AIAA JOURNAL
Vol. 46, No. 5, May 2008

Integrated System-of-Systems Synthesis

Jaroslaw Sobieszczanski-Sobieski*
NASA Langley Research Center, Hampton, Virginia 236811

DOI: 10.2514/1.27953

This paper defines a system of systems as an assemblage of components organized in three levels, each level featuring both analysis and optimization. The choice of only three levels is deliberate, based on examination of the nature of a system of systems, a sample of references, and the range of potential applications. The solution methodology introduced in the paper derives as an extension of an existing bilevel integrated system synthesis method for which an application and implementation experience, as well as rigorous proof of correctness, are available. The new method, called trilevel integrated system synthesis, addresses the engineering practice requirements discussed in the paper. It builds on the formal representation of the system-of-systems coupling data known as the data dependency matrix, on the use of surrogate models, and on a recursive similarity of the data coupling among the systems in a system of systems and among the components within the system. The paper defines three variants of trilevel integrated system synthesis and identifies one of the three for implementation by the criteria of computational cost and engineering practicality. The closing discussion points out that because of extraordinary cost development and doubtful utility of simplified test cases, a system-of-systems optimization method ought to be developed in conjunction with a real, full-scale, application.

Nomenclature

B = number of black boxes in a system
C = number of cycles to convergence
c = equality constraint
E = number of arithmetical operations
E_s = total computational effort
F = objective function for a system or system of systems
f = black-box-level objective function
g = inequality behavior constraint
H = number of coordinates in space Q^{sos}
h = equality constraints
K = number of coordinates in space Q
L = lower bound
M = number of coordinates in the response surface space
$\mathcal{O}$ = order of magnitude
P = number of trial points for constructing a response surface, field variable in the condensation technique
p = parameter in condensation technique
Q = design space for a system
Q^{e} = extended space
Q^{s} = space formed by merging black-box spaces Q with Q^{sos}
Q^{sosBB} = subset of Q^{sos}
R = number of systems in a system of systems
v = variable in a basis function in condensation technique
w = weighting coefficient in weighted sum objective function
X = design variable vector
Xloc = local design variable vector
Xsh = shared design variable vector
Y = coupling variable vector
Y_{as} = Y output from BB_a to BB_s
Y_{sa} = Y output from BB_s to BB_a
Y^* = Y issued by the optimizer
$Y^\wedge$ = Y received by the optimizer
Z^{sos} = vector of space of variables for a system of systems in the all-in-one method

Subscripts

a = aerodynamics
s = structures, Y variable selected as the objective in generic discussion
0 = optimized quantity

Superscripts

a = approximate value of a variable
sos = quantities at the system-of-systems level

I. Introduction

ADVANCES in economy, technology, and science increasingly result in assemblages of many interacting entities of a diverse nature, governed by a variety of natural and man-made phenomena. These assemblages are usually referred to as systems, and at any instant there are systems that exist and are evolving, those that are in the process of being designed and implemented, and those that appear as future prospects. The need to deal with systems motivated research that has recently been showing increasing vigor and contributed to a growing body of literature examining various aspects of the system subject. A confluence of that body of literature with the literature that has been developing for years in the field of the multidisciplinary design optimization (MDO), in which the notion of a system became a well-established part of the lexicon, resulted in a mature methodology for at least two-level system optimization, as documented in the survey papers [1,2]. Growth of system complexity naturally leads to the notion of an assemblage of systems as a system of systems (SOS). Examples abound, ranging from a person's residence, through aircraft, to national transportation.

A recent paper [3] proposes a very general idea of a SOS that goes much beyond the now-conventional MDO interpretation of a system as a two-level entity and extends it to as many as five levels. A generic process for SOS development is examined in [4]. That hypothetical process could potentially be a systematic, rational way to build the SOS for a variety of applications, some examples of which include national aeronautics enterprise [5], urban transportation controlled growth [6], introduction of small aircraft to augment airlines in the U.S. national transportation system [7], and defense systems [8]. The

Presented as Paper 7064 at the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, VA, 6–8 September 2006; received 21 September 2006; revision received 19 July 2007; accepted for publication 19 July 2007. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/08 $10.00 in correspondence with the CCC.

*Distinguished Research Associate, Research and Technology Directorate, Mail Stop 188E; Jaroslaw.Sobieski-1@nasa.gov. Fellow AIAA.

concept of the SOS is so general that it could also be a candidate for a generic business methodology, as discussed in [9].

NASA has exhibited intense, explicit and implicit, interest in the SOS approach to planning its exploration mission, as illustrated by [10–13]. The SOS in that case would comprise a hierarchy extending from the interplanetary expedition architecture on top (e.g., the voyage to Mars), through the planetary habitats and crew transportation vehicle(s), to the basic elements of, say, the rocket engines at the bottom.

The preceding application examples and the NASA-sponsored studies of various aspects and options of the exploration mission architectures [10–13] further expand the SOS subject's diversity and complexity. The latter reflects the need to accommodate systems that, in addition to becoming components in a SOS, may have functionality of their own outside of that SOS. Furthermore, separate, and potentially incompatible or even conflicting, design objectives may exist at every level of a decomposed SOS, hence the need for a multi-objective optimization.

Because of the SOS subject's diversity and complexity, it is not surprising that no universally accepted definition of the SOS optimization problem has as yet evolved. For the same reasons, so far, no comprehensive solution has been developed to be effective for all conceivable types of SOS and to account for all possible quantitative and qualitative SOS characteristics. Recognizing that the feasibility of creating such a comprehensive solution is inherently at odds with the SOS definition generality, this paper takes the pragmatic approach of defining a SOS as an assemblage of components that may be organized in three levels, with nonhierarchal couplings operative within each level. This definition may be admittedly narrow, but it has an advantage of posing an optimization problem (analysis and synthesis) solvable by a specific algorithm presented herein. Thus, it is a definition inspired by existence of a particular solution method intended to address the salient, quantitative functional characteristics of a generic SOS.

The proposed SOS optimization method builds on a proven algorithm for optimization of two-level systems [the bilevel integrated system synthesis (BLISS)] for which the early theoretical development found in [14,15] culminated in [16]. It has recently become available as a commercial software [17,18].

