JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 30, No. 4, July–August 2007

# Flight Control System Architecture Optimization for Fly-By-Wire Airliners

Christophe Bauer* and Kristen Lagadec†
*Airbus France, 31300 Toulouse Cedex 3, France*
and
Christian Bès‡ and Marcel Mongeau§
*Université de Toulouse, 31062 Toulouse Cedex 9, France*

DOI: 10.2514/1.26311

**The design problem of a flight control system on a large fly-by-wire airliner is to find combinations of actuator(s), power circuit(s), and computer(s) for each control surface, to fulfill the constraints imposed by the safety regulations, while keeping the resulting system weight as low as possible. The trend toward more electrical aircraft makes it harder and harder to determine, in a reasonable computer time, optimal architectures solely by traditional trial-and-error methods. This paper introduces a flight control architecture optimization process, intended as a decision aid for system engineers at early stages of the flight control architecture definition. We present an optimization model for the design process, based on a safety constraint and a weight criterion, that allows the exploitation of traditional design rules in a systematic manner. We start by reducing the initial search domain through introducing the notion of *surface possible architecture*, which takes into account technological constraints and empirical practices. Then, we use an adaptation of branch-and-bound methods to solve the remaining discrete optimization problem. Finally, an application to the Airbus A340 roll control system is addressed. An exact optimum is found among $10^{14}$ possible architectures in less than 25 min on a standard desktop computer. Our methodology is currently under the process of industrial implementation at Airbus, where it will be used in the early design stage as a decision-analysis tool.**

## I. Introduction

THE trend toward more electrical aircraft is gradually being implemented to back up or replace with electrical power the hydraulic circuits that power the flight control actuators. This phenomenon increases the number of available choices for the architecture of a control surface; for each flight control actuator, we can now install a classical servo-control (S/C) connected to a hydraulic circuit, an electrohydrostatic actuator (EHA) connected to an electrical circuit, or even an electrical-backup hydraulic actuator (EBHA), connected to both types of power sources. In addition, all fly-by-wire architectures have to define, for each actuator, the associated flight control computer(s) to provide the control signals.

The optimal architecture is the one that is sufficiently robust to failures so as to ensure flight safety, while minimizing the weight of the control system. Current approaches for solving this design problem are based on expertise, trial and error, and iterations between various disciplines (aerodynamics, functional hazard assessment, handling qualities, system architecture, etc.). However, the possible number of candidate architectures can now be extremely high, especially for aircraft with many control surfaces such as the Airbus A380. The design process involves so many alternatives and has to consider so many failure cases that manual optimization is impractical. This is especially true at early design stages when frequent changes require complete new iterations. Some level of automation is therefore necessary to assist the flight control system designer in this task. The purpose of this paper is to build a methodology that can aid the designer faced with this hard combinatorial problem.

The roll [1] architecture has been chosen as an example for the purpose of this study. We concentrate on the roll axis as it is the most complex in terms of number of architecture possibilities. For instance, on the A380 there are 18 roll control surfaces (6 ailerons and 12 spoilers), compared with 5 for pitch, and 3 for yaw. The design problem that we consider here is to determine an architecture for roll control, i.e., a series of technologically-feasible combinations of actuator(s), power circuit(s), and computer(s) for each roll control surface, that guarantees flight safety while minimizing the weight. Note that the method we propose for roll can be extended straightforwardly to the whole flight control system (roll, pitch, and yaw).

The paper is organized as follows: in Sec. II, we present the general problem and propose a preliminary model. We introduce the safety constraints and the weight model. We then show that enumerating all possible architectures is far beyond the reach of any computer, and that additional knowledge has to be somehow introduced to reduce the algorithmic complexity. In Sec. III, we first reduce the size of the search domain by exploiting symmetry. Then, we integrate technological constraints directly in the architecture construction process, through the notion of *aileron-* and *spoiler-possible architectures*. Finally, we introduce a resolution methodology relying on a branch and bound tree-search method. In Sec. IV, we report the results obtained by running the proposed methodology on the A340 architecture. The method succeeds in finding an exact optimal solution among $10^{14}$ possibilities, within 25 min on a standard desktop computer. In Sec. V, we conclude and open further perspectives.

## II. Roll Architecture Design

Currently on large airliners, roll motion is controlled by ailerons and spoilers, although other types of control surfaces may be considered in the future. Each of these surfaces can be deflected by one or more actuators, which can withstand large aerodynamic efforts. Actuators require at least one power source (hydraulic or

Received 10 July 2006; revision received 8 December 2006; accepted for publication 8 December 2006. 

*Ph.D. Student, Flight Control Systems Department.

†Engineer, Flight Control Systems Department; currently Astrium European Aeronautic Defense and Space Company, Toulouse.

‡Professor, Laboratoire de Génie Mécanique de Toulouse, Université Paul Sabatier.

