New Modular Product-Platform-Planning Approach to Design Macroscale Reconfigurable Unmanned Aerial Vehicles

Souma Chowdhury*
University at Buffalo, Buffalo, New York 14260
Victor Maldonado†
University of Texas at San Antonio, San Antonio, Texas 78249
Weiyang Tong‡
Syracuse University, Syracuse, New York 13244
and
Achille Messac§
Mississippi State University, Mississippi State, Mississippi 39762

DOI: 10.2514/1.C033262

The benefits of a family of macroscale reconfigurable unmanned aerial vehicles to meet distinct flight requirements are readily evident. The reconfiguration capability of an unmanned-aerial-vehicle family for different aerial tasks offers a clear cost advantage to end users over acquiring separate unmanned aerial vehicles dedicated to specific types of missions. At the same time, it allows the manufacturer the opportunity to capture distinct market segments, while saving on overhead costs, transportation costs, and after-market services. Such macroscale reconfigurability can be introduced through effective application of modular product-platform-planning concepts. This paper advances and implements the Comprehensive Product Platform Planning framework to design a family of three reconfigurable twin-boom unmanned aerial vehicles with different mission requirements. The original Comprehensive Product Platform Planning method was suitable for scale-based product-family design. In this paper, important modifications to the commonality matrix and the commonality constraint formulation in Comprehensive Product Platform Planning are performed. These advancements enable the Comprehensive Product Platform Planning to design an optimum set of distinct unmanned-aerial-vehicle modules, different groups of which could be assembled to configure twin-boom unmanned aerial vehicles that provide three different combinations of payload capacity and endurance. The six key modules that participate in the platform planning are 1) the fuselage/pod, 2) the wing, 3) the booms, 4) the vertical tails, 5) the horizontal tail, and 6) the fuel tank. The performance of each unmanned aerial vehicle is defined in terms of its range per unit fuel consumption (miles/gallon). It is found that, when the average unmanned-aerial-vehicle performance (miles/gallon) and the commonality among the unmanned-aerial-vehicle variants are simultaneously maximized, a one-third reduction in the number of unique modules is accomplished at a 66% compromise in performance. On the other hand, when simultaneously maximizing performance and minimizing costs, the best tradeoff unmanned-aerial-vehicle-family designs provide a remarkable 26% reduction in cost for a 6% compromise in performance. In this case, the cost savings are attributed to both material reduction and increased module sharing across the three unmanned-aerial-vehicle variants. It is also observed that, among the best tradeoff unmanned-aerial-vehicle families, the individual unmanned aerial vehicles are most likely to share the horizontal tail and tail booms, and are least likely to share the wing.

Nomenclature

C_F	=	fuselage module cost
C_{FT}	=	fuel-tank module cost
C_{HT}	=	wing horizontal-tail cost
C_r	=	unmanned-aerial-vehicle reference cost
C_{TB}	=	tail-boom module cost
C_{UAV}	=	unmanned-aerial-vehicle configuration cost
C_{VT}	=	vertical-tail module cost
C_W	=	wing module cost
d_F	=	fuselage maximum diameter
E	=	endurance
F	=	fuel-tank size
f_{coml}	=	interproduct-commonality objective
f_{cost}	=	unmanned-aerial-vehicle-family-cost objective
f_{perf}	=	unmanned-aerial-vehicle-family-performance objective
g_j	=	inequality constraints
h_{cc}	=	commonality constraint
h_j	=	equality constraints
L_B	=	tail-boom length
L/D	=	lift-to-drag ratio
L_F	=	fuselage length
M_{fuel}	=	fuel mass fraction
N	=	number of product variants
n_k	=	number of parts in the kth product
R	=	range
R_λ	=	rank of commonality matrix λ
S_{HT}	=	horizontal-tail planform area
S_{VT}	=	vertical-tail planform area
S_W	=	wing planform area
U_∞	=	cruise speed
u	=	number of unique parts in product family
V_{fuel}	=	volume of fuel consumed
η_p	=	propeller efficiency
λ	=	commonality matrix

Received 3 November 2014; revision received 3 July 2015; accepted for publication 13 September 2015; published online 27 January 2016.

*Assistant Professor, Department of Mechanical and Aerospace Engineering. Senior Member AIAA.

†Assistant Professor, Department of Mechanical Engineering; victor.maldonado@utsa.edu. Senior Member AIAA (Corresponding Author).

‡Ph.D., Department of Mechanical and Aerospace Engineering. Student Member AIAA.

§Professor, Department of Aerospace Engineering. Lifetime Fellow AIAA.

I. Introduction

SOPHISTICATED and expensive systems, such as unmanned aerial vehicles (UAVs), which are also required to perform different types of operations (missions), can uniquely benefit from the flexibility to be readily reconfigurable. Modular product-platform-planning concepts provide a unique opportunity to design such reconfigurable systems — as a more streamlined and less expensive alternative to having multiple dedicated systems for diverse missions or operations. Such a design strategy (that allows functional flexibility) thereby provides additional cost savings to the manufacturer and the user, on top of the typical overhead savings generally attributed to product-family design (PFD). In this context, *macroscale reconfiguration* refers to the reassembly of a set of modules before a mission. This reassembly creates a system that is optimally suited to the mission at hand, in which the mapping between mission categories and optimal configurations is defined by the PFD process.

In this paper, a modular product-platform-planning approach is developed and applied to design a family of three "twin-boom" UAVs that offer three distinct combinations of payload capacity and endurance (targeted toward civilian applications). Examples of industries in which such a UAV family could provide unique benefits are the offshore petroleum industry and the offshore wind-energy industry, specifically deepwater platforms. The modular UAV platform-planning process is performed by advancing the Comprehensive Product Platform Planning (CP^3) method, from designing typical scale-based families to designing both scale-based and modular families. The overall goal of this process is to design a set of modules, some of which can be shared by three different UAV configurations — thereby requiring a significantly smaller set of modules to execute three different missions than that required by three dedicated UAVs. The optimized platform designs are derived from a baseline "twin-boom" UAV design. In doing so, the following important question is also explored: What are the cost and mission benefits (or tradeoffs) of a macroscale reconfigurable UAV family compared to a set of dedicated UAVs suited to different applications?

In the next paragraphs, a survey of product-family concepts in aircraft design (from literature and industry) is provided, followed by a brief introduction to modular PFD.

Product-family concepts are utilized by aircraft manufacturers in order to design a series of multimission-capable aircraft with superior performance at a lower cost. Unlike the conventional application of design-optimization techniques to optimize a single aircraft for a specific mission, in this case, a family of aircraft designs is optimized with a certain degree of commonality among them, while interchanging key components to satisfy a wide range of mission requirements. Historically, this has been accomplished though derivatives or variants of the baseline aircraft. For example, the original Boeing 737-100, which first flew in 1967, has evolved (through 11 major design variants in 39 years) to increase passenger capacity, fuel efficiency, and flight range. However, despite the steady increase in performance, the Boeing 737 series continues to operate primarily domestic routes. The goal of modern product-family methods is to design aircraft with a significant variation in performance to serve multiple market segments (i.e., domestic and transatlantic routes). Such a motivation is discussed in the study by [1] to design a family of two blended-wing–body aircraft with a capacity of 272 and 475 passengers with built-in commonality. Other noteworthy investigations include the use of decomposition-based methods [2] and genetic-algorithm techniques [3] for aircraft family design.

In recent years, the academic community has seen an explosive growth in research toward UAVs fueled by sharp sales projections from a nascent civilian UAV market. The industry now seeks to develop unmanned aircraft for a wide range of applications and mission profiles. This presents a unique opportunity to use product-family methods to design a modular UAV family that simultaneously meets the needs of diverse customer requirements, while reducing design and fabrication costs to the manufacturer. A methodology for the design of a two-UAV family operating under aerial firefighting in the vicinity of the Greek islands and maritime surveillance off the coast of Norway is described by Cabral and Paglione [4]. However, although the applications are different, both missions require long endurance monitoring with similar camera payloads. Limited research has been done in leveraging modular or module-based product-family concepts to allow reconfiguration of macroscale aircraft features. One unique example is the work by Pate et al. [5], in which a family of reconfigurable aircraft was designed through interchangeable wings and engines. In the current paper, a family of three UAVs is considered, which is designed to fulfill missions with distinct endurance and payload/weight requirements. Moreover, it becomes more profitable to both the manufacturer and the consumers (or end users) when the UAV family is designed for industries in which the same end user can take advantage of the modular design and use all three UAV configurations.

PFD methods can be broadly classified into scale-based methods and module-based methods. In scale-based or scaling product families, all the product variants are composed of the same set of modules/components, and product variation is attributed to the scaling of the nonplatform design variables that define the product modules (or the overall product configuration). In a module-based product family, distinct modules are added or substituted (to a common platform) to develop different products [6,7]. A popular example of a modular product family is the series of Sony Walkman [8,9], whereas a standard example of a scalable product family is Boeing's 777 aircraft series [10]. Owing to the distinct mathematical characteristics of a modular product family and a scaling product family, a majority of the scale-based PFD methods do not readily apply to designing modular families.

One of the popular approaches to module-based PFD conceptually divides the process into the following three levels: 1) architectural level: to establish a system structure and its variations; 2) configuration level: to establish standard configuration(s), and its variations of products and modules; and 3) instantiation level: to develop a practical product family through variable quantification and combinatorial selection of the modules. In this paper, the instantiation level of modular PFD has been particularly addressed. The instantiation task level is composed of the following two phases: 1) variable quantification: to develop modules across product prototypes by quantifying design variables, and 2) combinatorial selection: to develop product prototypes by selecting desirable combinations from the feasible ones. Based on these phases, approaches to the instantiation task level can be divided into the following three classes: 1) optimization of module attributes under a fixed module combination, 2) optimization of module combinations using predefined module candidates, and 3) simultaneous optimization of module attributes and module combinations. The majority of the existing modular PFD approaches require specifying the platform (fixed module combination, i.e., class 1) before optimization, to reduce the design space and render the problem computationally more tractable. Most other modular PFD approaches are geared toward class 2 optimization problems (e.g., [11]). The assumptions involved in these two classes may lead to suboptimal module-based product families. Very few optimization approaches exist to solve class 3-type optimization problems, such as developed by Fujita and Yoshida [12,13].

Several other well-known methods exist in modular PFD, such as those presented by Stone et al. [14], Dahmus et al. [15], Guo and Gershenson [16], Jose and Tollenaere [17], Kalligeros et al. [18], Sharon et al. [19], Yu et al. [20], and Lewis [21]. In this paper, fundamental advancements are made to a powerful scaling PFD method, called the CP^3 framework [22,23], to enable it to design modular product families. A brief description of the original CP^3 method and of the major modifications made to this method (in this paper) is provided later.

Before delving further into the PFD method that is considered, modified, and implemented in this paper, it is important to precisely define some of the important PFD-related terms that are frequently used in this paper. A review of the vast amount of product-family literature from the past three decades suggests that most of these terms have been defined diversely [24], and hence, the definitions provided as follows (some of which quoted from existing literature) are such chosen that they most closely reflect the context in which these terms have been used in this paper:

1) *Product platform* is "a set of common components, modules, or parts from which a stream of derivative products can be efficiently developed and launched" [25].