The method description begins with a discussion of the intrinsic characteristics of the SOS pertaining to the underlying algorithm. The discussion includes practical considerations related to the real-world engineering that any method must address to be accepted. A proposal for a flexible and adaptable definition for the SOS is presented next, followed by an introduction of the method for trilevel integrated system synthesis (TLISS), developed as an extension of the BLISS algorithm, summarized from [16] in the Appendix to make the paper self-contained. The method is introduced in three variants. All three variants produce solutions equivalent to the SOS optimization without decomposition. However, they differ in regard to the computational effort gauged by the number of optimizations at the lowest level, opportunities for concurrent computing, and the real-world engineering practicality. By these criteria, two variants are disqualified and one is recommended for further development and implementation. The computational-effort estimates in the preceding evaluation of the three variants of the method also imply that testing of the proposed method by a test case, abridged and simplified enough to fit space limitations, could not demonstrate the method performance reliably. Hence, no such test is included in the paper. Instead, the closing discussion of the recommended further development and implementation of the method suggests meeting the challenges of the extraordinary cost and complexity, inherent in any SOS optimization method development, by pursuing such development in conjunction with a real, full-scale, SOS design.

II. SOS Characteristics and Methodology Development Considerations

To elaborate on the SOS definition proposed in the Introduction, consider that lexicographically, the term *system* implies at least two levels (i.e., components and their assemblage); hence, it follows that the SOS must include at least three levels. In fact, as mentioned before, it has been proposed [3] to extend the SOS to as many as five levels.

Although there are no firm logical limits to the number of levels, one needs to be cautious when increasing that number, lest it may lead to abstraction more complex than the very object being abstracted. It therefore appears prudent to limit the number of levels in a definition of the SOS to three, at least for the purposes of introducing an applicable method. The restriction to three levels need not result in a loss of generality, provided that the definition interpretation allows the concept of the SOS to be flexible, adapted to the complexity of the application, and bounded by the extremes illustrated by the following examples:

1) A liquid-fuel rocket engine is a SOS composed of the structure, fuel system, and control system, each system comprising components such as the fuel tank and the nozzle. Paralleling the preceding hierarchy of physical parts is a hierarchy of the corresponding physical models aligned with the appropriate engineering disciplines.

2) Planetary exploration mission architecture is conceptualized as a SOS, which entails the systems of infrastructure on the Earth, interplanetary transportation, and the planet surface habitat, the systems comprising components such as the launch vehicle, transfer vehicle, and life support system.

Consider now a vertical sliding scale to illustrate the nature of the information governing the SOS design changing from predominantly quantitative at the bottom to mostly qualitative at the top. A SOS illustrated by extreme 1 falls at the low end of the scale because it entails the core computational engineering at its component level, in which mathematical models and their solvers, such as finite element analysis or computational fluid dynamics, operate. However, even at that level, engineering decisions involve much more than computation and so means must provided to admit information from any source such as judgment, experiment, historical data, etc. As one moves the SOS up the scale toward extreme 2, the nature of information involved changes from hard to soft. The judgment, economics, people factors, psychology, regulatory issues, and law become dominant and, more often, the design decisions require choices among discretely different alternatives. Consequently, utility of a quantitative methodology varies gradually from high at the bottom of the scale to merely advisory at the top.

In a free economy, a SOS at the upper reach of the scale is hardly ever designed; for example, one would not expect the U.S. national transportation system ever to be redesigned and implemented from scratch. Hence, the verb *design* does not fully apply. Instead, the best that can be expected is a SOS methodology used to guide the quantitative aspects of the SOS evolution to the largest extent practical. Such evolution must accommodate the existing systems and components; hence, optimization opportunities arise only at islands on which initial conditions do not unduly constrain the design freedom. Thus, the SOS methodology must enable analysis only applied to the existing parts, alongside optimization used wherever possible. Typically, fitting a new system into an existing SOS is an important design requirement; for example, a new transport aircraft must fit the present air traffic control and airport infrastructure to be a viable product.

Recognition of the role of the formal physics-based methodology being limited to the foot of the SOS scale does not degrade importance of the underlying physics in shaping the entire SOS. The history of technology provides an abundance of examples in which new physics and technology introduced at the component level (i.e., the bottom of the scale) have profoundly changed the SOS from its foundation to the top. Replacement of the beast power with mechanical power, starting with Watt's steam engine, makes the point.

Separate from the preceding quality-vs-quantity aspect of the SOS is the issue of decomposition as an approach to construction of a SOS optimization method. In principle, an abstraction of the SOS at hand decomposed into N levels may then be placed on a sliding scale anywhere between the extremes illustrated by examples 1 and 2.

Pragmatically speaking, the decision whether to use decomposition into N levels, three levels, two levels, or none at all depends on the complexity of the mathematical models involved and the number of people working with these models. Obviously, any decomposition exacts a price of an overhead worth paying only if the benefits are clear. The benefits to justify use of decomposition are, typically, manageability in a large project that may be intractable when treated monolithically, compression of the overall elapsed time by executing subtasks concurrently over a broad and geographically distributed work front, and use of massively concurrent computing.

If this discussion of the SOS nature leaves the impression that the SOS definition is somewhat fluid, that is simply a consequence of the multifaceted and ever-diverse nature of a SOS. That is an important reason why development of SOS methodology is a challenging task. In addition to the preceding aspects of a SOS, that task must also address the realities of the real-world engineering, such as

1) Engineers exercise judgment, individually and collectively, using all available information.

2) Engineers form groups aligned with project components: disciplines, parts, and processes.

3) Groups have authority to decide design issues within their domains; they resist being restricted to analysis only.

4) Groups own their methods and tools, control their work schedules within team deadlines, and may be geographically dispersed.

5) Concurrent operations compress the project elapsed time.

6) Independent, inconsistent, and even conflicting design objectives may exist at all levels and must be reconciled by group collaboration.

7) Existing components and systems, and interfaces to the external world, may have to be accommodated in the SOS design.

Any new method must accommodate all of the preceding aspects to gain user acceptance.

Consistent with the preceding discussion, a method for SOS synthesis introduced in the remainder of this paper aims at a SOS that is decomposable into three levels and placed near the bottom of the scale, where well-defined mathematical models exist. Emphatically, however, the method admits information external to these models and reflects the engineering-world realities listed in the foregoing.

III. Data Dependency in SOS

Formalization of the data dependency and transfers in a SOS is a prerequisite to the introduction of the SOS synthesis method in the next section. Figures 1–3 illustrate the underlying concept of a N-square diagram (data dependency matrix) [19].

To begin at a system level, consider Fig. 1 as a generic example. The rectangular boxes strung along the diagonal (order immaterial) represent black boxes (BBs) meant to be recipients and sources of information that admit input data through their horizontal sides, vertically from above or below, and output the data through their vertical sides, horizontally left or right. The term *black box* is meant here in its cybernetic sense (i.e., an entity that generates an output as a function of input); the inner workings responsible for that generation must be, of course, well-defined and functional, but are treated as given and need not be considered in examination of the data flow among the BBs.