§Maître de Conférences, Institut de Mathématiques de Toulouse, Université Paul Sabatier; on leave at Laboratoire d'Analyse et d'Architecture des Systèmes–Centre National de la Recherche Scientifique.

electrical) and at least one control signal from one of the flight control computers. The design problem of a flight control system architecture is to find combinations [actuator(s), power circuit(s), computer(s)] for each aileron and spoiler, so as to meet the roll-performance constraints imposed by safety regulations (Federal Aviation Regulations/Joint Aviation Requirements), while keeping system weight as low as possible. Flight safety requirements are defined by air-worthiness regulations and are driven by considerations of global robustness of the roll control function against the probability of various system failures.

In this section, we first detail the elements of the roll architecture on Airbus aircraft, and then we formalize the safety constraints. Then, we define an approximate model for the estimation of the weight criterion. Finally, we assess the combinatorial complexity of the discrete optimization problem.

### A. Elements of the Roll Architecture

#### 1. Wing Control Surfaces

Wing control surfaces can control the roll motion of the aircraft by creating differential lift across the wings (see Fig. 1). There are typically two types of roll control surfaces (see Fig. 2): ailerons are hinged portions of the trailing edge that can be deflected downward (or upward) to create local uplift (or downlift, respectively) on the wing; spoilers are hinged over-wing panels that can be deflected upward to upset the airflow and efficiently destroy local lift.

#### 2. Actuators

To deflect these control surfaces, actuators are necessary. For example, three actuator technologies have been retained for the A380 flight control system: 1) conventional hydraulic servo-controls, powered by one hydraulic circuit; 2) electrohydrostatic actuators, powered by one electrical circuit, and 3) electrical-backup hydraulic actuators, powered by one hydraulic circuit and one electrical circuit for backup.

For each actuator, the choice of technology influences its individual failure rate and weight contribution, allowing to improve either system robustness or weight. Weights and failure rates depend on the components' manufacturers. To give an order of magnitude, there can be as much as 17 kg of weight difference between the lightest and the heaviest actuators, and failure rates range from $0.17 \times 10^{-5}$ to $3 \times 10^{-5}$ per flight hour.

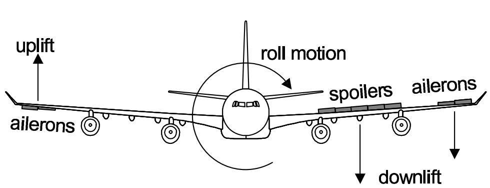


**Fig. 1 Differential lift across the wings induces a rolling motion.**

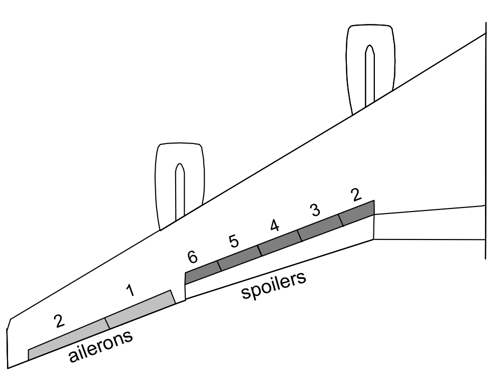


**Fig. 2 Ailerons and spoilers on the Airbus A340 wing.**

#### 3. Flight Control Computers

Flight control computers (FCCs) translate pilot or autopilot orders into aileron or spoiler deflections, and then control the stroke of each actuator to achieve the correct motion. One computer can control several actuators, and each actuator can be connected to more than one computer to mitigate the consequences of individual computer failures. There are usually five to six flight control computers on current aircraft.

#### 4. Power Circuits

Power circuits distribute the energy produced by the engines to the flight control actuators. These circuits can be hydraulic and/or electrical. Before the A380, Airbus aircraft relied on three hydraulic circuits, but on the A380 one of these hydraulic sources was replaced by two electrical circuits, resulting in a so-called 2H-2E architecture.

### B. Safety Constraints

#### 1. Roll Performance

For a failure case affecting one or more roll control surfaces, the roll-performance degradation is assessed through the notion of residual roll rate, which is the steady-state rate of rotation around the roll axis that can be achieved with the remaining control surfaces. It is approximated by the following formula:

$$p_\infty = \min_{\text{right/left}} \left( \frac{V}{Cl_p \cdot L} \cdot \sum_i Cl_{\delta l_i} \cdot \delta l_i^{\max} \right)$$

where $p_\infty$ is the residual roll rate, $V$ is the (true) airspeed, $Cl_p$ is the wing roll damping coefficient opposing the roll motion, $L$ is a reference length (wing aerodynamic mean chord), $Cl_{\delta l_i}$ is the roll efficiency coefficient for control surface $i$, and $\delta l_i^{\max}$ is the maximum deflection for control surface $i$. As efficiency coefficients $Cl_{\delta l_i}$ depend on the flight conditions (Mach number, dynamic pressure, high-lifted configuration), the consequences of each failure case have to be evaluated for several flight points. We consider only failures in which the control surfaces return to their zero-deflection position. Past experience has shown that taking into account more complex failure scenarios (runaway, jam, free-floating, aerodynamic, or fuel imbalance) does not increase the accuracy of the prediction at this early design stage.