2) *Product architecture* is "a) the arrangement of functional elements, b) the mapping from functional elements to physical components, and c) the specification of the interfaces among interacting physical components" [26].

3) *Product variants* are members of a product family that both share certain common features among themselves, and comprise certain unique features that allow them to address different target markets [e.g., the CL-600, CL-601, and CL-604 in the Bombardier 600 series (family of business jets)].

4) *Module* is a component or a part of a product, which can be either unique in each product variant or shared across multiple product variants (e.g., the GE CF34 turbofan that is shared across different members of the Bombardier CRJ series).

5) *Module combination* refers to a unique collection or a unique set of modules that together comprise a functional product.

The CP^3 framework, introduced by Chowdhury et al. [22] and Messac et al. [27], seeks to coherently address a wide range of PFD scenarios. The CP^3 framework presents a generalized mathematical model of the platform-planning process based on the formulation of a commonality matrix. This model yields a mixed-integer nonlinear programming (MINLP) problem with a large number of binary variables. Originally, Chowdhury et al. [22] developed and implemented a platform-segregating-mapping-function method to convert the MINLP problem into a less expensive continuous optimization problem, and the approximated problem was solved using conventional particle swarm optimization (PSO). A reduction of the high-dimensional MINLP problem into a more tractable MINLP problem was later performed [23], and the reduced problem was solved using a mixed-discrete PSO algorithm [28].

The commonality matrix in the original CP^3 method, as well as other similar commonality formulations [29], does not readily represent the platform plan for modular products, in which each module comprises multiple design variables. This paper modifies the commonality-matrix definition for application to modular families. Subsequently, the commonality constraint, which ensures feasible product-platform plans, is also modified to enable effective sharing of multivariate modules among product variants. More importantly, the scaling attributes of the original CP^3 model are also favorably retained, which is unique in the PFD literature. The important features of this new modular CP^3 model include:

1) This modular platform-planning model facilitates sharing of entire multivariate modules among product variants.

2) Modules are allowed to be included or excluded, based on allowed physical product configurations.

3) If necessary (from a practical manufacturing standpoint), individual design variables within particular modules are allowed to be independently shared or scaled (without necessitating the entire module to be shared or scaled correspondingly).

4) This model enables the simultaneous identification of platforming modules and the determination of the optimal module attributes (design-variable quantification) during the product-family-optimization process.

In this paper, the allowed physical combinations of modules are, however, assumed to be known, based on a preconceived product architecture. In this context, the product architecture defines combination(s) of modules that are physically required for the basic structural and functional feasibility of the product. Developing strategies to investigate the product architecture, and explore product-structure options in that context, are important components of product family and product design research [30–32]. It can be said that allowing the product structure itself to be decided within the process of modular product-platform planning can ideally provide an increased level of flexibility in PFD. However, if the steps of 1) deciding/varying the basic product structure, 2) deciding the module-sharing scheme among product variants, and 3) deciding the individual module configurations are combined, it increases the complexity of the problem formulation and optimization process, and makes the product-platform-planning model dependent on the specific application. Hence, product-platform-planning methods that implement these three steps simultaneously are rare. Deciding the product structure, which might lead to a combination of physically different numbers/types of modules, is allowed in the platform-planning model developed in this paper; however, in situ optimization of the product structure is not within the scope of the optimization method implemented and the case study performed in this paper. This paper particularly focuses on deciding the module-sharing scheme and the design of the individual modules (steps 2 and 3, as stated previously).

The modular CP^3 method developed in this paper is applied to design a family of three fixed-wing (twin-boom type) UAVs that are required to satisfy different endurance and payload specifications. Optimization of the platform-planning process in the new modular CP^3 method is performed using the powerful mixed-discrete PSO algorithm. The following section describes the important advancements made to the CP^3 method to introduce modular PFD capabilities, including the generalized problem formulation for optimizing a modular product family. Section III presents the application of the new modular CP^3 method to design the family of UAVs, starting with brief descriptions of the UAV performance and cost models, and ending with the discussion of the results obtained from the optimal PFD of UAVs.

II. Advancing CP^3 for Modular Products

A. Modification of the CP^3 Model

The original CP^3 framework [22] introduced a compact mathematical model of the PFD problem. The key features of the original scale-based CP^3 model are the following:

1) This model presents a generalized and compact mathematical representation of the platform-planning process, which is independent of any optimization strategy.

2) This model avoids the "all or none" restriction [33]; thereby, allowing any subset of all product variants to share parts (platform parts).

3) This model facilitates simultaneous a) selection of platform/scaling design variables, and b) quantification of the optimal design-variable values.

In the original CP^3 model, commonality was defined strictly in terms of the product design variables (as in most other scale-based PFD methods), which is not directly applicable in the context of designing reconfigurable products. To allow product variants to be readily reconfigured from one another, it is required to design an encompassing set of modules, the subsets of which can be combined to create each product variant. In this case, one product can be reconfigured into another by swapping out some (not all) of the parts/modules, in which the remaining modules are ones that are shared across these two products. Therefore, a product-platform-planning process in this context should seek to increase the proportion of parts/modules that do not need to be swapped out during reconfiguration (common modules); increasing commonality, however, must be accompanied with preserving or maximizing the performance of each product, and meeting individual product specifications.

The configuration of a part or module of a product can be (and is more often than not) defined in terms of multiple design variables for example, the fuselage or pod of a UAV is defined in terms of diameter/cross section and length. Now, in seeking to increase commonality, scale-based PFD methods could lead to two UAV variants having the same fuselage length, but different fuselage diameters; from a reconfiguration perspective, such a design platform is not beneficial, because the fuselage is not shared in totality between the two UAVs, and will still need to be swapped out when converting one UAV to another. To summarize, scale-based PFD methods are not cognizant of the connection or correlation among design variables in how they physically define module configurations. In contrast, module-based PFD methods directly encourage commonality across sets of design variables that comprise different modules. Hence, in the process of increasing commonality, a modular PFD approach is expected to drive the values of both the fuselage variables (length and diameter) of each UAV toward common values — potentially leading to both UAVs sharing the same fuselage that does not need to be swapped out during reconfiguration.

In this paper, flexibility is provided in defining commonality among products, in terms of both individual design variables and modules (in which each module can be a collection of variables). "A product platform is said to be created when more than one product variant in a family shares a particular part." In this case, a part can be both a module and an individual design variable. Based on this concept, the commonality among modular products is concisely represented using the generalized matrix, called commonality matrix, represented by λ. The commonality matrix provides a compact representation of the product-platform plan (i.e., the module- or attribute-sharing scheme), and allows the construction of a tractable constraint that readily guides the creation of feasible platform plans during the optimization process. In addition, the CP^3 commonality matrix provides the flexibility to include both scaling and modular features in the product platform. It is important to note that, although the original CP^3 method [22] made novel advancements to the definition and usage of the commonality matrix, the commonality-matrix concept has also been proposed and used by others [29] in the PFD literature.

For ease of illustration, the commonality matrix is represented in terms of module sharing/variation. The commonality matrix for a family of N products, comprising a maximum of m modules, is given by

$$\lambda = \begin{bmatrix} \lambda_1^{11} & \cdots & \lambda_1^{1N} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_1^{N1} & \cdots & \lambda_1^{NN} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \vdots & \lambda_i^{11} & \cdots & \lambda_i^{1N} & \vdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \vdots & \lambda_i^{N1} & \cdots & \lambda_i^{NN} & \vdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_m^{11} & \cdots & \lambda_m^{1N} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_m^{N1} & \cdots & \lambda_m^{NN} \end{bmatrix}$$

$$\lambda_i^{kl} = \left\{ \begin{matrix} 1, & \text{if } \lambda_i^{kk} = \lambda_i^{ll} = 1 \quad \text{and} \quad X_i^k = X_i^l \\ 0, & \text{othrwise} \end{matrix} \right\} \quad \forall\, k \neq l$$

$$\lambda_i^{kk} = \begin{cases} 1, & \text{if the } i\text{th module is included in product } k \\ 0, & \text{if the } i\text{th module is not included in product } k \end{cases}$$

$$i = 1, 2, \ldots, m, k, \qquad l = 1, 2, \ldots, N \tag{1}$$

In this matrix definition, the generic vector $\boldsymbol{X}_i^k$ represents the ith module in product k (i.e., the vector of variables comprising the ith module in product k). The commonality matrix is a symmetric block diagonal matrix, in which the ith block corresponds to the ith module. The binary variables (λ_i^{kk}) are called module-inclusion variables. In the case of the original CP^3 model, all the diagonal elements (λ_i^{kk}) are fixed at one — because all product variants were composed of the same set of physical design variables. However, in this new commonality matrix, the diagonal elements (λ_i^{kk}) can be allowed to vary during optimization, depending on the product architecture.

In the original CP^3 commonality matrix, each off-diagonal element is a binary variable (scalar values) that denotes whether a design variable is shared or varied across the product variants. In this new commonality matrix, each off-diagonal element can be both a scalar binary variable and a vector of binary variables. In this case, the off-diagonal elements of the commonality matrix, (λ_j^{kl}), determine whether the ith module is shared by product k and product l, in which $\lambda_i^{kl} = 1$ if shared, and $\lambda_0^{kl} = 0$ if not shared. If certain physical design variables comprising the modules need to be shared or scaled independently (for an application), the corresponding module-based commonality blocks can be readily expanded into subblocks to account for such a scenario. In that case, the generic $\boldsymbol{\lambda}_i^{kl}$ is a vector, in which λ_{ij}^{kl} determines whether the jth variable in the ith module is shared by product k and product l. Thus, by treating $\boldsymbol{\lambda}_i^{kl}$ or λ_{ij}^{kl} as variables during the product-family optimization, the new CP^3 method uniquely allows both modular and scaling attributes to be incorporated into the product platforms being designed. In the context of optimization, these binary variables (or variable vectors) are termed as *commonality variables*, to distinguish them from the physical design variables that define the product configurations.

Using the modified commonality-matrix definition, the new commonality constraint for modular products is formulated. The commonality constraint ensures compatibility between the product-platform plan and the physical design of each product. In the original (scale-based) CP^3 method, the commonality constraint would tend to penalize a candidate product family only when the design variables are not shared across product variants. Here, the commonality constraint is required to be such formulated that it tends to penalize a candidate family of reconfigurable products, anytime the entire modules are not shared across product variants (irrespective of whether individual variables are shared or not). The new commonality constraint is thus given by

$$\sum_{i=1}^{m} \sum_{\forall\, k \neq l} \mu_i^{kl} = 0$$

in which

$$\mu_i^{kl} = \begin{cases} \lambda_i^{kl} |X_i^k - X_i^l|^2, & \text{if } \lambda_i^{kk} = \lambda_i^{ll} = 1 \\ 0, & \text{otherwise} \end{cases} \qquad k, l = 1, 2, \ldots, N \tag{2}$$

In this equation, $|X_i^k - X_i^l|^2 = (X_i^k - X_i^l)^T (X_i^k - X_i^l)$. The parameter m represents the maximum number of physical modules that can form a single product variant. It is evident from Eq. (2) that each term in the commonality constraint, μ_i^{kl}, becomes zero only if 1) the ith module is not included in one or both products — in which case sharing is not possible; 2) the ith module is shared by product k and product l; or 3) the commonality variable is equal to zero (i.e., $\boldsymbol{\lambda}_i^{kl} = 0$).