The data transmitted between the BBs designated by Y are the system coupling data. A black dot at a row–column intersection

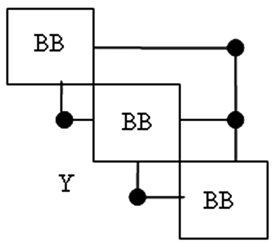

Fig. 1 Generic system.

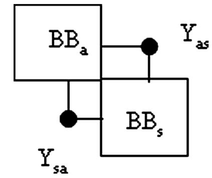

Fig. 2 Wing system.

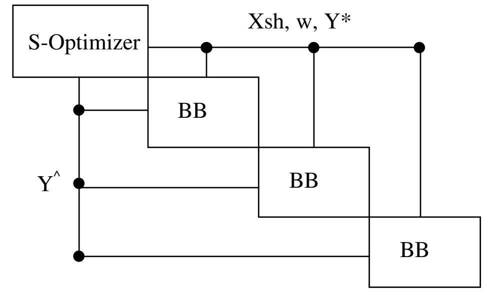

Fig. 3 Generic system of systems.

indicates that the output-to-input data transmission occurs (a data junction) between the corresponding two BBs. Absence of a black dot means that no such transmission takes place.

The Y data transmitted may be of any type and format (scalar, vector, matrix, table, etc.) and are defined for every black dot in the N-square diagram. The definition of a BB as an input-to-output information converter is understood in the broadest sense possible, ranging from educated guess, through handbook (or experiment), to a computer code that implements a mathematical model.

BBs that appear on the N-square diagram diagonal represent physical parts or disciplines exchanging data, as in Fig. 1. The content of the BBs and the data-exchange pattern shown in the diagram completely define the system physics.

For example, one may regard the system in Fig. 2 to be an aircraft wing for which the design involves the disciplines of aerodynamics BB_a and structures BB_s. Then the coupling data Y_{as} and Y_{sa} correspond to the aerodynamic loads and structural deformations. The inter-BB coupling requires consistency such that, in the preceding example, the deformation received in BB_a from BB_s must be generated in the latter from the aerodynamic loads reflecting the preceding deformation impact on BB_a.

Introduction of a system of systems is now straightforward because each BB in a system may be a system within itself (i.e., may be defined by a diagram such as Fig. 1). If the BBs in a system possess such internal structure, they may be interpreted as systems and their assemblage becomes a *system of systems*, such as illustrated in Fig. 3. In other words, the BBs are to a system what the systems are to a system of systems: a recursive relationship.

In the method introduced in the next section, optimization enters the picture at the BB level, the system level, and the SOS level. At the BB level, the optimization is internal to a BB; hence, it does not affect the data-exchange pattern in Figs. 1 and 3. It remains invisible in these figures. However, at the system level, the data-exchange pattern takes on a new form because optimization appears as a new BB, labeled S-optimizer (Fig. 4). The S-optimizer acts as the sole source of the input to the BBs and the only recipient of the output from these BBs according the concept known as simultaneous analysis and design (SAND) [1]. As Fig. 4 indicates, the optimizer-generated data include, in addition to Y, the design variables Xsh and the weighting coefficients w, which will be given due attention later (discussion of method 1 in Sec. IV.B.1 and definition of the variables in step 4 of problem 8). At this point, the discussion focuses on the coupling variables Y.

Comparison of Figs. 1 and 4 shows that the SAND concept modifies the original N-square diagram for a system. The inputs marked by Y^* designate the data that originally arrived from another BB and are now supplied by the S-optimizer. The outputs labeled by $Y^\wedge$ indicate the outputs that originally were routed directly to another

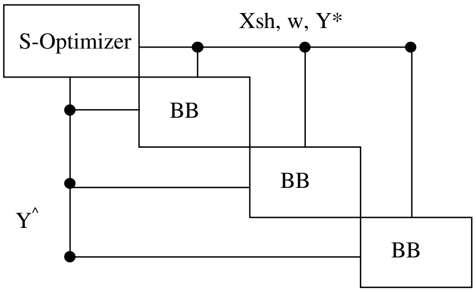

Fig. 4 System with optimizer.

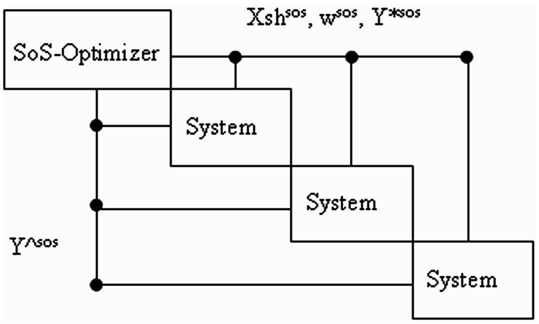

Fig. 5 System of systems with optimizer.

BB and are now sent to the S-optimizer. The S-optimizer recognizes $Y^\wedge$ as the given constants but treats Y^* as design variables appended to the usual design variables Xsh. It simultaneously manipulates Xsh, w, and Y^* to minimize the system objective or to find a Pareto minimum if it is a multi-objective case, and to enforce equality constraints $Y^* = Y^\wedge$. The equality constraints restore the direct output-to-input couplings between the BBs temporarily severed by breaking each Y into Y^* and $Y^\wedge$.

Returning to the wing example, if Fig. 2 were redrawn in the pattern of Fig. 4, it would show the optimizer sending Y^*_{as} to BB structures and Y^*_{sa} to BB aerodynamics and receiving $Y^\wedge_{sa}$ from BB structures and $Y^\wedge_{as}$ from BB aerodynamics. Of course, the couplings require that $Y^*_{as} = Y^\wedge_{as}$ and $Y^*_{sa} = Y^\wedge_{sa}$.

Satisfaction of the preceding equalities, together with the solution of equations internal to each BB, amounts to the system analysis. Under the SAND approach, that analysis is concurrent with the system optimization.

Moving now to the SOS level, the recursive relationship pointed out in the foregoing enables construction of the N-square diagram incorporating the SOS optimizer by converting Figs. 4 and 5. Merely relabeling the BB to the system and superscripting the variables with sos accomplishes the conversion.

As an example, one may consider an aircraft and a missile it carries as a SOS of two systems. The objective of the SOS is maximizing the probability of destroying a target under some constraints. Should the aircraft and the missile be designed together, the issue is how to balance design features (e.g., agility, range, weight, cost, etc.) between the aircraft and the missile. In this context, one instance of the Y^{sos} data might be the aircraft speed, angle of attack, acceleration, and altitude as the launch initial conditions for the missile, and the missile weight, geometrical dimensions, and weight to be accommodated in the carrier aircraft design.