#### 2. Safety Requirements

Flight safety requires that each failure affecting roll control has a consequence in relation to its probability of occurrence: the higher the failure rate, the lower the acceptable degradation of roll performance. Airbus effectively uses internally a *required roll performance $p_r$ vs failure rate* $\lambda$ template (see Fig. 3), which encompasses all regulatory requirements while conveying pilot judgment of acceptable roll-performance degradations. During the flight control system design process, when one failure case among all

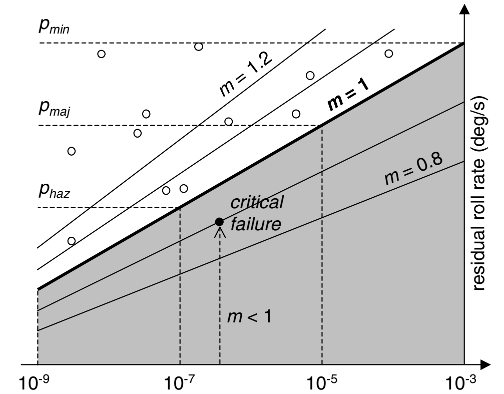


**Fig. 3 Safety constraints (template of roll performance with respect to failure rate).**

possible combinations of failures (actuators/circuits/computers) achieves an insufficient roll performance with respect to its probability of occurrence, the corresponding roll architecture must be rejected (see Fig. 3). We define a normalized safety indicator $m$, which is below (or above) one when the roll performance for the critical failure case is below (or, respectively, above) the template :

$$m := \min_f \frac{p_\infty(f)}{p_r[\lambda(f)]} \quad \text{for every failure case } f$$

### 3. *Safety Constraint Evaluation*

Evaluating the safety constraint requires the generation of each possible failure case affecting the roll function, then assessing its residual roll rate for every relevant flight condition and comparing it to the template. This CPU-intensive evaluation process will be considered here as a given black-box function.

## C. Technological Constraints

We say that an architecture is *technologically acceptable* if it respects given rules related to technological choices or in-house design practices (generally dependent on the specific aircraft program choices). For example

1) Each actuator should be connected to the appropriate power source type (e.g., an S/C to a hydraulic circuit, an EHA to an electrical circuit).

2) Each actuator should be connected to at least one computer (single computer) and at most two (dual computer).

3) Some routing rules should be respected: for a given actuator, electrical power and computer signals should come via the same route (wiring is routed via a discrete set of routes in the wing).

4) A spoiler actuator should rely on a single computer.

5) Each aileron should be moved by (at least) two actuators to avoid flutter in case of single actuator failure.

6) There should be at most one EHA on each aileron, and no EBHAs.

7) Actuators on a same aileron should have different architectures (i.e., different power source and/or computers).

8) Actuators on a same aileron should be connected to the same number of computers.

For a chosen architecture $A$, these constraints are summarized by a function $t$:

$$t(A) = \begin{cases} 1 & \text{if } A \text{ complies to the rules} \\ 0 & \text{otherwise} \end{cases}$$

## D. Model Formulation for the Weight Criterion

### 1. *Global Weight Model*

The objective of the flight control system optimization problem is to minimize the weight impact of the retained flight control architecture. Airbus has weight models based on statistical data on existing aircraft, which can reasonably assess the weight impact of a given architecture. Each choice of actuator technology, and connections to power sources and computers, has a consequence on the global system weight. For each architecture $A$, the system weight $w(A)$ is influenced by the following factors: 1) heavier/lighter actuator; 2) longer/shorter piping or wiring to convey the chosen power source to the actuator; and 3) marginal increase/reduction of the power generation and distribution equipment, resulting from the marginal consumption of the actuator on the chosen power circuit.

### 2. *Linear Weight Model*

In early design phases, an accurate weight evaluation is generally not available. Statistical data based on previous aircraft programs can provide a regression-based weight model. For a new aircraft program, the weight of a reference architecture is assessed with the weight model, and the variations from this reference are expressed as linear combinations of the design variables. This linear formulation was proved sufficiently accurate for the early design stages, which is detailed next.

Let the algebraic weight difference $\delta w$ of architecture $A$, with respect to (wrt) the reference architecture $R$, be expressed via the weight model $w$ through

$$\delta w(A) = w(A) - w(R) = w(a_1, \ldots, a_n) - w(r_1, \ldots, r_n)$$

where architecture $A$ is the list of all individual architecture choices $a_i$ for each control surface $i$, $i = 1, \ldots, n$, and $R$ is the global definition of the reference architecture determining the reference weight, with individual architecture choices $r_i$, $i = 1, \ldots, n$. The first-order weight impact $\delta_i w$ of an individual architecture choice $a_i$ for control surface $i$ is then obtained through

$$\delta_i w(a_i) = w(r_1, \ldots, r_{i-1}, a_i, r_{i+1}, \ldots, r_n) - w(R)$$

Then, the first-order global weight impact of architecture $A$ is the sum of the individual contributions of its components $a_i$:

$$\delta w(A) = \sum_{i=1}^{n} \delta_i w(a_i)$$

An architecture choice $a_i$ for control surface $i$ is an index into the vector of the $N_i$ possible architecture choices for control surface $i$ ($a_i = 1, \ldots, N_i$). Then, we can define $\delta W$ as the matrix of weight impacts for all possible architecture choices for all control surfaces. Component $(i, j)$ of matrix $\delta W$ is the individual weight contribution of architecture choice $j$ for control surface $i$:

$$\delta W(i, j) = \delta_i w(j), \qquad (i = 1, \ldots, n, \quad j = 1, \ldots, N_i)$$

Therefore, the weight impact of architecture $A$ is expressed through

$$\delta w(A) = \sum_{i=1}^{n} \delta W(i, a_i) \tag{1}$$

Matrix $\delta W$ is built offline from the aforementioned statistical data. The optimization process uses Eq. (1) for very fast weight estimation.