The first case shows that, when the module itself is not present in one or both of the products being compared, the commonality matrix does not penalize the platform plan. In the third case, the ith module is generally not shared by product k and product l.

The process of testing whether a product family, comprising N products and a maximum of m modules, satisfies this commonality constraint is explained by the flowchart in Fig. 1. In this figure, the black arrows represent the process direction and the gray arrows represent the flow of information (on an as-needed basis). In Fig. 1, the parameter M is equal to $N(N-1)/2$; the tolerance parameter ϵ is used to relax the equality criterion into an inequality criterion — to allow manufacturing tolerances and/or to ease the optimization process. In Fig. 1, the generic parameter $\boldsymbol{X}^k$ represents the overall design vector of product k.

The provision of the module-inclusion variables (λ_i^{kk}) and the formulation of the commonality constraint [Eq. (2)] uniquely allow for the consideration of different product structures within the product-platform-planning process. In that case, feasible module combinations (product structure) will be guided by the product architecture that defines the module interactions, that is, the structural/functional interdependencies among modules for example, 1) if module A is included, module B cannot be included; or 2) module A can be included only when modules B and C are included. Such dependencies can be implemented as constraints in the optimal

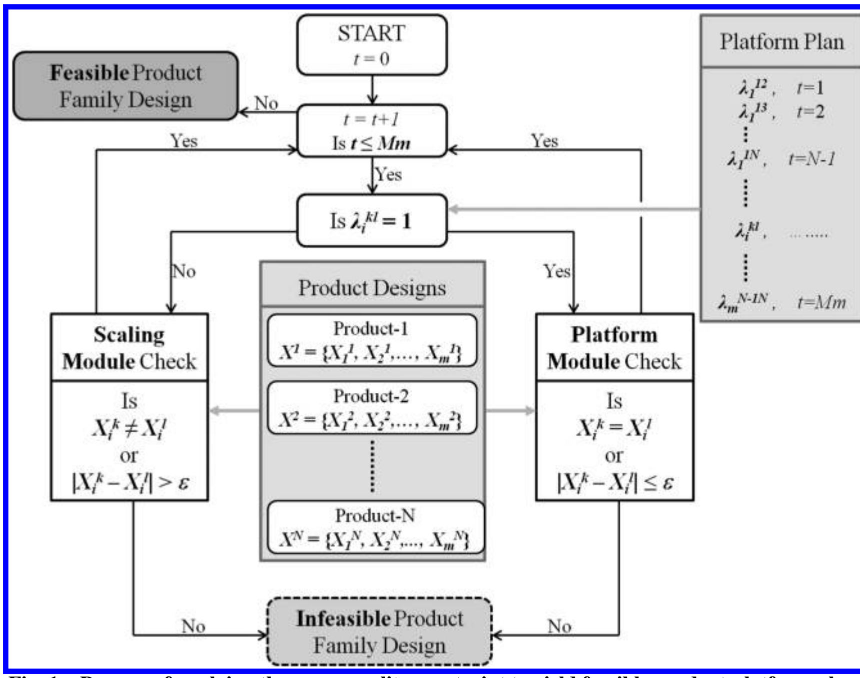


Fig. 1 Process of applying the commonality constraint to yield feasible product-platform plans.

search process. If the products' structure is so allowed to vary during CP^3, candidate PFDs might include products that comprise differing numbers of modules and/or physically different modules (e.g., UAVs with different on-board sensor systems) — the design vectors representing candidate designs cannot be readily compared in that case. It is therefore important to note that adopting such a product-platform-planning strategy (although ideal in theory) would severely increase the challenges encountered in the optimization process, and would very likely require constructing and solving a complex decomposed problem. Therefore, for the optimization-problem formulation presented (Sec. II.C), the optimization method adopted, and the case study performed (Sec. III) in this paper, the structure of the product (fixed-wing UAVs) is assumed to be known/fixed.

B. Commonality Objectives

In the PFD literature, the minimization of the overhead costs (through product-platform planning) is often substituted by the maximization of a commonality measure/metric that represents the net degree of commonality. This substitution is especially helpful in the absence of reliable application-specific cost models or cost data. Among the several proposed metrics that provide a measure of tooling cost savings attributed to component sharing, the commonality index (CI) developed by Martin and Ishii [34] has been reported to be an effective metric [29]. This CI is essentially based on the ratio of the number of unique parts to the total number of parts in the product family. For a family of N product variants, the CI can be mathematically defined as

$$\mathrm{CI} = 1 - \frac{u - \max(n_k)}{\sum_{k=1}^{N} n_k - \max(n_k)} \tag{3}$$

in which u represents the actual number of unique parts in the whole product family, and n_k represents the number of parts in the kth product. The $-\max(n_k)$ term is included in the definition to ensure that the CI varies between 0 and 1. The total number of unique parts in a product family is equal to the rank of the commonality matrix in CP^3 [35]. Hence, a more generalized definition of the CI is given by

$$\mathrm{CI} = 1 - \frac{R_\lambda - m}{\sum_{k=1}^{N} n_k - m} \tag{4}$$

in which R_λ is the rank of the commonality matrix λ; n_k is the number of modules in the kth product; and $n_k = m, \forall\ k$, when all the product variants are composed of the same set of physical modules. During optimal PFD, the commonality objective is defined to be equal to the CI (i.e., $f_{\mathrm{com1}} = \mathrm{CI}$).

C. Optimization-Problem Formulation

Every candidate product-platform plan (in CP^3) involves a large number of binary integer variables — the commonality variables and the module-inclusion variables. Each block of the commonality matrix λ_i, corresponding to a module, is composed of 1) at least $N(N-1)/2$ commonality variables: λ_i^{kl} ($\forall\ k \neq l$), and 2) at most, N module-inclusion variables: λ_i^{kk}.

In practice, the number of module-inclusion variables will be less than N, due to likely prior knowledge that certain modules cannot be included/excluded from certain product variants (depending on the product architecture).

Owing to the transitivity constraints [29,35], the commonality variables (λ_i^{kl}) are, however, not necessarily independent of each other. The set of feasible commonality matrices can be readily identified by applying the transitivity constraints to all the possible commonality-matrix variations, as reported by Chowdhury et al. [23]. In the case of a module-based family, the possibility of module inclusion/exclusion/substitution poses additional constraints on the commonality variables, which can be intuitively defined as

$$\lambda_i^{kl} \leq \lambda_i^{kk} \lambda_i^{ll}, \quad \forall\ i = 1, 2, \ldots, m; \qquad k, l = 1, 2, \ldots, N \tag{5}$$

This constraint, to be called the *module-inclusion constraint*, ensures that a module-sharing scheme ($\lambda_i^{kl} = 1$) is possible between any two products, if and only if the concerned module is included in both products (i.e., $\lambda_i^{kk} = \lambda_i^{ll} = 1$).

In a scale-based family, the diagonal elements of the commonality matrix are all equal to 1. With this consideration, Chowdhury et al. [35] aggregated the $N(N-1)/2$ binary commonality variables (in each commonality-matrix block) into a single binary string of length, $L = N(N-1)/2$; this string was subsequently converted into an integer variable. Therefore, the original CP^3 method yielded one integer variable (defining the platform plan) for each part that participated in the platform planning. For a modular family, some of the diagonal elements of the commonality matrix (module-inclusion variables) could be unknown. Secondly, in a scale-based family, each product variant is assumed to be composed of the same set of design variables or parts (i.e., $N_i = N$ for any ith part). This condition does not hold in the case of a generalized modular family, in which instead

$N_i \leq N$. Subsequently, the ranges of the integer variables associated with different modules could be significantly different, leading to differing rates of evolution during the optimization search process. Hence, a new aggregation of the binary variables is defined as follows:

$$z_i = s_1^i \times 2^{L_i - 1} + s_2^i \times 2^{L_i - 2} + \ldots + s_{L_i}^i \times 2^0 \tag{6}$$

in which

$$L_i = N_i + N_i(N_i - 1)/2; \qquad s_i \in \{0, 1\}; \qquad i = 1, 2, \ldots, L_i$$

In the preceding equation, s_j^i is the jth element of the binary string that first includes the unknown module-inclusion variables (λ_i^{kk}), followed by the unknown commonality variables (λ_i^{kl}) for the ith module.

Therefore, for a product family comprising a maximum of m modules, the commonality matrix is replaced by a tractable set of m integer variables, to be known as *integer commonality variables*. Each of these integer commonality variables (z_i) is allowed to take integer values in the range $[0, L_i]$. The integer values (in this range) that correspond to infeasible combinations of the binary variables can be eliminated from the allowed set before optimization, thereby easing the optimization process significantly. Infeasible combinations of the binary variables (i.e., infeasible commonality matrices) are attributed to violation of 1) the transitivity constraints, 2) the module-inclusion constraints, or 3) any known intermodule dependencies. The creation of the set of allowed integer values before optimization relieves the application of these constraints during the optimization process, and reduces the optimization complexity — a unique attribute of the CP^3 method. It is important to note that, when intermodule relationships are considered, the allowed integer values can become combinatorial in nature.

The generalized objectives of product-family optimization can be stated as 1) maximization of the product performances, and 2) maximization of the net interproduct commonality (or cost minimization), while ensuring that the individual products satisfy their specified design requirements — essentially a constrained multiobjective optimization problem. The performance objective depends on the class of products and the user/designer standpoint, and is usually reflective of the product quality. The second objective in this case is given by the CI defined in Eq. (4). The specified design requirements can be generally modeled as constraints in the optimization problem. The generalized MINLP problem for a modular family of N products, derived from the new CP^3 model, can be expressed as

$$\mathrm{Max}\, f_p(\boldsymbol{X}, \boldsymbol{Y})$$
$$\mathrm{Max}\, f_c(\boldsymbol{Z})$$

subject to

$$\begin{aligned} & g_j(\boldsymbol{X}, \boldsymbol{Y}) \leq 0, \qquad j = 1, 2, \ldots, p \\ & h_j(\boldsymbol{X}, \boldsymbol{Y}) = 0, \qquad j = 1, 2, \ldots, q \\ & h_{cc}(\boldsymbol{X}, \boldsymbol{Z}) = 0 \end{aligned} \tag{7}$$

in which

$$X = [X_1^1 \ \ldots \ X_1^N \ \ldots \ X_i^1 \ \ldots \ X_i^N \ \ldots \ X_m^1 \ \ldots \ X_m^N]^T$$
$$Z = [z_1 \ \ldots \ z_i \ \ldots \ z_m]$$
$$k, l = 1, 2, \ldots, N; \qquad i = 1, 2, \ldots, m$$

In Eq. (7), f_p and f_c are the objective functions that represent the performance and the interproduct commonality [given by Eq. (4)] of the product family, respectively; the generic terms g_j and h_j, respectively, represent the inequality and the equality constraints related to the physical design of the products; and the equality constraint h_{cc} represents the commonality constraint given by Eq. (2). In Eq. (7), the generic vector $\boldsymbol{X}_i^k$ represents the design vector for the ith module in the kth product, which considers this module to participate in platform planning and allows it to be defined in terms of multiple design variables; and $\boldsymbol{Y}$ represents the vector of physical design variables that do not participate in platform planning. The vector $\boldsymbol{Z}$ in Eq. (7) comprises the m integer commonality variables that define the platform plan.