To conclude the data structure description, the key point is in placing all the Y and Y^{sos} data traffic at the system and SOS levels under the sole control of the corresponding optimizers. As the principal benefit, this enables simultaneous executions of the BBs within a system and of the systems within the SOS and provides a basis for the method introduced next.

IV. Proposed Methodology

This section describes the TLISS method for optimization of a system of systems. The method rests on the SAND concept and the data organization discussed in the preceding section. It is a generalization to three levels of the method known as BLISS. To make the paper self-contained, the Appendix provides a summary of BLISS from [16].

A. Fundamental All-in-One Formulation

Any candidate solution to an optimization problem large and complex enough to warrant a decomposition approach should be equivalent to the solution of the same problem stated in its original form without decomposition. For a system of systems, such a fundamental, all-in-one (AIO), optimization problem statement collects the design variables X and the behavior (state) variables in Z^{sos} as follows:

Problem 1:

1) Find $Z^{\mathrm{sos}} = \{X\mathrm{sh}^{\mathrm{sos}} | X\mathrm{sh} | X\mathrm{loc}, Y\}$.

2) Minimize $\{F(Z^{\mathrm{sos}})\}$.

3) Satisfy $h(Y) = 0$, $g(Y) \leq 0$, and $Z^{\mathrm{sos}}_{lb} \leq Z^{\mathrm{sos}} \leq Z^{\mathrm{sos}}_{ub}$

The equality constraints in step 3 may include the governing equations (i.e., the problem analysis). Hence, the preceding notation reflects SAND. It is adopted here merely for its compactness and without implication that the problem analysis cannot be solved, at least in principle, as a separate operation nested in the optimization loop. Expressing the objective as a vector $\{F(Z^{\mathrm{sos}})\}$ acknowledges the possibility of more than one objective, implying that a Pareto optimization may be necessary.

B. System-of-Systems Three-Level Optimization

Three-level system-of-systems optimization readily derives from the similarity of the data exchange with an optimizer present, internal to the SOS illustrated in Fig. 5, and the data exchange depicted in Fig. 4, internal to any of the systems in the SOS. This section shows that there are at least three alternative formulations possible for the preceding optimization, each having its merits and demerits.

1. *Method 1*

When the superscript sos distinguishes the data exchanged among the systems in the SOS from those internal to a system, it is obvious that the variables $\{X\mathrm{sh}^{\mathrm{sos}}\}$, $\{w^{\mathrm{sos}}\}$, $\{Y^{*\mathrm{sos}}\}$, and $\{Y^{\wedge\mathrm{sos}}\}$ at the SOS level are analogous to the variables $\{X\mathrm{sh}\}$, $\{w\}$, $\{Y^*\}$, and $\{Y^\wedge\}$ within any system comprised by SOS. The w variables are auxiliary variables introduced to control the BB or system designs, per the discussion of Problem 8 in the Appendix.

Consequently, the same BLISS algorithm described in the Appendix that optimizes a system composed of the black boxes applies recursively to a SOS comprising the systems. Reconstruction of BLISS at the SOS level requires the definition of space

$$Q^{\mathrm{sos}} = \{X\mathrm{sh}^{\mathrm{sos}} | Y^{*\mathrm{sos}} | w^{\mathrm{sos}}\} \tag{1a}$$

as the counterpart to $Q = \{X\mathrm{sh} | Y^* | w\}$. For the reasons to be apparent later, it is also necessary to define a subset Q^{sosBB} of Q^{sos} as

$$Q^{\mathrm{sosBB}} = \{X\mathrm{sh}^{\mathrm{sos}} | Y^{*\mathrm{sos}}\} \tag{1b}$$

that is, Q^{sosBB} is Q^{sos} with w^{sos} deleted.

Now considering one of the systems in the SOS, strict interpretation of the recursive similarity of the systems assembled into the SOS and the BBs assembled into a system compels subjecting the system to the same treatment that BLISS applies to a BB in a system. That implies repetition of the BLISS application to that system at a number of design-of-experiments (DOE) selected points in the space Q^{sos}, noted formally as follows:

Problem 2:

1) Given $Q^{\mathrm{sos}} = \{X\mathrm{sh}^{\mathrm{sos}} | Y^{*\mathrm{sos}} | w^{\mathrm{sos}}\}$

2) Execute BLISS as described in the Appendix for one system in the SOS. The system comprises BBs. To optimize a system within the SOS in the same manner that a BB is optimized within the system, the system objective function in step 3 of problem 9 must be replaced to resemble the BB objective as in step 3 of problem 8, hence

$$F(Q) = \sum w_i^{\text{sos}} Y_i^{\wedge\text{sos}} \quad (2)$$

The preceding procedure is executed at trial points established by a DOE technique in Q^{sos} to obtain $Y_0^{\wedge\text{sos}}$ and Q_0.

3) Fit a response surface (RS) to each element of $Y_0^{\wedge\text{sos}}$ to obtain a collection of RS that may be thought of as a sheaf of leaves, each leaf corresponding to one RS, hence the generic name sheaf of Rs (SRS). The SRS representing Y^{sos} is designated SRS^{sos} and is analogous to SRS representing a BB in the system optimization in BLISS [note that surrogate models (SM) other than RS may be used, as discussed in [16]].

4) Repeat steps 2 and 3 for all systems. Note that an opportunity to operate concurrently on all systems arises here.

5) Output one SRS^{sos} in Q^{sos} per system for all systems in the SOS.

Once the SRS^{sos} have been obtained, the SOS optimization proceeds analogous to the system optimization in problem 9.

Problem 3:

1) Given a set of SRS^{sos}, one SRS^{sos} for each system.

2) Find $\{Q^{\text{sos}}\} = \{X\text{sh}^{\text{sos}}|Y^{*\text{sos}}|w^{\text{sos}}\}$.

3) Minimize $F^{\text{sos}}(Q^{\text{sos}}) = Y_{0s}^{\wedge\text{sos}\,a}$.

4) Satisfy $c = Y^{*\text{sos}} - Y_0^{\wedge\text{sos}\,a}(Q^{\text{sos}}) = 0$ and $QL^{\text{sos}} \leq Q^{\text{sos}} \leq QU^{\text{sos}}$.

5) Output Q_0^{sos} and F_0^{sos}.

