cutting out infeasible designs, but does not necessarily lead the selection of one single design point. Over-performing or over-sized alternatives can not be eliminated, for example.

The next step is therefore to select what is the "most appropriate" system for the job. The identification of "most appropriate" indicates the use of some objective-function which will enable the ranking of feasible alternatives and selection of the "best" design given some objective-function (lowest weight, energy demand, cost, etc…). Therefore, the sizing processes can be formulated under the following optimization setup:

$$\underset{\overline{X}_j}{\mathrm{Min}} : F_j\left(\overline{\Gamma_{\cdot,j}}, {}^{j}\overline{A}\left(\overline{X}_j,\overline{O}\right), {}^{j}_{1}FR\left(\overline{X}_j,\overline{O}\right), \ldots, {}^{j}_{1}FR\left(\overline{X}_j,\overline{O}\right)\right)$$

$$\text{S.t.:} \quad {}^{m}_{j}FR_k - Cap\left(\overline{X}_j, \overline{O_{k,\cdot}}\right) \leq 0 \quad \text{for } k = 1,K \text{ and } m = 1,M$$

$$X^{l}_{d,j} \leq X_{d,j} \leq X^{u}_{d,j} \qquad d = 1,D \qquad (2)$$

$$\text{where} \quad \overline{X}_j = \begin{Bmatrix} X_{1,j} \\ X_{2,j} \\ X_{3,j} \\ \vdots \\ X_{D,j} \end{Bmatrix} \text{ and } \overline{O} = \begin{Bmatrix} O_{1,1} & O_{1,2} & & O_{1,J} \\ O_{2,1} & & & \vdots \\ & & & \vdots \\ O_{K,1} & & & O_{K,J} \end{Bmatrix}$$

<u>Note:</u> Variables for all equations are described at the end of the paper.

This optimizer based structure for design oriented sizing is very powerful in its applications. First, it provides an alternative functional sizing methods often based on regression. Regressions are by definition empirical and not accurate when it comes to extrapolating beyond previously explored solutions. The optimizer based sizing facilitates the application of design models, which are validated by experts and more transparent in their assumptions. Also design models are more-likely to be physics-based models which are typically more accurate in their predictions for new designs. Secondly, using an optimization setup forces the formulation of an objective-function. This formulation improves the tracking of assumptions used in the sizing process by highlighting how the chosen design was selected. The third important advantage of this formulation is the automation of the process. Once properly setup, optimization processes can be automated. This automation provides a means for the system expert to concentrate on the monitoring of assumptions used by the models rather than performing basic design tasks. Improving the automation of the process while protecting the sizing procedure from ill based assumptions, is an important step toward automated architecture sizing. Allied with the functional induction setup, presented in the previous section, these methods provide great improvements toward the automated analysis of ASA.

6. Multi-level optimization of sizing processes

As previously mentioned, one of the greatest difficulties faced by the industry is how to relate architecture level objectives to system level objectives. This challenge was expressed by the formation of diverse technical committees and research efforts. Two of those are the Power Optimized Aircraft (a European project), and the Energy Optimized Aircraft Systems (Program committee organized by AIAA). It should be noted that both titles have the word "optimized" in their name and both refer to optimization based on energy criteria. Now that those objectives are defined, the question of how to actually optimize remains. Which metrics should be optimized at the system level?

Should it be mass, energy consumption, cost, drag? Obviously the answer is: It depends. That is precisely why we have architects. Theoretically, one can consider that the architect is the one directing the system development by formulating the functional specifications of the system (system-level constraints) and by indicating which attributes of the system should be optimized (system-level objective-function). Let us now consider the challenge in light of the solutions proposed previously. On one side we have the question: Which metrics should be optimized at the system-level? On the other side we have an automated sizing model based on system-level objective functions. In order to address the question, we propose to base the sizing process of the architecture on a two level optimization process.

Let us assume, for now, that objective-functions at the system-level are structured in an OEC (Overall Evaluation Criterion) described in reference [5]. We consider an architecture composed of power consuming systems and a power providing system. For the sake of the example, the architecture should be optimized on overall weight and the power providing system is simply scaled on power consumption (i.e. its weight is proportional to power delivered). We also assume that the power consuming systems are scaled using the optimizer-based sizing method presented in the previous section. The sizing optimization processes for each system will be based on an objective-function of the form:

$$F_m\left(\overline{X}\right) = \gamma_{1,m} \times \frac{{}^m r_1\left(\overline{X}\right)}{{}^m r_{1-ref}} + \gamma_{2,m} \times \frac{{}^m r_2\left(\overline{X}\right)}{{}^m r_{2-ref}} \qquad (3)$$

where $\gamma_{1,m} + \gamma_{2,m} = 1$

${}^m r_1$: Weight of system "m"

${}^m r_2$: Energy consumed by system "m"

The OEC is a weighted sum of attributes. The weightings $\gamma_{1,m}$ and $\gamma_{2,m}$ give a relative importance to the attributes constituting the objective function. These weightings indicate if the design of the motor should optimize weight or power consumption.