III. Designing a Family of Multimission-Capable UAVs

A. Civilian UAV Applications

A NASA report summarized the key barriers that need to be overcome for UAVs to become viable, cost-effective, and regulated alternatives to current technologies [36] in civilian or nonmilitary applications. Some of these barriers are 1) affordability (price and customization), 2) capacity for payload flexibility, and 3) multimission capability. The development of robust platforms for modular and/or reconfigurable UAVs can offer a powerful solution to these challenges. With this vision, we pursue the design of a family of three modular UAVs with the following potential applications and mission classes: 1) transportation of commercial goods (low endurance: 3–5 h, high payload: 20 lb), 2) environmental survey (high endurance: 22–26 h, medium payload: 7 lb), and 3) search/surveillance (medium endurance: 14–18 h, low payload: 3 lb).

An example of an industry in which such macroscale reconfigurable UAVs are pragmatic is the petroleum industry, specifically the extraction of oil and natural gas from offshore platforms. In such platforms located tens of miles from shore, the supply of goods during the planning and operational stages is a regular occurrence. Although normally supplied by ships, UAVs may be able to rapidly and cost-effectively transport small cargo during routine and emergency situations. Environmental surveying is an important factor to consider before and after securing oil platforms on-site. In this capacity, UAVs can aid in studying the ecological effects (such as the population of fish and water chemistry) surrounding the platform utilizing video cameras and hyperspectral image sensors. Finally, safety is of prime concern and a topic that came to the national spotlight following the explosion of the BP Deepwater Horizon platform off the coast of Louisiana on 21 April 2010. Subsequently, it became the worst oil spill in U.S. history. To help prevent such accidents from happening again, medium-endurance UAVs are envisioned to be deployed as preventative safety measures for surveillance and remote detection of oil leaks in the platform infrastructure. Alternatively, the same UAV can be used for postdisaster relief efforts by searching for survivors and assessing the damage.

B. Twin-Boom UAV Design

The family of fixed-wing UAVs considered in this paper is based on a popular design for UAVs referred to as a twin-boom configuration. Its most distinctive feature is the installation of the engine in the rear of the fuselage/pod, allowing sensitive sensors to be mounted near the nose and away from engine obstruction. The baseline UAV design used to develop the UAV family in this paper is derived from a "twin-boom" UAV configuration. An illustration the conceptual designs (with different payload and endurance capabilities) is shown in Fig. 2.

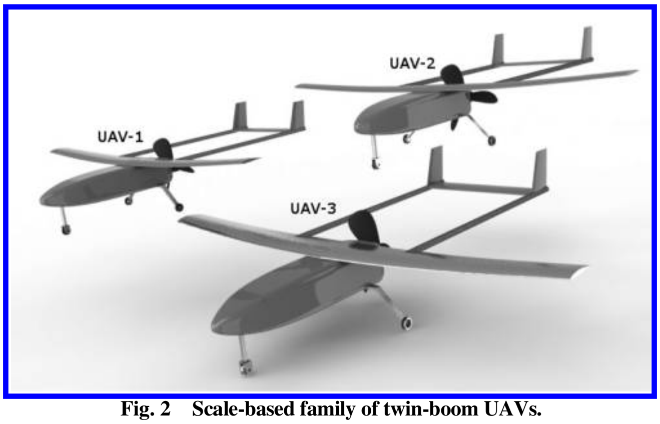


Fig. 2 Scale-based family of twin-boom UAVs.

For given payload and endurance specifications, greater flight range and lower fuel consumption are generally the desirable performance attributes (in UAV design). As such, the net range per unit fuel consumption [in miles/gallon (MPG)] is considered as the primary performance objective (to be maximized) in this paper. The design formulation of the UAV performance relied on accurate approximations of the aircraft's initial and final cruise weight based on the size of the carbon-fiber airframe, four-stroke internal combustion engine, fuel, landing gear and wheels, and typical control avionics. The expression for endurance E governing the efficient flight of the aircraft is commonly known as the Breguet endurance equation [37] for reciprocating engines, which is defined as follows:

$$E = \frac{(L/D)\eta_p}{U_\infty \mathrm{SFC}} \ell n\left(\frac{1}{1 - M_{\mathrm{fuel}}}\right) \tag{8}$$

In this equation, (L/D) is the combined lift-to-drag ratio of the entire aircraft (wing, fuselage, and vertical- and horizontal-tail surfaces), η_p is the propeller efficiency, U_∞ is the cruise speed of the aircraft, SFC is the brake specific fuel consumption of the aircraft engine, and M_{fuel} is the fuel mass fraction. Subsequently, the maximum range of the aircraft in miles (R) and the range per unit fuel consumption in MPG are given by

$$R = E \times U_\infty \quad \mathrm{MPG} = R/V_{\mathrm{fuel}} \tag{9}$$

in which V_{fuel} is the net volume of fuel consumed (expressed in gallons). The performance of the UAV is estimated using a series of analytical and empirical expressions [38,39]. The detailed formulation of the UAV performance and constraints is, however, not within the scope of this paper.

The performance objective f_{perf} is given by the scaled average of the range per unit fuel consumption of the three UAVs:

$$f_{\mathrm{perf}} = \frac{1}{3 \times \mathrm{MPG}_{\mathrm{AS}}} \sum_{k=1}^{3} \mathrm{MPG}_k \tag{10}$$

in which $\mathrm{MPG}_{\mathrm{AS}}$ is the approximate range per unit fuel consumption of the Aerosonde UAV, which is equal to 1350 MPG [40], and MPG_k is the range per unit fuel consumption of the kth UAV. The Aerosonde UAV is used as a benchmark in this case, because it is one of the most well-known UAVs with a twin-boom design.

It is important to note that, here, we have assumed all three UAVs to be of equal importance. In other words, the performance of the three UAVs is equally weighted, assuming they will be subject to comparable number of flying hours over their lifetime. Although practically feasible in the case of small low-altitude UAVs as considered in this paper, this assumption may not necessarily hold for other classes of UAVs or other non-UAV reconfigurable products. However, the formulation of the performance objective is not an integral component of the generalized product-platform-planning model; the performance objective can thus be perceived as a black box in the context of CP^3.

A block diagram, illustrating the modules, the module attributes (physical variables), the operational variables, the constants/ or specifications, and the performance outputs of the UAV performance model, is shown in Fig. 3.

It can be seen from Fig. 3 that each UAV is composed of six modules: 1) wing, 2) fuselage or pod, 3) vertical tails, 4) horizontal tail, 5) booms, and 6) fuel tank. The modules comprise a total of 14 physical design variables (or module attributes), as seen from Fig. 3. In addition, the aircraft cruise velocity is treated as an operational variable, which does not participate in platform planning. It is important to note that, although payload is a performance attribute in practice, it is treated as an input to the UAV performance model — the payload specification is used to calculate the initial and final weights of the UAV. The bounds of the 16 design variables are given in Table 1. In this table, LE sweep angle denotes the leading-edge sweep angle.

The variable bounds are determined based on the baseline UAV designs (Fig. 2). These bounds are expected to allow sufficient flexibility without introducing structural or fabrication issues, which are otherwise not explicitly addressed in the current aerodynamic performance model. The allowed airfoil types are integer coded, and characterized in terms of their maximum thickness and its location,

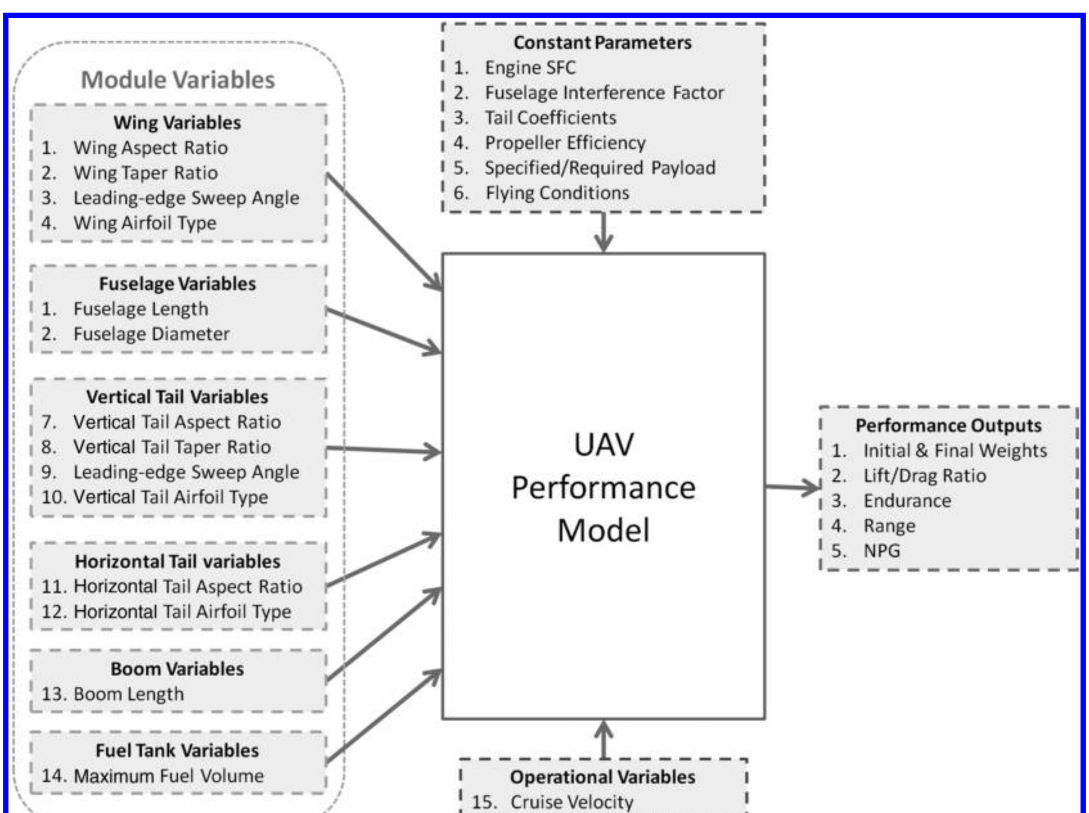


Fig. 3 Block diagram of the UAV module attributes, design constants, and performance criteria.