## E. Combinatorial Complexity of the Problem

### 1. *Optimization Problem*

We can summarize the architecture optimization problem as follows: find a combination $A$ of individual architecture choices $a_i$ for each control surface $i = 1, \ldots, n$ that minimizes

$$\delta w(A)$$

subject to safety constraints

$$m(A) \geq 1$$

and technological constraints

$$t(A) = 1 \tag{2}$$

where the weight model $\delta w$ is given through matrix $\delta W$, the safety criterion $m$ is given under the form of a black-box function, and the technological constraint $t$ is given as an explicit set of rules.

### 2. *Size of the Search Domain*

For each individual architecture choice $a_i$ for control surface $i$, there are many actuator configuration choices: 1) several actuator technologies, 2) several power sources, and 3) several flight control computers.

The number of actuator configurations $N_{\mathrm{act}}$ can be computed formally through

**Table 1 Number of architecture combinations on four example architectures**

| | | A320 | $A340_{3H}$ | $A340_{2H2E}$ | A380 |
|---|---|---|---|---|---|
| Number of hydraulic circuits | $n_h$ | 3 | 3 | 2 | 2 |
| Number of electrical circuits | $n_e$ | 0 | 0 | 2 | 2 |
| Number of computers | $n_c$ | 5 | 5 | 6 | 6 |
| Number of spoilers | $n_s$ | 8 | 10 | 10 | 12 |
| Number of ailerons | $n_a$ | 2 | 4 | 4 | 6 |
| Actuator combinations | $N_{\mathrm{act}}$ | 75 | 75 | 288 | 288 |
| Spoiler possibilities | $N_s$ | 75 | 75 | 288 | 288 |
| Aileron possibilities | $N_a$ | >5000 | >5000 | >80,000 | >80,000 |
| Candidate architectures | $N$ | $>10^{22}$ | $>10^{33}$ | $>10^{44}$ | $>10^{59}$ |

$$N_{\mathrm{act}} = \underbrace{(\underbrace{n_h}_{\text{S/C}} + \underbrace{n_e}_{\text{EHA}} + \underbrace{n_h \cdot n_e}_{\text{EBHA}})}_{\text{power sources}} \cdot \underbrace{[\underbrace{n_c}_{\text{if 1 FCC}} + \underbrace{n_c \cdot (n_c - 1)}_{\text{if 2 FCC}}]}_{\text{computers}}$$

where $n_e$, $n_h$, $n_c$ are defined in Table 1.

We can deduce the number of architecture combinations for the various types of control surfaces, depending on how many actuators each control surface has to rely on: for a spoiler $s$ (requiring one actuator), $N_s = N_{\mathrm{act}}$; for an aileron $a$ (requiring two actuators), $N_a \approx N_{\mathrm{act}}^2$; for a control surface $x$ (requiring $k$ actuators), $N_x \approx N_{\mathrm{act}}^k$.

For a flight control system $A$ featuring $n_a$ ailerons, $n_s$ spoilers (optionally $n_x$ control surfaces of another type), the total number $N$ of candidate architectures is given by

$$N = N_a^{n_a} \cdot N_s^{n_s} \cdot N_x^{n_x}$$

It is worth mentioning that a priori not all such candidate architectures enumerated in the preceding paragraph observe any of the technological constraints listed in Sec. II.C, i.e., this number is a theoretical maximum. Thus, it is in the specific case of large airliners that our methodology is more likely to prove useful, as will be discussed in the next paragraph.

*3. Illustrative Example*

For large airliners, the total number of candidate architectures involves a combinatorial complexity far beyond the reach of any computer, as shown in Table 1. As an example, let us consider the case of the A380 architecture, for which there are more than $10^{59}$ architecture combinations. Assuming that the criterion and constraints can be evaluated in only 1 ns, and that the optimization process only has to test one architecture in one billion, the required CPU time would still be over $10^{33}$ years.

Consequently, a practical optimization algorithm must drastically reduce the search domain. In the next section, we shall for this purpose

1) Consider specific symmetries of the roll function.

2) Build architectures that fulfill the technological constraints *by construction*.

3) Use prefilters to reject unsuitable architectures before running the evaluation tests.

## III. Model Formulation and Problem Resolution

### A. Reduction of the Search Domain

*1. Symmetries in the Roll Function*

As illustrated in Fig. 1, the spoilers on the left wing only contribute to left turns, and conversely those on the right wing only contribute to right turns. As the roll-performance requirement applies to both turn directions, the optimum architecture for spoilers has to be symmetrical. This allows us to reduce significantly the number of spoilers considered (6 instead of 12 on the A380 example). Note that the same does not apply to ailerons, as both left and right ailerons contribute to both left and right turns; an aileron failure can be mitigated by a nonfailed counterpart on the opposite wing, justifying nonsymmetrical aileron architectures.