6) Iterate from step 4 of problem 2 until the consecutive iterates of the SM in step 3 of problem 2 converge to an acceptable tolerance.

The preceding iteration controls the approximation error inherent in the SM generated in step 3 of problem 2. It is analogous to the BLISS iteration discussed in detail in the Iterative Procedure section of the Appendix, in which it is called the major iteration of the cycle. Similar to BLISS, the equality constraints c in step 4 of problem 3 enforce the couplings among the systems in the SOS.

A formal proof in [16] shows that the BLISS solution is equivalent to the solution of a two-level system without decomposition (i.e., the AIO formulation, in the sense that, at least for convex problems, the optimization solution from BLISS satisfies the Kuhn–Tucker optimality conditions written for the AIO formulation, with the deviation of the residuals of the Kuhn–Tucker equations from zero being dependent only on the accuracy of the SMs employed). One advantage of method 1 lies in the preceding proof extending to that method because of the recursive similarity of the SOS comprising the systems and the system composed of the BBs. As in BLISS, the proof of equivalence is not tantamount to the proof of convergence for every conceivable nonconvexity of SOS.

However, in method 1, the system optimization nests in the loop that generates SRS^{sos}; that is, for each trial point in Q^{sos}, one must complete the BLISS optimization of each of the constituent systems. This results in a severe computational-effort penalty. As pointed out in [16], it is a reasonable approximation to measure that effort by the number of operations at the BB level, because in large-scale problems, that number tends to be much larger than that required by the system-level optimization, which merely retrieves data from already established SMs. Let E be that number, and let H designate the number of elements in the vector Q^{sos}. Typically, the least number of trial points P for which the system optimization by BLISS must be repeated to obtain a SM for M variables is on the order of M^2. For example, for a SM as a quadratic polynomial response surface (RS) in M variables,

$$P = 1 + 3/2M + M^2/2 \quad (3)$$

Assuming that the number of BBs in the system is B, that each BB requires the same E for its optimization, and that the number of variables in the system space $\{Q\}$ is K, the estimate of the total computational effort E_s to optimize the system by BLISS converging in C cycles is

$$E_s = CB\mathcal{O}(K^2)E \quad (4a)$$

This effort has to be expended at each of the DOE-trial points for H variables in Q^{sos} in a SOS comprising R systems. Referring to Eq. (3) again, the total SOS computational effort E_{sos} for one iteration of problem 3 is

$$E_{\text{sos}} = RE_s\mathcal{O}(H^2) = RCEB\mathcal{O}(K^2)\mathcal{O}(H^2) \quad (4b)$$

Even though in a commercial implementation (e.g., [17,18]) the use of concurrent computing may eliminate the term $\mathcal{O}(\text{K}^2)$ from the corresponding expression for the total elapsed computing time, the product $\mathcal{O}(K^2)\mathcal{O}(H^2)$ still implies an excessive computational effort and puts utility of method 1 in doubt as a practical tool for realistic large-scale applications. Nevertheless, method 1 remains of interest as a vehicle by which the proof of equivalence to AIO extends from bilevel to trilevel optimization. The preceding assessment of the method 1 utility motivates the search for alternative methods 2 and 3, introduced next.

2. *Method 2*

This method reduces the computational effort per SOS optimization iteration by decreasing the number of repetitions of the BB optimization. It exploits the SAND formalism in BLISS by entering subset Q^{sosBB} of Q^{sos} [Eq. (1b)], received from the SOS level, as direct input to the BB. The use of Q^{sosBB} instead of Q^{sos} reflects the fact that w^{sos} in Q^{sos} matters only in the SOS optimization but not at the BB level. Therefore, method 2 simply appends Q^{sosBB} to Q defined in problem 8 of that BB to form an extended space Q^e

$$Q^e = \{X\text{sh}|Y^*|w|Q^{\text{sosBB}}\} \quad (5)$$

of the length $H + K$.

Construction of a RS as a SM in Q^e requires repetition of the BB optimization at the DOE-selected points for which the number is $\mathcal{O}(B(H+K)^2)$. That number is significantly smaller than the product $B\mathcal{O}(K^2)\mathcal{O}(H^2)$ in method 1 because, typically, $B \ll (H+K)$.

Method 2 optimizes the SOS by blending the SOS optimization with the system optimizations in the space Q^s formed by merging the BB spaces Q with Q^{sos}:

$$\{Q^s\} = \{Q_1|Q_2|Q_3|\cdots|Q_{\text{i}}|\cdots|Q^{\text{sos}}\} \quad (6)$$

The following states the entire method 2: For each BB in each system in the SOS, solve problem 4 at DOE-selected points in Q^e.

Problem 4: This optimization is the same as problem 8 in BLISS, except that the space in which it is repeated at DOE points is Q^e, which includes Q and Q^{sosBB} appended to it.

1) Given $Q^e = \{X\text{sh}|Y^*|w|Q^{\text{sosBB}}\}$.

2) Find $U = \{X\text{loc}|Y^\wedge\}$.

3) Minimize $f(U) = \sum w_i Y_i^\wedge$.

4) Satisfy $g(U) \leq 0$, $h(U) = 0$, and $UL \leq U \leq UU$.

5) Output $U_0 = \{Y_0^\wedge, X\text{loc}_0\}$.

6) For each BB, generate a SM analogous to Eq. (A1)

$$Y_0^{\wedge a} = Y_0^{\wedge a}(\text{SRS}(Q^e)) \quad (7a)$$

and, optionally,

$$U_0^a = U_0^a(\text{SRS}(Q^e)) \quad (7b)$$

Next, perform the SOS optimization blended with the system optimizations.

Problem 5:

1) Given a set of BB SRS in space $Q^e = \{X\text{sh}|Y^*|w|Q^{\text{sosBB}}\}$ for each BB in each system

2) Find $Q^e = \{X\text{sh}|Y^*|w\}$ for all systems and $Q^{\text{sos}} = \{X\text{sh}^{\text{sos}}|Y^{*\text{sos}}|w^{\text{sos}}\}$, all merged into a space Q^s for the SOS as in Eq. (6).

3) Minimize $F(Q^s) = Y_{0s}^{\wedge\text{sos}a}$.

4) Satisfy $c^{\text{sos}} = Y^{*\text{sos}} - Y^{\wedge\text{sos}} = 0$ (i.e., couplings among systems), $c = Y^* - Y^\wedge = 0$ (i.e., couplings among BBs in each system), and $QL \leq Q^s \leq QU$.

5) Output Q_0^s, $F_0(Q^s) = Y_{0s}^{\wedge\text{sos}a}$.