Table 1 Design-variable bounds of the twin-boom UAVs

Module	Variable	Upper limit	Lower limit
Wing	Aspect ratio	12.0	4.0
	Taper ratio	1.0	0.3
	LE sweep angle, deg	30.0	0.0
	Airfoil type	7	1
Fuselage	Diameter, ft	1.5	0.25
	Length, ft	6.0	1.5
Vertical tails	Aspect ratio	2.0	1.0
	Taper ratio	1.0	0.3
	LE sweep angle, deg	60.0	0.0
	Airfoil type	9	8
Horizontal tail	Aspect ratio	7.0	2.0
	Airfoil type	9	8
Booms	Length, ft	8.0	2.5
Fuel tank	Maximum fuel volume, L	20	0.5
Operational variable	Cruise velocity, ft/s	146.0	58

lift-curve slope, zero-lift angle of attack, and maximum lift coefficient. The wing is allowed to use seven different nonsymmetric airfoils, namely, 1) Eppler 393, 2) NACA 4412, 3) NACA 4415, 4) NACA 2212, 5) Eppler 176, 6) Eppler 393 modified, and 7) Eppler 422. The tail sections are allowed to use two different symmetric NACA airfoils, typical of small- to medium-sized unmanned aircraft, namely 1) NACA 0008 and 2) NACA 0010.

C. UAV-Family Cost Model

The purpose of cost modeling is to estimate the cost of a UAV development program or the cost of an individual UAV. Typically, a life-cycle cost-modeling approach is employed, which consists of development, manufacturing, and operations costs. However, at the platform-planning stage, UAV cost models are based on simplified system parameters known within the conceptual-design framework. This includes variables, such as the UAV's structure weight, engine thrust, intended mission, and payload.

In this investigation, we wish to quantify the cost of optimized UAV configurations with varying degrees of commonality relative to individual UAVs with no shared modules. The cost model developed here considers the weight of the module attributed to the manufacturing material and the hardware associated with each module (i.e., landing gear for the fuselage and linkages). The fabrication material considered is the aerospace grade 12 K biaxial carbon fiber with a weight of 0.0778 $\mathrm{lb/ft^2}$. The hardware weight component is approximated as relatively low fraction (10–20%) of the composite material weight of each module, which is a reasonable assumption. Finally, the module cost is estimated by applying an empty weight cost metric of \$1500 per pound, as suggested by the Office of the Secretary of Defense UAV Roadmap [41] for modern UAVs. This estimate includes the electronics, communications, and propulsion subsystems that are integrated into the modules and make up the empty weight of the UAV. Expressions for the cost of each module are given as follows, in terms of original aircraft-design parameters: 1) wing: $C_W = 1500[0.317S_W + 0.15(0.317S_W)]$, in which S_W is the wing area; 2) fuselage: $C_F = 1500[0.176L_F d_F^2 + 0.2(0.176L_F d_F^2)]$, in which L_F and d_F are the fuselage length and fuselage maximum diameter, respectively; 3) vertical tails: $C_{\mathrm{VT}} = 1500[0.317S_{\mathrm{VT}} + 0.10(0.317S_{\mathrm{VT}})]$, in which S_{VT} is the vertical-tail area; 4) horizontal tail: $C_{\mathrm{HT}} = 1500[0.158S_{\mathrm{HT}} + 0.10(0.158S_{\mathrm{HT}})]$, in which S_{HT} is the horizontal-tail area; 5) tail booms: $C_{\mathrm{TB}} = 1500[0.0916L_B]$, in which L_B is the tail-boom length; and 6) fuel tank: $C_{\mathrm{FT}} = 90F$, in which F is the fuel-tank size in liters.

The cost of each UAV configuration is an aggregate sum of the six module costs (i.e., $C_{\mathrm{UAV}} = C_W + C_F + C_{\mathrm{VT}} + C_{\mathrm{HT}} + C_{\mathrm{TB}} + C_{\mathrm{FT}}$). It is important to note that this model does not account for the payload or sensor costs, which depends on the mission and user requirements, and furthermore cannot be correlated to any degree of accuracy to the UAV weight. The net cost of the three UAV variants when solely optimized for individual performance [42] is treated as the reference cost. The cost objective to be used in UAV-family optimization, f_{cost}, is expressed in this normalized form:

$$f_{\mathrm{cost}} = \frac{\sum_{i=1}^{m} C_{\mathrm{mod}}^{i}}{(C_{\mathrm{UAV1}}^{*} + C_{\mathrm{UAV2}}^{*} + C_{\mathrm{UAV3}}^{*})} \tag{11}$$

in which the generic C_{mod}^{i} represents the cost of the ith unique module in a candidate UAV family, in which the total number of unique modules in the family (m) could be less than three times the number of modules in each individual UAV, and $C_{\mathrm{UAV}k}^{*}$ represents the cost of the kth UAV when individually optimized for maximum performance. The aggregate cost of the three UAVs when they are individually optimized for MPG performance is \$3224.0.

D. Application of CP^3: Family of UAVs

In this paper, CP^3 is applied to design a family of three UAVs with different endurance and payload-capacity specifications. These specifications, which are summarized in Table 2, are derived for the three different civilian applications discussed in Sec. III.A.

In terms of optimization objectives, the following two different paths are pursued toward designing the family of macroscale reconfigurable UAVs:

1) Case 1: Maximize the average range/fuel consumption (MPG) of the three UAVs, and maximize the net commonality (CI) among the three UAVs.

2) Case 2: Maximize the average range/fuel consumption (MPG) of the three UAVs, and minimize the net cost of the three UAVs.

Although maximizing a measure of commonality (e.g., CI) is a popular choice of objective in PFD, in this application, it was realized that the CI objective by itself (without any cost considerations) might lead to economically undesirable designs — a significant portion of the designs could be driven toward relatively large UAV configurations (with wide wingspans) that fulfill both the maximum endurance and maximum payload-capacity specifications (among those listed in Table 2). The fabrication cost of such large UAV configurations, even with more commonality among the three variants, could be significantly high; in other words, the substantial increase in material cost attributed to greater size could very well dominate the cost savings attributed to commonality. By pursuing the two different optimization pathways (defined previously), we therefore get the opportunity to compare and contrast the impact of commonality and cost objectives on the UAV-family design process.

For this application, all the UAV variants comprise the same physical set of modules (Fig. 3). Hence, the module-inclusion variables are known a priori (i.e., $\lambda_i^{kk} = 1,\ \forall\ i,\ k$). The two optimization problems thus formulated are given by

Case1:

$$\mathrm{Max}\ \bar{f} = [f_{\mathrm{perf}}(X, Y), f_{\mathrm{coml}}(Z)]$$

Case2:

$$\mathrm{Max}\bar{f} = [f_{\mathrm{perf}}(X, Y), -f_{\mathrm{cost}}(X, Z)]$$

subject to

$$g_j(X, Y) \le 0, \qquad j = 1, 2, \ldots, p \quad h_{\mathrm{cc}}(X, Z) = 0 \tag{12}$$

in which

Table 2 Design requirements of the UAVs

UAV	Application	Endurance, h	Payload, lb
1	Environmental survey	24	7
2	Search/surveillance	16	3
3	Cargo transportation	4	20

$$X = [X_1^1 \quad \ldots \quad X_1^3 \quad \ldots \quad X_i^1 \quad \ldots \quad X_i^3 \quad \ldots \quad X_6^1 \quad \ldots \quad X_6^3]^T$$
$$Y = U_\infty$$
$$Z = [z_1 \quad \ldots \quad z_i \quad \ldots \quad z_6]; \qquad z_i \in \{0, 1, 2, 4, 7\}$$
$$k, l = 1, 2, 3; \qquad i = 1, 2, \ldots, 6$$

In Eq. (12), the design vector $\boldsymbol{X}$ includes the 14 module attributes/variables in the same order as listed in Fig. 3. In this case, all six modules are included in all three UAVs (i.e., all $\lambda_i^{kk} = 1$); the ensuing allowed values for the integer commonality variables (z_i) for this three-UAV family are shown in Eq. (12). The inequality constraints (g_j) include four primary physical design constraints that address 1) conflicts between fuel volume and fuselage size, 2) conflicts between wing root chord and fuselage size, 3) satisfaction of the required endurance, and 4) avoidance of aircraft stalling. The variable bounds (Table 1) are also formulated as inequality constraints.

It is important to note that the UAV-family design problem is expected to be fairly complex relative to other standard test examples used in the product-family literature (e.g., the universal electric motor [43] and the general-aviation aircraft [44]). In this case, complexity is attributed to the appreciable design dimension (per product), functional nonlinearities, consideration of intermodule relationships, and the number of design constraints. Together with the presence of the multimodal commonality constraint and the integer commonality variables yielded by CP^3, the UAV design complexities present appreciable challenges to the optimization effort. The current product-family-optimization problem involves a total of 45 physical design variables (36 continuous and 9 discrete variables) and 6 integer commonality variables. In this paper, a multiobjective variation of the powerful mixed-discrete particle swarm optimization (MDPSO) algorithm [28] is adopted to solve the challenging optimization problem. The single-objective MDPSO has been previously (successfully) applied to design families of universal electric motors [35]. The nondominated sorting genetic algorithm (NSGA-II) is also applied to solve both MINLP problems. The MDPSO yielded better solutions for case 1 and the NSGA-II yielded better solutions for case 2. The case 1 and case 2 results (from the MDPSO and NSGA-II, respectively) are discussed next.

1. Case 1 (Maximizing Commonality): Results and Discussion

A population size of 200 particles is used in this case study, and the optimization is allowed to run for a maximum of 200,000 function (or system) evaluations, with additional termination criteria similar to that reported by Chowdhury et al. [35]. The values of the prescribed MDPSO parameters and the detailed description of these parameters can be found in the original paper on MDPSO [28]. The equality constraint h_{cc} (commonality constraint) in the optimization [Eq. (12)] is relaxed by a tolerance of $\epsilon = 1.0e\text{-}06$, and converted to an inequality constraint (i.e., $g_{cc} = h_{cc} - \epsilon \le 0$).

The MDPSO is run multiple times, yielding solutions acceptably close to each other. The Pareto solutions obtained by one of the representative runs of MDPSO are shown in Fig. 4. In this case, the principle of strict dominance is used to compare solutions and create the Pareto front — depicted by the blue circle symbols. As a result, solutions with the same value of the CI and different values of performance are included in the Pareto front. The Pareto solutions derived from the weak-dominance principle are retagged by the red X symbol in this figure.

It is seen from Fig. 4 that 29 Pareto points (circles) are yielded by strict dominance, among which seven points (X symbol) form the Pareto front based on weak dominance. The simultaneous exhibition of the weak-dominance and strict-dominance Pareto solutions helps illustrate the discontinuous/discrete nature of the commonality objective (CI). In addition, the weakly dominated solutions (that provide lower performance at the same CI) might provide alternative module-sharing options, which can have advantages from a practical manufacturing perspective.

Theoretically, each module can be shared in five different schemes (five different platform configurations) among the three UAVs. The Pareto designs in this case involved platforming of a maximum of four modules, that is, no more than four modules were ever shared by the UAVs in any scheme among the best tradeoff families. In the best tradeoff family designs, it was found that 1) the fourth module or the horizontal tail is shared the most among any two or all three UAV variants, and 2) the first module or the wing is shared the least. The latter observation is expected, because the aircraft aerodynamic performance is strongly influenced by the wing design; conversely, the wing design is most sensitive to the variations in the specified endurance and payload requirements.