*2. Conforming to Technological Constraints*

Instead of the classical approach of designing a flight control system (FCS), which is then certified by certain criteria, it is possible, using the constraints mentioned in Sec. II.C to construct FCS architectures from a requirements perspective. We introduce the notion of aileron-possible and spoiler-possible architectures (APA and SPA, respectively). They are subsets of the possible combinations mentioned earlier, restricted to the combinations that fulfill the technological constraints. To create these APA and SPA (and XPA for control surfaces of any other type), we first determine the compliant architectures for individual actuators. Then, for each control surface (relying on $k$ actuators), we combine the possible actuators while still satisfying the technological rules. For each type of control surface, this results in a short list of the only architecture choices that are technologically acceptable under the rules of Sec. II.C.

The construction of APA and SPA is illustrated on the simplified example of Fig. 4 (in which there are two hydraulic systems $G$ and $Y$, one electrical system $E$, and four computers $P1$, $S1$, $P2$, and $S2$). Let us first consider every power circuit (top left). Following the first rule of Sec. II.C, this leads to five power circuit choices: two possible S/C arrangements ($G$ or $Y$), two EBHA arrangements ($G + E$ or $Y + E$), and one EHA arrangement (only $E$). Second, we do the same (second rule) with every computer (top right), leading to four single-computer arrangements ($P1$, $S1$, $P2$, $S2$) and two dual-computer arrangements ($P1 + S1$ or $P2 + S2$); routing rules (third rule) for computer signals prevent $P1 + S2$ or $S1 + P2$ solutions. In a third step, we combine all power circuit choices with all computer arrangements to get every possible actuator architecture; note that once again, routing rules prevent the EBHAs and the EHA technology choices to combine with $P2$ or $S2$. As each spoiler is moved by one actuator, the list of spoiler-possible architectures is identical to the list of possible actuator architectures with only one computer (fourth rule). As each aileron is moved by two actuators (fifth rule), we need to combine actuator architectures by pairs to obtain all aileron-possible architectures. Here, technological constraints (rules 3, 6, 7, and 8) limit the number of acceptable combinations to 16 APAs. The APA highlighted in the bottom-right portion of Fig. 4 corresponds to $G + (P2 + S2)$ and $E + (P1 + S1)$.

*3. Illustrative Example (Continued)*

By using APA and SPA, the number of possible architectures is considerably reduced. Table 2 displays the results obtained (to be compared with Table 1).

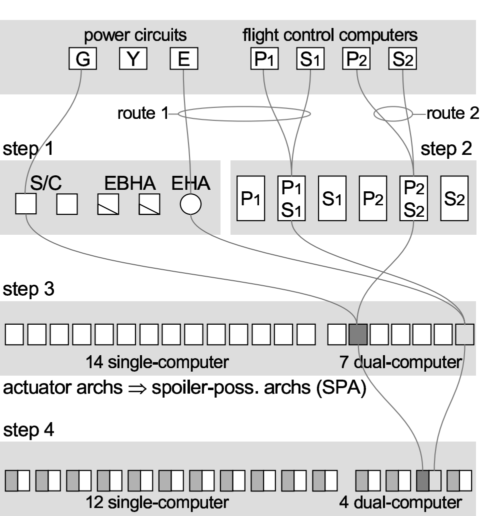


**Fig. 4 Construction of spoiler-possible and aileron-possible architectures on a simplified (fictitious) example.**

**Table 2 Number of architecture combinations on four example architectures**

| | | A320 | $A340_{3H}$ | $A340_{2H2E}$ | A380 |
|---|---|---|---|---|---|
| (new) Number of spoilers | $n_s$ | 4 | 5 | 5 | 6 |
| Number of ailerons | $n_a$ | 2 | 4 | 4 | 6 |
| Number of SPA | $N_s$ | 15 | 15 | 24 | 24 |
| Number of APA | $N_a$ | 54 | 54 | 70 | 70 |
| Possible architectures | N | $<10^9$ | $<10^{13}$ | $<10^{15}$ | $<10^{20}$ |

Note that the fictitious algorithm of Sec. III that would have taken $10^{33}$ years to terminate (without taking into account APA and SPA) would now need 1 min. This confirms the shift in order of magnitude.

*4. Prefilters*

Additional considerations can be expressed at architecture level (outside of APA/SPA), corresponding to designers' practices, to detect architectures that will not pass the safety constraints, including

1) Two ailerons next to each other should have different architectures.

2) Power sources should be reasonably evenly distributed between actuators.

3) Computers should be reasonably evenly distributed between actuators.

4) There should not be three spoilers with the same architectures.

These prefilters allow to reduce further the number of costly safety constraint evaluations by narrowing the search domain.

### B. Problem Solution via Branch and Bound

Once we can fulfill constraint (2) by construction, the problem now corresponds to a black-box constrained allocation problem: find optimal APA/SPA for each aileron/spoiler so as to minimize the weight criterion, under the safety constraints (considered as a black box).