6) Iterate from step 1 of problem 4 until the consecutive iterates of the SM in step 6 of problem 4 converge to an acceptable tolerance.

The method 2 merit lies in reducing the total computational effort of one iteration (one cycle) of problems 4 and 5, down to

$$E_{\text{sos}} = RCE\mathcal{O}(B(K + H)^2) \quad (8)$$

where the term $\mathcal{O}(B(K + H)^2)$ replaces the term $B\mathcal{O}(K^2)\mathcal{O}(H^2)$ in Eq. (4b) for method 1. The ratio

$$(K + H)^2/(K^2H^2) = (K^{-2} + H^{-2} + 2/(KH))$$

shows how radically method 2 reduces E_{sos} compared with method 1, especially when K and H increase.

However, as a demerit, method 2 obliterates the individuality of the system optimizations. In effect, it becomes a bilevel approach because of the blending of the SOS with the systems optimizations. As pointed out in the discussion of the human organization attributes that influence multilevel optimization intended for large engineering design projects, the autonomy of the groups associated with BBs and systems in the SOS design is important and may be decisive in the choice of the method. Therefore, method 2 will be excluded from further considerations, and motivation persists to formulate yet another method to combine reduction of the computational effort with the protection of the system design autonomy.

3. *Method 3*

This method restores the autonomy of the individual system optimization in the SOS by altering the SOS optimization in problem 5 in method 2 so as to restore the separate autonomous system optimizations and to retain the BB optimization exactly as in problem 4. The latter enables construction of SRS for the BBs in Q^e for each system. However, the next step of method 3 optimizes each system individually in its space Q at DOE points established in Q^{sos} to generate the system SRS in Q^{sos}. The ensuing SOS optimization uses that SRS.

This approach leaves the computational effort about the same as that in method 2, because the number of repeated optimizations for a BB is the same, but preserves the system optimization autonomy at the price of accepting additional error from compounded approximations. The approximations compound because the system SRS, already an approximation, used in the SOS optimization derives from the system optimization that was conducted using the SRS approximating the optimized BBs in that system. Consequently, an increase of the problem-dependent approximation error is to be expected. However, the proof of equivalence for the BLISS algorithm in [16] remains unaffected by the approximation compounding, even though that compounding may affect the rate of convergence of the method 3 algorithm.

The step-by-step method 3 algorithm is as follows:

Algorithm 1:

1) Perform individual BB optimizations in Q^e per problem 4 in each system in the SOS.

2) Generate $Y_0^{\wedge a} = Y_0^{\wedge}(\text{SRS}(Q^e))$ and, optionally, $U_0^a = U_0(\text{SRS}(Q^e))$ in Q^e, as in method 2 [Eq. (7)].

3) Use DOE to select points in Q^{sos}.

4) At each of the preceding Q^{sos} points, perform system optimization in $\{Q\}$ for each system s in BLISS per problem 9, with step 3 of problem 9 replaced by Eq. (2), using $Y_0^{\wedge a} = Y_0^{\wedge}(\text{SRS}(Q^e))$ per step 2 of this algorithm.

5) For each system, use the preceding optimization results to construct $\text{SRS}(Q^{\text{sos}})$, approximating $Y_0^{\wedge\text{sos}a}(\text{SRS}(Q^{\text{sos}}))$.

6) Use the preceding $\text{SRS}(Q^{\text{sos}})$ to optimize the SOS in Q^{sos}.

The SOS optimization is stated as follows:

Problem 6:

1) Given $(\text{SRS}(Q^{\text{sos}}))$ for all SOS

2) Find $Q^{\text{sos}} = \{X\text{sh}^{\text{sos}}|Y^{*\text{sos}}|w^{\text{sos}}\}$.

3) Minimize $F = \{Y_{0s}^{\wedge\text{sos}a}\}$.

4) Satisfy $Y^{*\text{sos}} - Y^{\wedge\text{sos}} = 0$.

5) Output F_0 and Q_0^{sos}.

6) Iterate from step 1 of Algorithm 1 until the consecutive iterates of the SM in step 5 of Algorithm 1 converge to an acceptable tolerance.

The { } in step 3 of Problem 6 indicates the possibility that the SOS may require a multi-objective optimization.

Individual optimization of each system enables the engineering group in charge of that system to include judgment and draw information from all available sources for optimal design. Hence, method 3 emerges as one for which the prospect for both computational efficiency and practical utility is the best among the three methods examined.

C. Option for Analysis Without Optimization

The design of systems, and the SOS even more so, is very likely to include BBs representing designs that exist and should remain unchanged. Because they interact with their counterparts, such BBs must be subject to analysis even though they are not being optimized. The requisite analysis-only option is straightforward because a BB analysis is represented by $h = 0$, hence omitting step 3 in Problem 8 and Xloc as follows:

Problem 7:

1 Given $Q = \{X\text{sh}|Y^*|w\}$

2) Find $U = \{Y^{\wedge}\}$.

3) Satisfy $(g(U) \leq 0,\ \text{monitor}), h(U) = 0, \text{and}\ UL \leq U \leq UU$.

4) Output $Y^{\wedge}$ allows the particular BB to remain unchanged in its design (Xloc given and constant), although its behavior $Y^{\wedge}$ may change.

Although Problem 7 omits minimization, it retains monitoring the behavior constraints $g \leq 0$ to alert to a possibility that incorporation of an existing component in a system may subject that component to demands exceeding its capacity.

D. Poor Prospects for Extension Beyond Three Levels

As mentioned in the discussion of references, there is an interest in formulating the SOS solutions in more than the three-level limit adopted herein; therefore, the potential extensibility of method 1 and its derivative method 3 to more levels of decomposition needs to be addressed (method 2 has already been dropped from further consideration in the foregoing). Unfortunately, prospects for such extensibility appear to be poor for the following reasons.

Method 1 requires the computational effort E_{sos} estimated by Eq. (4b) in which every new level of decomposition adds another term of the $\mathcal{O}(H^2)$ type to the product. This reflects the nature of the method as a set of nested loops and puts the method utility in question even for SOS applications in three levels. Turning to method 3, the method is extensible beyond three levels by adding the system optimization at each new level, analogous to step 6 of Algorithm 1, based on the SM developed for the levels already existing. However, this stacking up of the SM compounds the approximation errors in a problem-dependent manner that may eventually exceed the error acceptability limits.

Finally, there is the issue of the human organization and management of the data handling and collaboration implicit in the methods presented. Obviously, complexity of these operations grows with the number of levels, as does the organizational inertia that hinders execution of a formal iterative process spanning several levels of management.