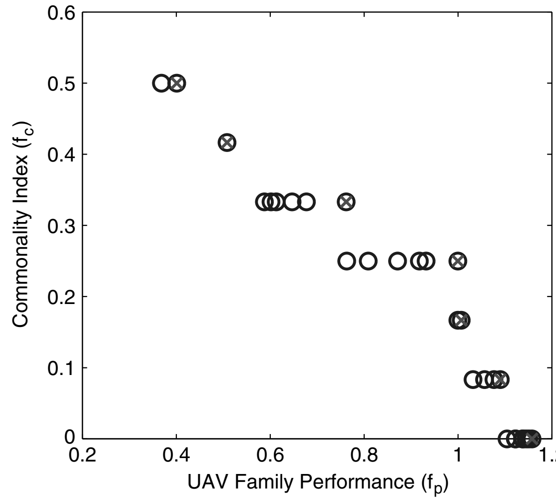


Fig. 4 Best tradeoff UAV families for case 1 (maximizing commonality).

The maximum CI obtained by optimization is equal to 0.5 (leftmost Pareto points in Fig. 4), which corresponds to a UAV family comprising 12 unique modules. This maximum commonality is accomplished at the cost of almost a two-thirds (66%) reduction in the average UAV performance (i.e., average UAV range/fuel consumption). The maximum performance accomplished is 1.16 MPG_{AS}, which corresponds to an average range/fuel consumption of 1566 MPG for the three UAVs. In the case of the maximum-performance design, the commonality is zero, that is, the UAV family is composed of (the maximum possible) 18 unique modules. The individual UAV performances given by the Pareto solutions with the maximum commonality and the maximum performance are compared in Table 3. In this table, max CI represents the Pareto solution with the maximum commonality (among three UAVs), and max MPG represents the Pareto solution with the best UAV-family performance based on average range/fuel consumption or MPG.

It is readily evident from Table 3 that a significant compromise in the individual UAV range/fuel consumption is necessary to accomplish greater commonality among the UAVs. The second UAV, with medium endurance and payload, experienced the maximum relative compromise in range/fuel consumption. On the other hand, the third UAV, with specified low endurance and high payload, accomplished the lowest range/fuel consumption in both cases (high CI and high-performance cases). This observation shows that, in general, high payloads are expected to result in low range/fuel consumption. Interestingly, although the endurance was specified to be 24, 16, and 4 h, respectively, for the three UAVs, all of them ended up with relatively high endurance values of 29–36 h for the optimized family with maximum commonality — attributed to large-sized wingspans (and longer tails for corresponding stability). It is also interesting to note that, even though the performances of the individual UAVs are equally weighted, higher commonality demanded a greater performance compromise from UAV 2 compared to that from UAV 3. Further exploration of how commonality and the importance of each product impacts the tolerance of each product variant toward performance compromise is an important topic of future research in reconfigurable PFD.

The observation discussed previously corroborates the hypothesis (presented in Sec. III.D) that maximizing commonality could drive the UAV variants toward high-endurance designs with larger wing-

Table 3 Comparing Pareto solutions with maximum commonality and maximum performance

	Endurance, h		Payload, lb		Range/fuel, MPG	
UAV	Maximum CI	Maximum MPG	Maximum CI	Maximum MPG	Maximum CI	Maximum MPG
1	36.0	24.0	7.0	7.0	553.5	1323.0
2	32.9	16.0	3.0	3.0	621.0	2470.5
3	29.4	4.0	20.0	20.0	445.5	904.5

Table 4 UAV platform plan for the best tradeoff family with maximum commonality

Module	UAV 1	UAV 2	UAV 3
Wing	A	B	C
Fuselage/pod	D	D	E
Vertical tails	F	G	H
Horizontal tail	I	I	I
Booms	J	J	K
Fuel tank	L	L	L

spans; this design trend, thus in spite of lower operational efficiency (lower MPG), is still able to satisfy the mission requirements. However, such larger wingspans could also lead to greater product fabrication costs. Hence, solely maximizing commonality might not yield a cost benefit on the whole. This issue is often ignored in the product-family literature, in which a bulk of the reported methods only pursues maximization of a commonality metric (along with performance maximization). It is to further explore the potential for practical economic benefit attributed to commonality, that we perform case 2, in which the commonality maximization is replaced by a cost-minimization objective.

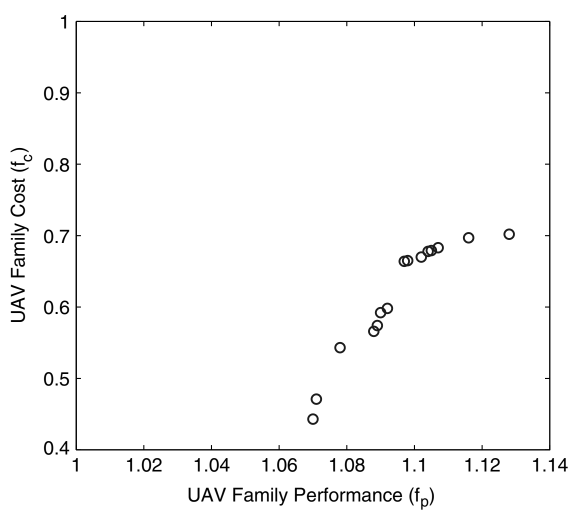


Fig. 5 Best tradeoff UAV families for case 2 (minimizing cost).

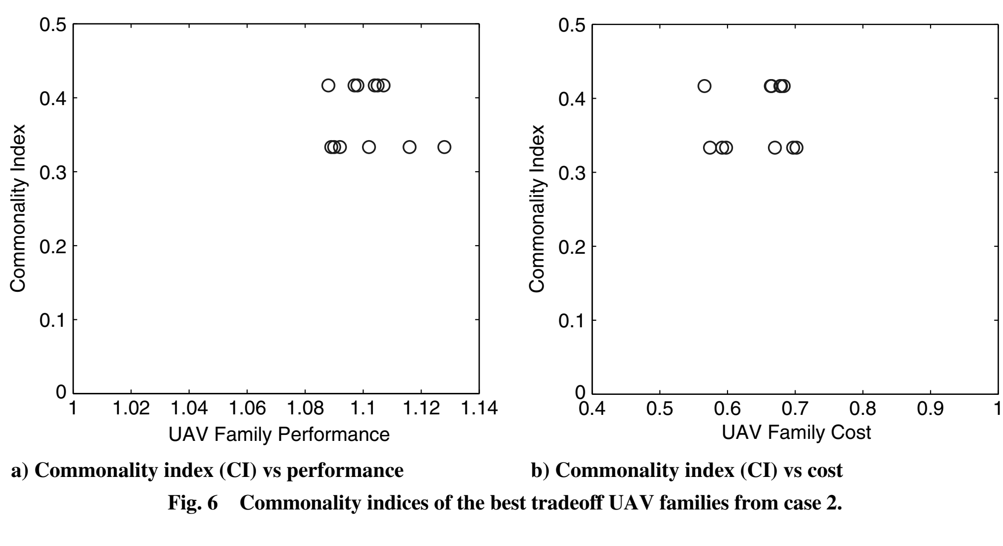
The platform configuration corresponding to the Pareto solution with maximum commonality is shown in Table 4. In this table, each uppercase letter represents a unique module. The fully shared modules are marked in italics, and the partially shared modules are marked in bold font.

As seen from Table 4, two modules (horizontal tail and fuel tank) are fully shared, and two other modules (fuselage and booms) are partially shared in the max-commonality Pareto solution. The wing and the vertical tails are each composed of four design variables, under the current formulation. Hence, the sharing of these two modules would require greater number of physical design variables to be shared among the three UAVs [Eq. (2)], compared to that required by the sharing of the other lower-dimensional modules. This mathematical attribute is likely to have partially promoted unique wing and vertical-tail designs among the three UAVs.

2. Case 2 (Minimizing Cost): Results and Discussion

The NSGA-II is used in case 2 with a population size of 200 candidate solutions. The optimization is allowed to run for a maximum of 200,000 function evaluations. The equality constraint h_{cc} is converted to an inequality constraint, as in the previous case. The Pareto solutions obtained, based on the weak-dominance principle, are shown in Fig. 5.

In Fig. 5, both objective functions appear in their normalized form, that is, the value of 1 for performance corresponds to the MPG of the Aerosonde UAV, and the value of 1 for cost corresponds to the net cost of the three UAVs individually optimized for performance. It is thus observed from Fig. 5 that the best tradeoff UAV families (Pareto solutions) have a lower cost than that of the three individually performance-optimized UAV variants. At the same time, the best tradeoff UAV families also provide better performance (fuel economy) than the Aerosonde UAV (subject to the assumptions made in this paper). The overall Pareto becomes nonconvex, thereby justifying the use of a powerful heuristic multiobjective optimization solver. Across the Pareto, the UAV-family performance changes from 1.07 to 1.13, a 6% increase; in contrast, the UAV-family cost changes from 0.70 to 0.44, a 26% decrease with respect to the reference cost.



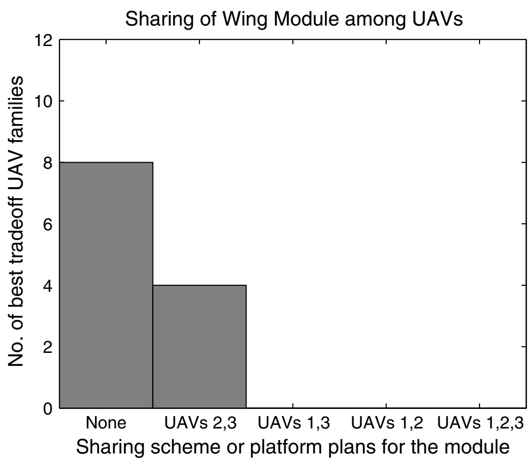
a) Commonality index (CI) vs performance **b) Commonality index (CI) vs cost**

Fig. 6 Commonality indices of the best tradeoff UAV families from case 2.

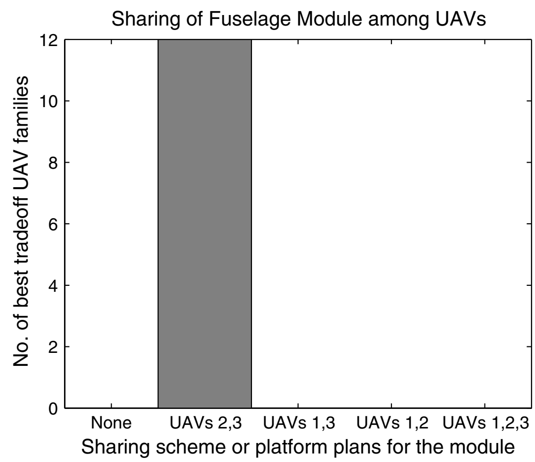
Chowdhury et al. [42] estimated the maximum aggregate performance of the individually optimized UAVs to be 1.16 times of that of the Aerosonde. Therefore, CP^3 offers promising cost savings for a relatively small compromise of the UAV performances.

It is important to note that the cost savings in this case are mainly attributed to 1) material reductions and 2) reconfigurability or reduction in the overall number of parts required to create the three different UAVs (lower than 3×6 parts). In practice, additional commonality-based cost savings are expected owing to 1) reduced manufacturing overhead, 2) reduced costs of transportation and aftermarket services, and 3) more streamlined design evolution (based on standard platforms). These Pareto results thus highlight the immense potential of applying product-platform-planning concepts to design efficient, affordable, and multimission-capable families of civilian UAVs.