We propose to solve this problem via a specialized adaptation of branch-and-bound methods [2,3]. Branch and bound involves an intelligent search of a tree of possibilities. It can practically be considered as the only deterministic optimization method that provides guaranteed optima for generic combinatorial optimization problems. Indeed, there are other efficient deterministic methods, but their application is restricted to specially structured problems such as integer linear programming, network and graph problems, dynamic programming, knapsack problems, and even allocation problems. However, our allocation problem features a particularly hard safety constraint which is given under the form of an expensive black box. Direct applications of branch and bound in aeronautics can be found in [4,5].

The method starts at the top node, from the reference architecture $A_0 = R = [r_1, \ldots, r_n]$ and tries the various possible architectures $a_1 = 1, \ldots, N_x$ for the first control surface ($N_x$ is $N_a$, the number of APA, if the first control surface is an aileron). This results in a partial architecture $A_1$. Then, the method successively tries all possible architectures for each control surface as it goes deeper into the tree of architectures. Subtrees are explored selectively as follows:

1) *Objective-function evaluation:* from a node at depth $k$, for which a partial architecture $A_k := [a_1, \ldots, a_k, r_{k+1}, \ldots, r_n]$ is defined, the method determines the completion $A_k^{\min} := [a_1, \ldots, a_k, a_{k+1}^{\min}, \ldots, a_n^{\min}]$ that minimizes weight regardless of the safety constraints. We use the corresponding lower bound $\delta w(A_k^{\min})$ to rank the node. The linear weight model approximation is a key feature for this method; when an architecture is partially defined, $A_k = [a_1, \ldots, a_k, r_{k+1}, \ldots, r_n]$, the weight-minimal completion is readily obtained by choosing all minimal components in rows $k+1, \ldots, n$ from weight matrix $\delta W$:

$$\text{for} \quad i = k+1, \ldots, n, \qquad a_i^{\min} = \min_{1 \le j \le N} \delta W(i, j)$$

The process is illustrated in Fig. 5, in which a larger dot represents a higher weight impact. For instance here, the partial architecture

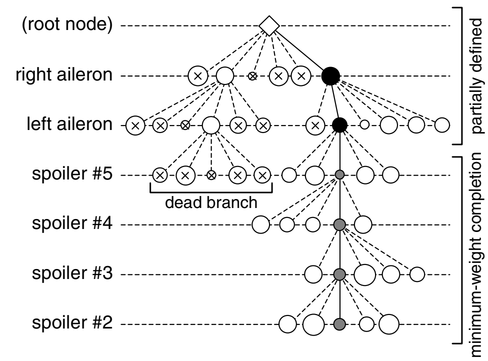


**Fig. 5 Branch-and-bound tree-search example (circle sizes denote weight impact).**

$A_k = [6, 2, r_3, \ldots, r_6]$ is completed by choosing the smallest weight contributions for the remaining control surfaces: $A_k^{\min} = [6, 2, 3, 4, 2, 3]$ represented by dark dots.

2) *Safety constraints evaluation and subtree elimination:* the completed architecture $A_k^{\min}$ is then tested against the safety constraints (or rejected by a prefilter). If it satisfies the constraint $[m(A_k^{\min}) \ge 1]$, then the process eliminates all previously unexplored subtrees for nodes that have a lower bound above $\delta w(A_k^{\min})$, because such solutions cannot be better than feasible solution $A_k^{\min}$.

3) *Choice of new node and branching:* then the algorithm turns to the remaining unexplored subtrees, ranked by the lower bound of their top node. It chooses to explore the subtree with maximum potential, i.e., the one with the lowest lower bound. Let $B_l = [b_1, \ldots, b_l, r_{l+1}, \ldots, r_n]$ be the partial architecture for this chosen node (it may be at another depth in the tree). The algorithm tries a new possible architecture $b_{l+1}$ for control surface $l+1$. Note that prefilters can be used also at this step to reduce the number of possible choices for $b_{l+1}$. The new partial architecture to start from is now $B_{l+1} = [b_1, \ldots, b_l, b_{l+1}, r_{l+2}, \ldots, r_n]$.

Figure 5 exemplifies a tree search at an intermediate step on the A320 architecture, in which a crossed out node denotes a branch cut either by its high lower bound, or because it violates prefilters. It shows that the search is not exhaustive, and that only a fraction of the tree nodes are actually tested.

It is worth mentioning here that the particular choice of tree order (e.g., top level is right aileron, next lower level is left aileron, etc.) has some impact on the optimization performance and is not completely arbitrary. For specific instances, preliminary tests were performed and one of the conclusions was that, as a rule of thumb, it is worthwhile to put at the top nodes the control surfaces having more APAs (or SPAs). For instance, for the $A340_{2H2E}$ instance (24 SPAs and 70 APAs), we would rather put spoilers at the top nodes, because when we cut a branch at the very top, we remove $1/24$ of the whole search domain (against one branch among 70 for ailerons). Moreover, with two actuators each, ailerons are much more reliable than spoilers, and therefore it is less likely that an aileron would be cut as early as a spoiler.