Although the preceding assessment does not firmly rule out formal SOS methods involving more than three levels of decomposition, it suggests that their practical utility and application success are doubtful.

V. Conclusions

The paper introduces a methodology for analysis and optimization of a system of systems (SOS). It opens with a discussion of selected references, the nature of the SOS and its components as they occur in the real world, and the requirements of engineering practice to ensure relevancy of the method to be proposed. The SOS is understood to be an assemblage of interacting components, the components themselves exhibiting the internal structure of interacting elements. This "box-in-the-box," or Russian matrioshka (doll-in-the-doll), composition of simpler constituents nested into ever-more-complex entities translates mathematically into a recursive relationship between the SOS levels and serves as a basis for the organization of the proposed solution methodology. It is shown that a particular two-level multidisciplinary design optimization (MDO) method documented in references as bilevel integrated system synthesis (BLISS), now available as a commercial tool, may be extended by the preceding recursion to solve a three-level SOS problem. Limiting the number of levels in the SOS to three is deliberate, per the reasons discussed. A corresponding formal definition of the SOS as a three-level entity is introduced. To accommodate a range of practical applications, the definition is flexible with respect to the complexity and generality of the SOS components. However, one should expect that the nature of the definition, which is predominantly quantitative and physics-based for a SOS at the engineering level (e.g., an aircraft), will gradually transform into a more qualitative description when the SOS scope broadens to an entity as general as a national transportation system. At that level, the underlying physics is still important but reduces to merely one of many facets of the problem. In this context, the methodology examined herein applies primarily to the SOS defined at the engineering level.

Three variants of trilevel integrated system synthesis (TLISS) are presented, all based on the techniques of design of experiments and surrogate modeling (via response surfaces). By a cited formal proof [16], all three produce solutions that are equivalent to the SOS optimization without decomposition. However, the engineering practice criteria and estimates of the computational effort disqualify two of the variants, leaving only one, called method 3, to be recommended for development and implementation.

One may expect that development, implementation, and qualification of any method for use in problems as large as those for which TLISS is intended will be a daunting task. Even though a full-scale test of correctness is not necessary, due to the existence of the aforementioned formal proof of equivalence, still the proposed method's overall performance must be experienced in at least near-full-scale application for the method to become a qualified tool. Such testing cost is likely to require resources far beyond those typically available for researchers. On the other hand, testing on a problem simplified enough to avoid excessive cost runs a risk of irrelevance. Thus, it appears that the subject development is in the same category as some of the space technology undertakings that by their nature defy simulation on Earth and require that a prototype must be the first operational artifact. However, the positive experience with two-level synthesis documented for BLISS (e.g., [15,20,21]) may encourage development of TLISS as a task inherent in an actual SOS design and development, even though opportunities to do this may be infrequent. The risk and cost of developing a SOS without a comprehensive formal method of the TLISS type may be a motive for undertaking development of a new SOS methodology in such an unconventional way.

Appendix: Summary of Bilevel Integrated System Synthesis

BLISS is a method for analysis and optimization of a system composed of interacting subsystems (black boxes). Optimization and analysis are performed at two levels: internally in each subsystem and at the system level. A full description of the BLISS algorithm appears in [16].

I. Subsystem Optimization

Consider one of the subsystems in a system to be a BB that receives the system-level variables $\{X\mathrm{sh}|Y^*|w\}$ (Fig. 4) from the system optimizer and internally manipulates the subsystem level variables $\{X\mathrm{loc}|Y^\wedge\}$. The following optimization problem is solved for each BB:

Problem 8:

1) Given $Q = \{X\mathrm{sh}|Y^*|w\}$
2) Find $U = \{X\mathrm{loc}|Y^\wedge\}$.
3) Minimize $f(U) = \sum w_i Y_i^\wedge$.
4) Satisfy $g(U) \le 0$, $h(U) = 0$, and $UL \le U \le UU$.
5) Output $Y_0^\wedge$ and $X\mathrm{loc}_0$.

The last term in step 4 of Problem 8 includes the BB governing equations $h(U) = 0$ to assure that the analysis of the physics problem simulated in the BB is solved in the optimization process. As mentioned in the body of the paper, this compact notation known as SAND should not be construed to mean that the analysis solution could not be treated separately as an operation embedded in the optimization loop (which is a prevailing practice). The Y variables are the coupling variables. To enable concurrent BB operations, each BB receives its Y^* variables from the system optimizer and sends its $Y^\wedge$ output to that optimizer, rather than exchanging its Y^* and $Y^\wedge$ directly with the other BBs.

BLISS uses the coefficients w in step 3 of Problem 8, one coefficient per element of the output vector $Y^\wedge$, as parameters by which the system controls the BB optimization outcome, in the sense that every setting of the vector w engenders a feasible design having a particular $Y^\wedge$ as a set of attributes unique to that setting. Using a wing as an example, and thinking of the $Y^\wedge$ elements such as the deformation and weight as the attributes, a variation of w produces a family of the wing designs. This family ranges from the wings that are relatively light but very flexible to those that are heavier but stiffer. Having a choice in the range, the system then uses weighting coefficients w as the system-level design variables to select a BB design that best serves the system objective. As shown in [22], such control of $Y^\wedge$ by means of w is effective and demonstrates that a change of w_i generates a corresponding change of the associated element of $Y_i^\wedge$.

The Problem 8 optimization executes at a number of points dispersed in the Q space associated with the BB. The formal DOE techniques may aid in forming the dispersal pattern to achieve a reasonable coverage of the domain.

In BLISS, the specialists in charge of a particular BB decide how to solve Problem 8 and what kind of information to use in that solution. In general, that information may use mathematical model, experimentation, historical data, judgment, or even a guess. Thus, a BB in BLISS has a broad meaning of an input-to-output information converter, of which a computer code is merely only one of many possible instantiations.

II. Optimized Subsystem Represented by an Approximate (Surrogate) Model

A SM separate for each BB captures the data from the multiple repetitions of the BB optimization. The response surfaces, kriging, or neural nets are a few examples of the SM types available. For brevity of discussion, assume that RS approximates each of the elements of $Y_0^\wedge$. Treating each RS as a leaf in a sheaf, the resulting database is a SRS. It constitutes an approximate model in the Q space of the BB optimized using the U coordinates, so that

$$Y_0^{\wedge a} = Y_0^{\wedge a}(\mathrm{SRS}(Q)) \tag{A1a}$$

and, optionally,

$$U_0^a = U_0^a(\mathrm{SRS}(Q)) \quad \text{for } QL \le Q \le QU \tag{A1b}$$

where superscript a denotes approximate values, and the notation $Y_0^{\wedge a} = Y_0^{\wedge a}(\mathrm{SRS}(Q))$ means that the approximate values are retrieved from precomputed SRS. The bounds on Q, QL, and QU are the best estimates, accounting for any side constraints and incorporating the move limits necessary for the ensuing interlevel iteration. The SRS is a domain approximation because it covers the entire Q space within the preceding bounds.