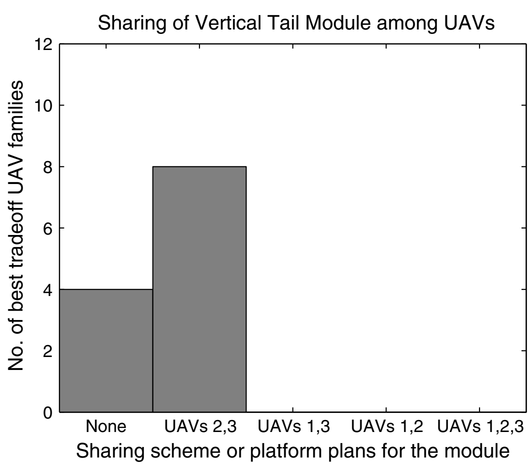
To better understand how commonality plays an important role in enabling cost savings through platform planning, the levels of commonality (CI) in the case 2 Pareto solutions are illustrated with respect to performance and cost (respectively, in Figs. 6a and 6b). It is observed from these figures that the best tradeoff UAV families fall into two levels of commonality: $CI = 0.33$ and 0.42. When the UAV-family performance and CI were maximized by Chowdhury et al.



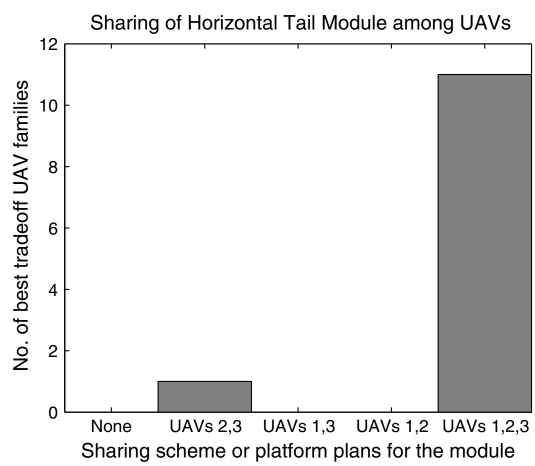
a) Platform plans for the wing module



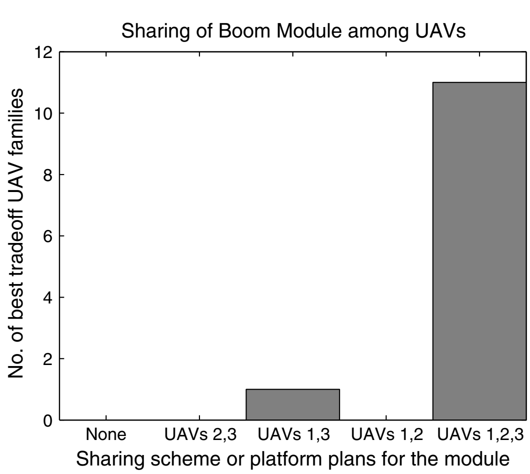
b) Platform plans for the fuselage module



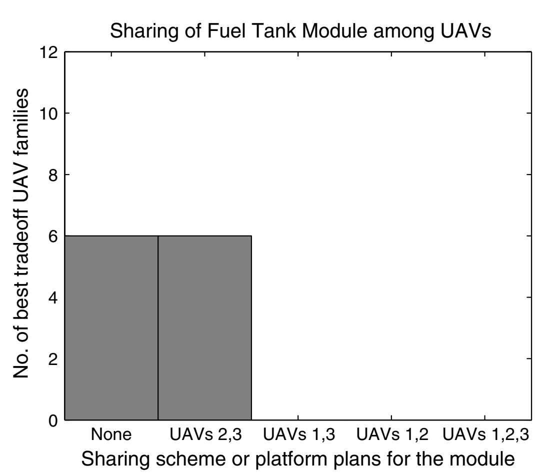
c) Platform plans for the vertical tail module



d) Platform plans for the horizontal tail module



e) Platform plans for the boom module



f) Platform plans for the fuel tank module

Fig. 7 Number of Pareto solutions under different possible platform plans for each module.

[42] (without any cost considerations), the CI varied from 0 to 0.5 across the Pareto solutions. The greater-than-zero values of CI for the Pareto solutions of case 2 show that the best tradeoff UAV families are pushed toward a higher degree of commonality — thereby establishing the impact of commonality (and hence reconfigurability) in enabling cost savings. In contrast, the occurrence of multiple Pareto solutions with different cost values at the same CI value shows that commonality is not the sole factor regulating potential cost reduction (in reconfigurable or product-family systems). Additionally, the highest value of commonality reached in case 1 (i.e., $\mathrm{CI} = 0.5$) is not reached in case 2, again indicating that solely optimizing commonality might not yield desirable cost savings on the whole. Figure 6b shows that the top-leftmost Pareto solution (with $\mathrm{CI} = 0.42$ and $\mathrm{cost} = 0.57 \times 3224$) could be practically the most desirable UAV-family design from a cost/commonality perspective.

To provide further insights into the degree of commonality or module sharing adopted by the 12 best tradeoff UAV families, we illustrate how many of the Pareto solutions fall into the different platform schemes with respect to each module. The histograms in Figs. 7a–7f provide these illustrations, with respect to each of the six UAV modules. The five different platform plans or module-sharing schemes possible in this case are the module is 1) not shared by any UAV, 2) shared by UAVs 2 and 3, 3) shared by UAVs 1 and 3, 4) shared by UAVs 1 and 2, and 5) shared by all three UAVs. The platform plans in Figs. 7a–7f appear in this same order. It is observed that the wing is the least likely module to be shared (Fig. 7a), and the horizontal tail and booms are the most likely modules to be shared (Figs. 7d and 7e). It is also observed that UAVs 2 and 3 are, in general, more likely to share the different modules compared to other possible pairs of UAVs.

This research provides a unique foundation for leveraging modular product-platform-planning methodologies to develop UAV families, comprising reconfigurable variants suited for different missions. In the current version, the overall product architecture is assumed to be known. Further advancement of the CP^3 framework is therefore necessary if the product architecture is to be planned within the optimization process; for example, when different sets of sensors and avionics could be incorporated into the UAVs, and the individual UAVs might not be composed of the same sets of modules. In the current form, the performance aspect of the aircraft-family design is mainly driven by aerodynamics (estimated from simplified analytical formulas); more comprehensive design would demand consideration of stability issues and structural performance. In that case, optimal platform planning would involve a significantly greater number of constraints; the increased complexity might necessitate the application of collaborative/distributed optimization methods [45], instead of a straightforward single heuristic optimization (as used in this paper).

IV. Conclusions

In this paper, modular product-family design (PFD) concepts were applied to design a family of unmanned aerial vehicles (UAVs), in which modules are allowed to be developed around common platforms. This design approach allows individual UAVs with unique mission capabilities to be configured/assembled from a set of available modules, immediately before a mission. However, this design effort necessitates certain attributes that are typically provided by state-of-the-art scale-based PFD methods (generally lacking in conventional module-based PFD methods), such as 1) allowing modules to be shared across a subset of the product variants in the family, and 2) the simultaneous identification and quantification of platform design variables; and at the same time, demands design attributes that are beyond the capabilities of most scale-based PFD methods (e.g., the provision for sharing entire multivariable modules).

Hence, in this paper, important advancements were made to a scale-based PFD method called CP^3. More specifically, the binary variables in the commonality matrix are now defined to represent the scheme of module sharing, as opposed to representing the design-variable sharing scheme as in the original CP^3. The commonality constraint is now formulated to consider the combination of these commonality variables and the entire design vector for each module (across all product variants). These modifications now allow a module to be shared across product variants (if platform derived), or to be substituted (if scaling) by a module of another configuration when one product variant is being reconfigured into another product variant — thereby allowing different product variants to be configured from one another by swapping out a few modules (that are not shared). Subsequently, in the process of reducing the platform-planning model from a high-dimensional binary problem into a more tractable integer problem, both module-inclusion variables and commonality variables are now considered, whereas only the latter was considered in the original CP^3 method. This formulation now not only allows the filtering of module combinations that are known a priori to be infeasible, but also (in principle) allows varying the product structure itself during optimization (which is helpful at a conceptual-design stage). The present formulation, however, yields an mixed-integer nonlinear programming (MINLP) problem that involves a multimodal commonality constraint, which, to an extent limits the optimization-algorithm choices (to heuristic algorithms). Further decomposition of the commonality constraint can be pursued in the future, to exploit powerful gradient-based MINLP solvers for (CP^3) optimization.

The modified CP^3 method is applied to design a family of three twin-boom UAVs with different endurance and payload-capacity specifications. The optimization of the UAV family is first performed to maximize the average range/fuel consumption, while maximizing the commonality among the three UAVs; to this end, a multiobjective variation of the mixed-discrete particle swarm optimization is implemented. The Pareto solutions obtained span from a maximum CI of 0.5 at a range/fuel consumption of 541 miles/gallon (MPG) to a maximum range/fuel consumption of 1566 MPG at no commonality among the three UAVs. In the Pareto-optimal family of UAVs with maximum commonality, UAV 1 could be reconfigured into UAV 2 by only swapping out the wing and vertical-tail-section modules — the new modular CP^3 method thus provides reconfigurability capabilities that significantly enhance user convenience and reduce cost to user/manufacturer, compared to the current commercial UAV platforms. It was found that, among the best tradeoff UAV families, the horizontal tail is the most likely to be shared, whereas the wing is the least likely to be shared. Further examination of the maximum-commonality designs showed that purely maximizing commonality was pushing the individual UAVs toward likely expensive designs (more material cost attributed to a larger wingspan that satisfies the performance constraints of each UAV variant). Hence, as a second case study, commonality maximization was replaced by cost minimization, in which the UAV costs were based on carbon-fiber-based fabrication and costs of off-the-shelf components. In this case, it was observed that the best tradeoff UAV families offered an attractive 26% reduction in cost for a 6% compromise in performance; this cost reduction was partially attributed to commonality, thereby further establishing the benefits of reconfigurable UAV families. In addition to providing manufacturing cost savings (as is expected from PFD concepts), the development of such modular UAVs with preflight reconfigurability provides greater affordability to the end user in a potentially emerging civilian market.

Although the present formulation in CP^3 does not necessarily require a priori modularization, the current optimization method allowed consideration of only a known fixed product structure; for example, each UAV is assumed to specifically comprise the wing, the tail sections, fuselage, fuel tank, and boom modules. Further flexibility in product-platform planning could be achieved by simultaneously optimizing the product structure, along with the platform plan and module configuration; this would in turn call for the exploration of more advanced methods, most likely a decomposed approach, to optimizing the modular product platform. In addition, the reconfigurability concepts presented here can further benefit through the consideration/exploration of premission assembly techniques and the formulation of the fabrication constraints (e.g., module interfaces) — both of which are important directions for further research in this area.

Acknowledgments

The information and illustrations provided by RenAir LLC regarding their unmanned-aerial-vehicle conceptual designs are gratefully acknowledged.