## IV. Computational Experiments

### A. Reference Aircraft A340

The objective of this experiment is to test the algorithm on a reference Airbus aircraft, in the same conditions as a virtual new aircraft project. The reference aircraft chosen for the application is the Airbus A340, with six pairs of spoilers and two pairs of ailerons. We consider two distinct cases: the standard 3H problem, and a more complex 2H-2E problem.

In the 3H case, we consider three hydraulic circuits ($B$, $G$, and $Y$) to power the flight control actuators. This problem is reasonably large ($10^{13}$ possible architectures). It is primarily considered for validation purposes, as the results can be compared to the currently certified A340 3H architecture.

In the 2H-2E case, we have two hydraulic circuits ($G$, $Y$) and two electrical circuits ($E1$ and $E2$). This problem is much larger ($10^{15}$

**Table 3 Comparison between the certified and the optimal architecture for the A340.**

| | Wing surface | Both spoilers | | | | | Left ailerons | | | | Right ailerons | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Number | 2 | 3 | 4 | 5 | 6 | 1 | | 2 | | 1 | | 2 | |
| | Actuator | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| Cert. | circuit | $B$ | $B$ | $\boldsymbol{Y}$ | $\boldsymbol{G}$ | $\boldsymbol{Y}$ | $\boldsymbol{G}$ | $B$ | $Y$ | $G$ | $\boldsymbol{G}$ | $B$ | $Y$ | $G$ |
| | FCC 1 | $P_3$ | $S_2$ | $P_2$ | $P_1$ | $S_1$ | $\boldsymbol{P_1}$ | $\boldsymbol{P_2}$ | $P_3$ | $S_1$ | $\boldsymbol{P_1}$ | $\boldsymbol{P_2}$ | $\boldsymbol{S_2}$ | $\boldsymbol{P_3}$ |
| | FCC 2 | —— | —— | —— | —— | —— | $\boldsymbol{S_1}$ | $\boldsymbol{S_2}$ | —— | —— | $\boldsymbol{S_1}$ | $\boldsymbol{S_2}$ | —— | —— |
| Opt. | circuit | $B$ | $B$ | $\boldsymbol{G}$ | $\boldsymbol{Y}$ | $\boldsymbol{G}$ | $\boldsymbol{Y}$ | $B$ | $Y$ | $G$ | $\boldsymbol{Y}$ | $B$ | $Y$ | $G$ |
| | FCC 1 | $P_3$ | $S_2$ | $P_2$ | $P_1$ | $S_1$ | $\boldsymbol{S_1}$ | $\boldsymbol{S_2}$ | $P_3$ | $S_1$ | $\boldsymbol{P_2}$ | $\boldsymbol{P_3}$ | $\boldsymbol{P_2}$ | $\boldsymbol{P_1}$ |
| | FCC 2 | —— | —— | —— | —— | —— | —— | —— | —— | —— | —— | —— | —— | —— |

possible architectures). As there is no A340 flying with a 2H-2E architecture, this exercise is purely illustrative. However, it provides a way to assess the performance of our methodology for future aircraft projects.

For these computational experiments, our methodology was programmed with MATLABv7, and run on a standard desktop computer (i786 at 1.8 GHz).

## B. Results

### 1. 3H Architecture

The algorithm terminates in 7 min. It requires 2 min to find the exact optimal architecture, and 5 min further to prove its global optimality. Among the $10^{13}$ possible solutions, only 740 are actually explicitly enumerated by the search tree, and the costly safety constraints are evaluated for only nine solutions. The number of stored solutions (standby nodes in the tree search) never exceeds 300. The resulting architecture fulfills the safety constraints and weighs 3.1 kg less than the weight of the reference certified architecture. Table 3 displays the qualitative difference (bold) between the actual A340 architecture and the optimized one.

The results found for the A340 3H problem have been submitted to flight control system experts for criticism. It appears that all formal criteria are captured by the methodology. Only some particular aspects related to particular risks (geometric segregation rules) were found missing. This explains why the optimal solution is lighter than the actual certified A340 architecture. Additional constraints such as the geometric segregation rules can be integrated into our methodology to enhance its validity further.

### 2. 2H-2E Architecture

The algorithm terminates in 25 min. It requires 3 min to obtain the exact optimal architecture, and 22 min further to confirm that this is indeed the global optimum. This problem is 100 times larger than the 3H instance (see Table 2). Yet, the computation time only quadruples. This tends to show that our methodology is relatively robust to combinatorial effects. Among the $10^{14}$ possible solutions, only 2200 were explicitly enumerated, and the safety constraints were evaluated for only 25 solutions. The number of stored solutions (standby nodes in the tree search) never exceeds 700. The resulting architecture, depicted in Fig. 6, fulfills the safety constraints and has a weight within 1% of the best possible weight not subject to safety constraints.