The problem can now be formulated as: "What is the system-level objective-function which will optimize the architecture-level objective-function?" This formulation can be translated into a new layer of optimization at the architecture level:

$$\begin{aligned}
&\underset{\Gamma}{\text{Minimize}} : Weight_{Architecture}\left(\overline{\Gamma}\right) && (4)\\
&\text{Subject to} : 1 - \gamma_{1,m} - \gamma_{2,m} = 0 && m = 1, M\\
&\qquad 0 \le \gamma_{i,m} \le 1 && \text{and } i = 1,2\\
&\text{where} \quad \overline{\Gamma} = \begin{Bmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,M} \\ \gamma_{2,1} & \gamma_{2,2} & \cdots & \gamma_{2,M} \end{Bmatrix}
\end{aligned}$$

Using this architecture-level optimization process will define the most appropriate weighting for the system-level objective-functions. In the example, this will identify the most appropriate objective function for the energy loading system in order to optimize the architecture overall weight. Based on our previous assumption, since the power providing system weight is proportional to power demand, it should be expected that attention should be dedicated to the power consumption of the power loading systems.

If we consider that the feasible alternatives for the power consuming systems lay on a Pareto front for weight and power requirements, a notional representation of the design space topology is provided in Figure 7.

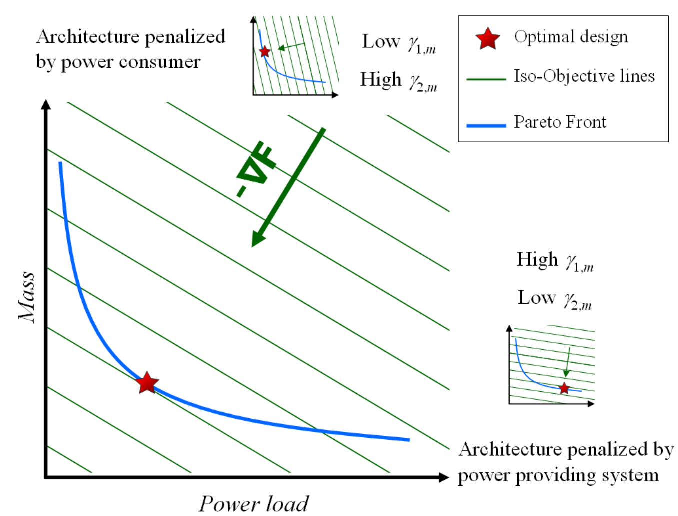

Figure 7: Topology of design space at system-level

In Figure 7, we can see how each level of the optimization process contributes to the sizing process. The architecture-level orients the topology of the space, while the system-level identifies the best design given the topology. As the architecture-level optimization increases the weighting on mass ($\gamma_{1,n=m}$), the slope of the green lines will decrease which drives the system-level optimizer toward lighter solutions (but to the detriment of power). The role of the architecture-level optimizer is to find the proper balance in the system-level objective-functions and drive the system-level optimization toward an optimum beneficial to architecture level objectives.

7. Comments on the overall process and future works

The methodology introduced in this paper proposes to perform a bottom-up approach to architecture sizing. The format in which the system-level analysis models are integrated is represented in Figure 8.

The models constituting the system-level sizing are based on an optimization process where functional requirements provide the constraints to the sizing process which selects the best solution meeting those requirements. The inputs/outputs between the sizing processes are defined using a functional induction approach which translates an architectural concept into structured flows of variables. This structure allows embedding the system-level sizing processes into an architecture-level optimization process. Based on an objective at the architecture level, the multi-level optimizer defines the system-level optimization objective-functions which, in turn, define the best system alternative.