References

[1] Willcox, K., and Wakayama, S., "Simultaneous Optimization of a Multiple-Aircraft Family," *Journal of Aircraft*, Vol. 52, No. 6, 2003, pp. 616–622.
doi:10.2514/2.3156

[2] Allison, J., Roth, B., Kokkolaras, M., Kroo, I., and Papalambros, P. Y., "Aircraft Family Design Using Decomposition-Based Methods," *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA, Reston, VA, Sept. 2006, pp. 1–12.

[3] Cabral, L. V., and Paglione, P., "Conceptual Design of Families of Aircraft Using Multi Objective Design Optimization Theory and Genetic Algorithm Techniques," *6th World Congresses of Structural and Multidisciplinary Optimization*, World Congress of Structural and Multidisciplinary Optimization (International), May 2005.

[4] Cabral, L. V., and Paglione, P., "Methodology for the Design of Unmanned Aircraft Product Families," *28th International Congress of the Aeronautical Sciences*, International Congress of the Aeronautical Sciences, Bonn, Sept. 2012.

[5] Pate, D. J., Patterson, M. D., and German, B. J., "Optimizing Families of Reconfigurable Aircraft for Multiple Missions," *L'Aerophile*, Vol. 49, No. 6, 2012, pp. 1988–2000.
doi:10.2514/1.C031667

[6] Simpson, T. W., Chen, W., Allen, J. K., and Mistree, F., "Conceptual Design of a Family of Products Through the Use of the Robust Concept Exploration Method," *6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 1996-4161-CP, 1996, pp. 1535–1545.

[7] Simpson, T. W., "Product Platform Design and Customization: Status and Promise," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 18, No. 1, 2004, pp. 3–20.
doi:10.1017/S0890060404040028

[8] Uzumeri, M., and Sanderson, S. W., "A Framework for Model and Product Family Competition," *Research Policy*, Vol. 24, No. 6, July 1995, pp. 583–607.
doi:10.1016/S0048-7333(94)00788-8

[9] Sanderson, S. W., and Uzumeri, M., *Managing Product Families*, Irwin, Willowbrook, IL, 1997, p. 218.

[10] Sabbagh, K., *Twenty-First Century Jet: The Making and Marketing of Boeing 777*, Scribner, New York, 1996, p. 368.

[11] Rai, R., and Allada, V., "Modular Product Family Design: An Agent Based Optimization Technique," *International Journal of Production Research*, Vol. 41, No. 17, 2007, pp. 4075–4098.
doi:10.1080/0020754031000149248

[12] Fujita, K., and Yoshida, H., "Product Variety Optimization: Simultaneous Optimization of Module Combination and Module Attributes," *ASME 2001 International Design Engineering Technical Conferences (IDETC)*, American Soc. of Mechanical Engineers Paper DETC2001/DAC-21058, Fairfield, NJ, Sept. 2001.

[13] Fujita, K., and Yoshida, H., "Product Variety Optimization Simultaneously Designing Module Combination and Module Attributes," *Concurrent Engineering*, Vol. 12, No. 2, 2004, pp. 105–118.
doi:10.1177/1063293X04044758

[14] Stone, R. B., Wood, K. L., and Crawford, R. H., "A Heuristic Method to Identify Modules from a Functional Description of a Product," *Design Studies*, Vol. 21, No. 1, 2000, pp. 5–31.
doi:10.1016/S0142-694X(99)00003-4

[15] Dahmus, J. B., Gonzalez-Zugasti, J. P., and Otto, K. N., "Modular Product Architecture," *Design Studies*, Vol. 22, No. 5, 2001, pp. 409–424.
doi:10.1016/S0142-694X(01)00004-7

[16] Guo, F., and Gershenson, J. K., "Comparison of Modular Measurement Methods Based on Consistency Analysis and Sensitivity Analysis," *ASME 2003 International Design Engineering Technical Conferences (IDETC)*, American Soc. of Mechanical Engineers Paper DETC2003/DTM-48634, Fairfield, NJ, Sept. 2003.

[17] Jose, A., and Tollenaere, M., "Modular and Platform Methods for Product Family Design: Literature Analysis," *Journal of Intelligent Manufacturing*, Vol. 16, No. 3, 2005, pp. 371–390.
doi:10.1007/s10845-005-7030-7

[18] Kalligeros, K., de Weck, O., and de Neufville, R., "Platform Identification Using Design Structure Matrices," *Sixteenth Annual International Symposium of the International Council On Systems Engineering (INCOSE)*, INCOSE, Seattle, WA, July 2006, pp. 1–14.

[19] Sharon, A., Dori, D., and de Weck, O., "Model-Based Design Structure Matrix: Deriving a DSM from an Object-Process Model," *Second International Symposium on Engineering Systems*, CESUN and MIT ESD, Cambridge, MA, June 2009, p. 1–12.

[20] Yu, T. L., Yassine, A. A., and Goldberg, D. E., "An Information Theoretic Method for Developing Modular Architectures Using Genetic Algorithms," *Research in Engineering Design*, Vol. 18, No. 2, 2007, pp. 91–109.
doi:10.1007/s00163-007-0030-1

[21] Lewis, P. K., "*Multiobjective Optimization Method for Identifying Modular Product Platforms and Modules that Account for Changing Needs over Time*," M.S. Thesis, Brigham Young Univ., Provo, UT, Aug. 2010.

[22] Chowdhury, S., Messac, A., and Khire, R., "Comprehensive Product Platform Planning (CP^3) Framework," *Journal of Mechanical Design*, Vol. 133, No. 10, Oct. 2011, Paper 101004.
doi:10.1115/1.4004969

[23] Chowdhury, S., Messac, A., and Khire, R., "Investigating the Commonality Attributes for Scaling Product Families Using Comprehensive Product Platform Planning (CP3)," *Structural and Multidisciplinary Optimization*, Vol. 48, No. 6, 2013, pp. 1089–1107.
doi:10.1007/s00158-013-0953-2

[24] Simpson, T. W., Siddique, Z., and Jiao, R. J., "Platform-Driven Development of Product Families," *Product Platform and Product Family Design: Methods and Applications*, 1st ed., Springer–Verlag, New York, 2006, pp. 27–47.

[25] Meyer, M. H., and Lehnerd, A. P., *The Power of Product Platforms: Building Value and Cost Leadership*, The Free Press, New York, 1997, p. 288.

[26] Ulrich, K. T., "The Role of Product Architecture in the Manufacturing Firm," *Research Policy*, Vol. 24, No. 3, 1995, pp. 419–440.
doi:10.1016/0048-7333(94)00775-3

[27] Messac, A., Chowdhury, S., and Khire, R., "One-Step Continuous Product Platform Planning: Methods and Applications," *Advances in Product Family and Product Platform Design*, 1st ed., Springer–Verlag, New York, 2014, pp. 295–321.

[28] Chowdhury, S., Tong, W., Messac, A., and Zhang, J., "A Mixed-Discrete Particle Swarm Optimization with Explicit Diversity-Preservation," *Structural and Multidisciplinary Optimization*, Vol. 47, No. 3, March 2013, pp. 367–388.
doi:10.1007/s00158-012-0851-z

[29] Khajavirad, A., and Michalek, J. J., "A Decomposed Gradient-Based Approach for Generalized Platform Selection and Variant Design in Product Family Optimization," *Journal of Mechanical Design*, Vol. 130, No. 7, July 2008, Paper 071101.
doi:10.1115/1.2918906

[30] Lang, M. W., "Modular Function Deployment: Using Module Drivers to Impart Strategies to a Product Architecture," *Advances in Product Family and Product Platform Design*, 1st ed., Springer–Verlag, New York, 2014, pp. 91–118.

[31] Hölttä-Otto, K., "Architectural Decomposition: The Role of Granularity and Decomposition Viewpoint," *Advances in Product Family and Product Platform Design*, 1st ed., Springer–Verlag, New York, 2014, pp. 221–243.

[32] Hölttä-Otto, K., "Defining Modules for Platforms: An Overview of the Architecting Process," *Advances in Product Family and Product Platform Design*, 1st ed., Springer–Verlag, New York, 2014, pp. 323–341.

[33] Khajavirad, A., Michalek, J. J., and Simpson, T. W., "An Efficient Decomposed Multiobjective Genetic Algorithm for Solving the Joint Product Platform Selection and Product Family Design Problem with Generalized Commonality," *Structural and Multidisciplinary Optimization*, Vol. 39, No. 2, 2009, pp. 187–201.
doi:10.1007/s00158-008-0321-9

[34] Martin, M., and Ishii, K., "Design for Variety: A Methodology for Understanding the Costs of Product Proliferation," *ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, American Soc. of Mechanical Engineers Paper 96-DETC/DTM-1610, Fairfield, NJ, Aug. 1996.

[35] Chowdhury, S., Messac, A., and Khire, R., "Comprehensive Product Platform Planning (CP3) Using Mixed Discrete Particle Swarm Optimization and a New Commonality Index," *ASME 2012 International Design Engineering Technical Conferences (IDETC)*, American Soc. of Mechanical Engineers Paper DETC2012-70954, Fairfield, NJ, Aug. 2012.

[36] Cox, T. H., Nagy, C. J., Skoog, M. A., Somers, I. A., and Warner, R., "A Report Overview of the Civil UAV Capability Assessment," NASA TR, 2005.

[37] Breguet, L., *L'Aerophile*, Vol. 29, No. 271, 1921.

[38] Corke, T., *Design of Aircraft*, 1st ed., Prentice–Hall, Upper Saddle River, NJ, 2002, p. 390.

[39] Gundlach, J., *Designing Unmanned Aircraft Systems: A Comprehensive Approach*, 1st ed., AIAA, Reston, VA, 2011, p. 805.

[40] Anon, Barnard Microsystems Limited, "First Atlantic Crossing by an Unmanned Aircraft," www.barnardmicrosystems.com, 2012 [retrieved 17 Nov. 2015].

[41] Anon., DOD, "Unmanned Aircraft Systems Roadmap: 2005–2030," Office of the Secretary of Defense Tech. Rept., Aug. 2005.

[42] Chowdhury, S., Maldonado, V., Tong, W., and Messac, A., "Comprehensive Product Platform Planning (CP^3) for a Modular Family of Unmanned Aerial Vehicles," *ASME International Design Engineering Technical Conferences*, American Soc. of Mechanical Engineers Paper DETC2013-13181, Fairfield, NJ, Aug. 2013.

[43] Simpson, T. W., Maier, J. R. A., and Mistree, F., "Product Platform Design: Method and Application," *Research in Engineering Design*, Vol. 13, No. 1, 2001, pp. 2–22. doi:10.1007/s001630100002

[44] Timothy,, W. S., Brayan, S., and D'souza, T. W. S. B. S. D., "Assessing Variable Levels of Platform Commonality Within a Product Family Using a Multiobjective Genetic Algorithm," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Reston, VA, Sept. 2002.

[45] Kroo, I., "Distributed Multidisciplinary Design and Collaborative Optimization," *VKI Lecture Series on Optimization Methods for Multicriteria/Multidisciplinary Design*, Stanford Univ., Palo Alto, CA, Nov. 2004, pp. 1–22.

[46] Chowdhury, S., Maldonado, V., and Patel, R., "Unmanned Aerial System-Based Wind Resource Assessment Technology," 2012, www.renairtech.com [retrieved July 2013].