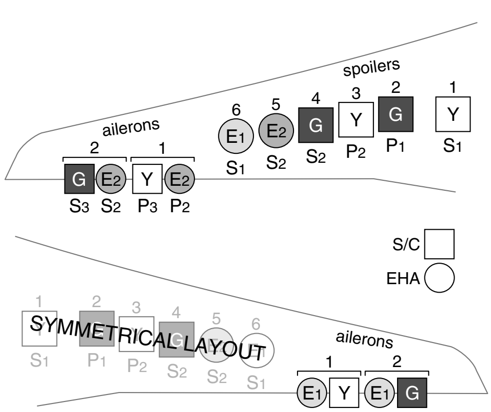


**Fig. 6 Example of a (virtual) 2H-2E architecture for the A340 wing control surfaces.**

## C. Behavior of the Algorithm

The macroscopic behavior is illustrated in Figs. 7 and 8 for the 2H-2E case. In the first phase, the algorithm explores the top tree nodes and eliminates branches having a lower-bound weight that is higher than the current upper bound for the optimal weight. The initialization of this upper bound is therefore important: it should be as low as possible, but higher than the optimal weight. Typically, this upper bound can be determined from any initial feasible solution based on traditional engineering design. The second phase starts when the algorithm finds feasible solutions. This lowers the upper bound even further, and rapidly eliminates a considerable proportion

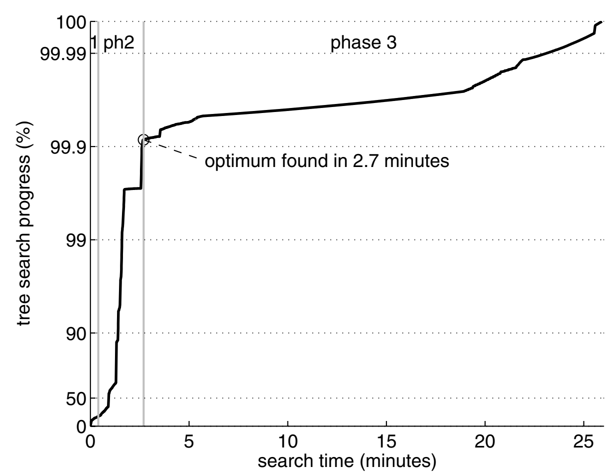


**Fig. 7 Proportion of tree search against time.**

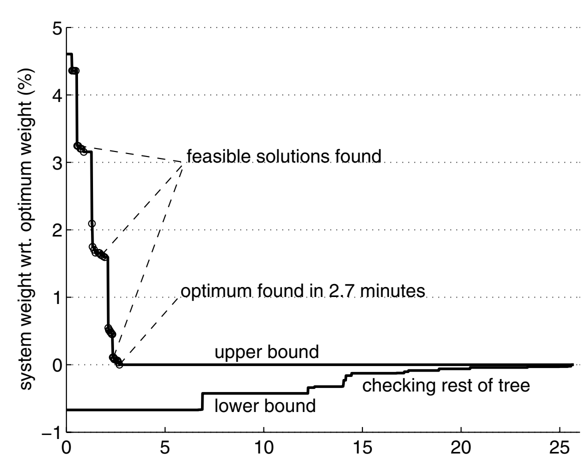


**Fig. 8 Evolution of the upper-bound and lower-bound values (for the optimal weight) against time.**

of branches. New solutions appear rapidly until a very good solution is found. This good solution can be found quite fast.

The final phase is the longest: the algorithm must search the remaining branches to verify that there is no better solution. Although the original tree has been reduced to a very small fraction of its original size, this still represents a relatively large number of architectures to check.

## V. Conclusions

In this paper, we proposed a decision-aid tool for the flight control system architecture design at the early stages of the project. This tool is based on a discrete optimization process that minimizes the weight subject to costly safety and technological constraints. It includes two steps. The first step drastically reduces the initial combinatorial complexity by taking advantage of technological constraints and in-house design rules. The second step uses an astute adaptation of a branch-and-bound search algorithm to find an optimal architecture. This methodology was validated on Airbus A340, for which we obtained very encouraging results. For example, an exact optimal roll architecture was found among $10^{14}$ possibilities in less than 25 min on a standard desktop computer.

In the proposed optimization process, the control surface sizes were considered frozen. One promising research avenue is to include parameters such as position, chord, and length within a global bilevel optimization process. This is the subject of an ongoing study combining stochastic global optimization methods and multi-criterion optimization approaches with deterministic branch and bound for speeding up the subtree selection. Preliminary results in this direction anticipate weight gains around 20–40%.

## References

[1] Boiffier, J.-L., *Dynamics of Flight: the Equations*, Wiley, New York, 1998.
[2] Bradley, S. P., Hax, A. C., and Magnanti, T. L., *Applied Mathematical Programming*, Addison–Wesley, Reading, MA, 1977.
[3] Papadimitriou, C. H., and Steiglitz, K., *Combinatorial Optimization: Algorithms and Complexity*, Prentice–Hall, Upper Saddle River, NJ, 1982.
[4] Mongeau, M., and Bès, C., "Optimization of Aircraft Container Loading," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 39, No. 1, 2003, pp. 140–150.
[5] Mongeau, M., and Bès, C., "Aircraft Maintenance Jacking Problem via Optimization," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 41, No. 1, 2005, pp. 99–109.