The purpose of this process goes beyond the estimation of the sizing of the architecture. The

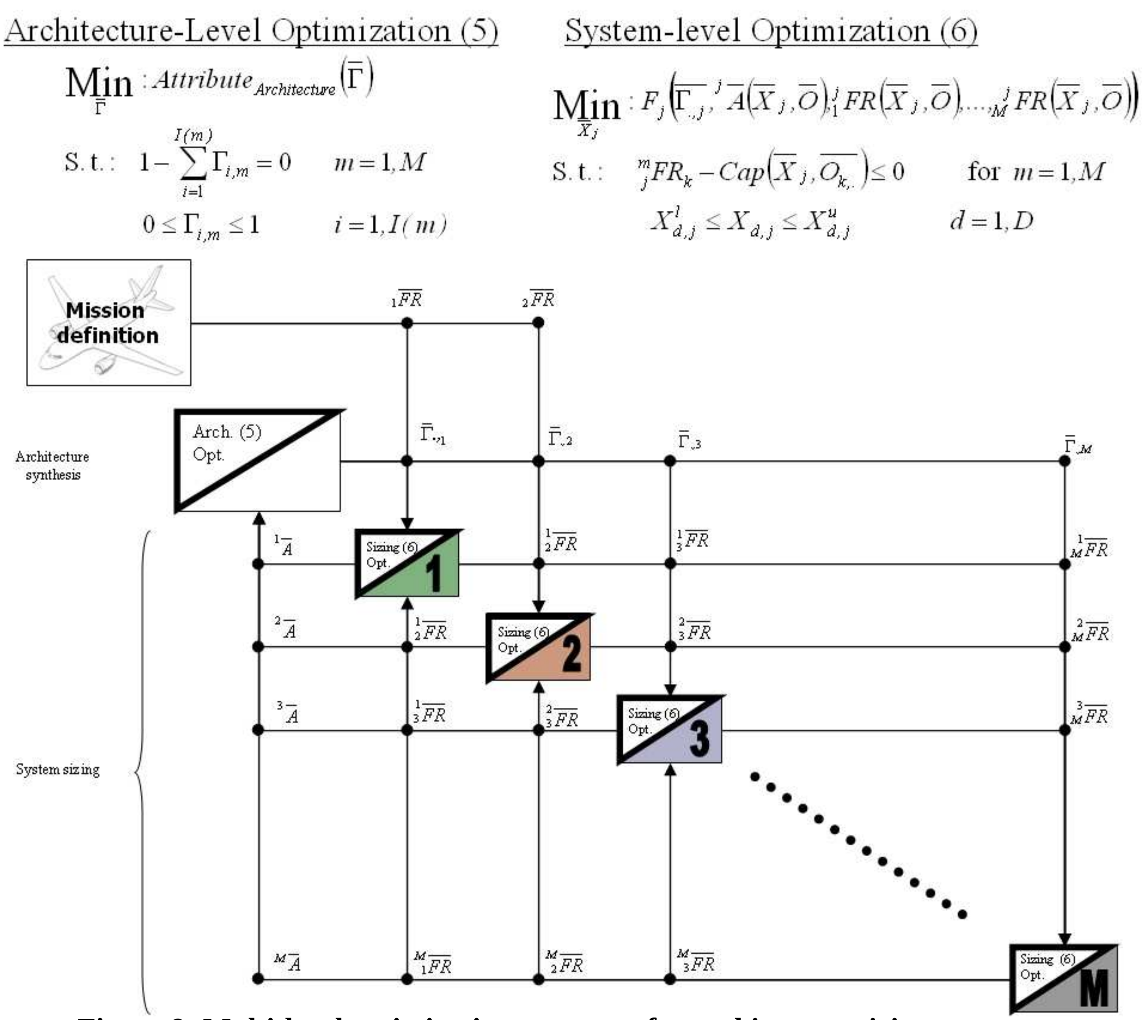

Figure 8: Multi-level optimization structure for architecture sizing

fundamental advantage of this method is its transparency from a system point of view. The bottom-up approach takes into consideration system-level trades while clarifying which assumptions were used to estimate the attributes of the systems.

The fact that this methodology provides a means to generate automated analysis opens the field for the application of advanced design techniques. Several trades have particular importance from the authors' perspective. The first is the comparison of different architectures in their ability to meet aircraft level-objectives. That is, given a technology portfolio, identify which architecture yields the best aircraft level performance.

The second is the optimization of system-level technological strategies in a direction which supports aircraft-level objectives. By defining system-level objective-functions, the synergistic optimization approach provides guidelines on technology development. The effects of technologies are easily traceable to the attributes of the system which host them. The optimization of the objective-functions at the system-level provides the expert with a clear and unambiguous formulation of objectives within their perimeter of action.

The foundation of the work proposed in this paper is the field of multi-level optimization. Multi-level optimization distributes the computational load of optimizing a large/complex analysis on multiple and comparatively simpler sub-analyses. The techniques proposed in this field offer ways to adapt a centralized optimization problem. This is done by considering coupling variables from contributing analyses as design variables at the top-level optimization process. Those pseudo independent variables are subjected to compatibility constraints with respect to the value produced by contributing analyses. At the level of contributing analyses, local design variables are optimized to minimize expressions corresponding to the compatibility constraints at the top-level. Following this approach, multiple techniques were proposed. Reference [6] presents a good review of those various techniques. Most techniques differ from the one presented in this paper by their focus, which influences the way levels collaborate. Rather than being centered on mathematical aspects, the focus of this study is on the integration of design knowledge and objectives between architecture and system-levels. As a result, rather that setting targets on exchanged information between analyses, the top-level optimizer steers the bottom-level optimizer processes by modifying their local objective-functions.

Multi-level optimization techniques can be complex to put in place. The convergence of collaborating analyses is sometimes difficult to achieve (see reference [6]). However, the multi-level optimization process is motivated by setting up design problems at the system-level through the specification of system functional specification and formulation of an optimal objective function. Thus, the advantages of fragmenting the design optimization process must be traded off with potential loss in the optimizer efficiency due to the presence of multiple layers of processes. The computational stability and efficiency of the overall process must, therefore, be explored by the authors on a prototype problem.

8. Conclusions

This paper presented a methodology which both facilitates the sizing process of an ASA and opens the field for new trade studies. The resulting automated process for the integration of numerical analyses at the system-level facilitates the creation of numerical models for architecture sizing. These models address the need for an automated computational capability necessary to the application of the state of the art methods in design.

Beyond the sizing of systems composing the architecture, the multi-level optimization approach proposed in this paper facilitates the translation of aircraft-level objectives into more focused objective-functions at the system-level. These focused objective-functions provide clearer objectives for technology developments. The methods presented in this paper will be illustrated by a proof of concept in future publications by the authors.

Acknowledgment

The authors would like to thank the European Aerospace Defense and Space Company for its support and leadership in motivating this paper. The authors also would like to thank the members of the Energy Optimized Aircraft Systems program committee (AIAA) for their initial contribution in defining the problem.

References

[1] Biltgen P, Ender T. and Mavris, D. **“**Development of a Collaborative Capability-Based Tradeoff Environment for Complex System Architectures” *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno NV, January 2006.

[2] Kirby, M.R. “A Methodology for Technology Identification, Evaluation and Selection in Conceptual and Preliminary Aircraft Design”, Ph.D. Dissertation, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA. Mar.2001.

[3] Mavris D., de Tenorio C., Armstrong M., “Methodology for Aircraft System Architecture Definition”, *46th AIAA Aerospace Sciences Meeting and Exhibit,* Reno NV, January 2008.

[4] Vanderplaats, D., “Numerical Optmization Techniques for Engineering Design”, 4th Edition Vanderplaats R&D, Inc. Colorado Springs, CO, 2005

[5] Hwang, C.-L., Yoon, K., “Multiple Attribute Decision Making: Methods and Applications: A State-of-the-Art Survey”, Springer-Verlag, Berlin, 1981.

[6] de Wit A.J., van Keulen F., “Numerical Comparison of Multi-Level Optimization Techniques”, 46th AIAA Structures, Structural Dynamics and Material Conference, Honolulu, Hawaii, April 2007.

Symbols

$\overline{X}_j$	Design variables for system *j*
$\overline{O}$	Variables defining the operational environment of the system
${}^{j}\overline{A}\left(\overline{X}_j,\overline{O}\right)$	Attributes of system *j*
${}^{j}_{m}\overline{FR}\left(\overline{X}_j,\overline{O}\right)$	Functional requirements induced by system *m* on system *j*
${}^{m}_{j}\overline{FR}_k$	Functional requirement imposed on system *m* by system *j* in scenario *k*
$\overline{Cap}_k\left(\overline{X}_j,\overline{O}_{k}\right)$	Capacity (i.e. maximum functional requirement which can be performed by system) under operational scenario *k*
${}^{m}r_i\left(\overline{X}_j,\overline{O}_{k}\right)$	Response *i* (corresponds to an attribute or functional requirement) of system *m*
${}^{m}r_{i-ref}$	Reference value for response *i*
$\overline{\Gamma}$	Matrix containing weighting factors of objective-functions at system-level
$\gamma_{i,m}$	Weighting factor on response *i* of system *m*
M	Number of systems composing the architecture
$I(m)$	Number of responses for system *m*
K	Number of scenarios used to explore the operational space