In principle, a similar RS approximation may be constructed for each element of $X\mathrm{loc}_0$. However, the resulting volume of data to be stored may be so large as to make this impractical. If so, $X\mathrm{loc}_0$ may be regenerated by reexecuting step 2 of Problem 8 for Q_0 whenever needed. Therefore, $U_0^a = U_0^a(\mathrm{SRS}(Q))$ is labeled as optional in Eq. (A1b).

To control the cost of RS, the number of variables in Q may be reduced by a condensation technique. Such condensation is imperative for those variables in Q that represent field quantities (e.g., the field of the pressure loads distributed over the wing) and the corresponding field of displacements. A condensation of field data may be accomplished by defining the field variable $P(\{v\})$ in the space of $\{v\}$ as

$$P(v) = P^a(p_i f_i(v)) \tag{A2}$$

where the numbers of the parameters p_i and the basis functions $f_i(\{v\})$ are made as small as possible.[†] The p parameters enter the procedures as if they were Y^* and $Y^\wedge$, each p being represented by its own RS.

When all the BBs are optimized, the approximations in Problem 8 for each BB are available to the system optimization that executes next.

III. System Optimization

The system-level optimization problem statement is as follows:

Problem 9:

1) Given a set of SRS, one SRS for each BB

2) Find $\{Q\} = \{X\mathrm{sh}|Y^*|w\}$.

3) Minimize $F(Q) = Y_{0s}^{\wedge a}$.

4) Satisfy $c = Y^* - Y_0^{\wedge a}(\mathrm{SRS}(Q)) = 0$ and $QL \leq Q \leq QU$.

5) Output Q_0 and F_0.

Problem 9 may be solved by any optimization technique, the data $Y_{0s}^{\wedge a}$ needed to calculate $F(Q)$ and $c(Q)$ being retrieved per Eq. (A1a) from the SRS database. The efficiency of the technique is not critical because that data retrieval is nearly instantaneous. Step 4 of Problem 9 enforces the inter-BB couplings. Analogous to $h(U) = 0$ in step 3 of Problem 1, $c = 0$ in step 4 of Problem 9 is tantamount to a system analysis; thus, the BLISS algorithm implements, in effect, the SAND method at the system level, but not necessarily at the BB level, for which the choice of methods for solving Problem 8 is autonomous.

IV. Iterative Procedure

The retrieval of data from SRS [as in Eq. (1a)] in a nonlinear system introduces an error, $\varepsilon = Y_0^\wedge - Y_0^{\wedge a}$, for which the control requires iteration between Problems 8 and 9. That iteration adjusts the move limits in QL and QU in step 4 of Problem 9 to balance the reductions of the width of the QL–QU brackets for improved accuracy against the total number of iterations.

Algorithm 2: A step-by-step recipe for BLISS 2000 optimization algorithm is as follows:

0) Start.

1) Initialize variables (system and local) Xsh, Xloc, Y^*, w, and their U and L bounds.

2) Initial system analysis by solving Problem 8:

a) $h = 0$ in all BBs (i.e., perform analyses in the individual BBs) to obtain $Y^\wedge$.

b) $c = 0$ to obtain new Y^*

c) Iterate steps 2a and 2b until convergence.

(This step improves the starting point for the remainder of the procedure by facilitating establishment of the RS bounds and Y^* values. It is optional.)

3) Approximate model development for each BB. This step may be done simultaneously for all the BBs.

a) Reduce dimensionality of the Q space per Eq. (A2).

b) Disperse, by a suitable DOE technique, a minimum number of points required to define an RS approximation in the BB Q space, bounded by QL and QU.

c) Solve Problem 8 in subspace U at the preceding points in space Q. This may be done simultaneously for all the points within Q.

d) Fit a SRS to the results of step 3b.

e) Verify quality SRS by random sampling. If needed, add new points and discard old points, and use a least-squares fit (or an equivalent technique) to improve the SRS quality.

f) After each system optimization (step 4, next), shift, extend, or shrink the intervals QL and QU to avoid excursions beyond the SRS bounds and to maintain the approximation quality.

4) Solve system optimization Problem 9 in space Q, accessing the SRS data per Eq. (1a).

5) Check the termination criteria. Exit or repeat from step 3 using SRS already available or updated per steps 3e and 3f.

6) Retrieve the optimal Xloc (option: reexecute Problem 8 for Q_0).

7) Stop.

One execution of the procedure from step 2 through step 4 is the major iteration of the cycle.

[†]For example, displacements P normal to a wing may be represented by a function in the chordwise and spanwise coordinates $\{v\}$. Making the function f_i be the product of polynomials linear chordwise and cubic spanwise, for which the coefficients are the parameters p, reduces the number of parameters p to only five.

References

[1] Balling, R. J., and Sobieszczanski-Sobieski, J., "Optimization of Coupled Systems: A Critical Overview of Approaches," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 6–17.

[2] Sobieszczanski-Sobieski, J., and Haftka, R. T., "Multidisciplinary Aerospace Optimization: Survey of Recent Developments," *Structural and Multidisciplinary Optimization*, Vol. 14, No. 1, Aug. 1997, pp. 101–119.

[3] DeLaurentis, D., and Callaway, R., "A System-of-Systems Perspective for Public Policy Decisions," *Review of Policy Research*, Vol. 21, No. 6, Nov. 2004, pp. 829–837. doi:10.1111/j.1541-1338.2004.00111.x

[4] DeLaurentis, D., Lewe, J.-H., and Schrage, D. P., "Abstraction and Modeling Hypothesis for Future Transportation Architectures," AIAA International Air and Space Symposium and Exposition: The Next 100 Years, Dayton, Ohio, AIAA Paper 2003-2514, 14–17 July 2003.

[5] Abbott, T. S., Hamilton, B. A., Jones, K. M., Consiglio, M. C., Williams, D. M., and Adams, C. A., "Small Aircraft Transportation System, Higher Volume Operations Concept: Normal Operations," NASA TM-2004-213022, Aug. 2004

[6] Meyer, M. D., and Miller, E. J., "*Urban Transportation Planning: A Decision-Oriented Approach*," 2nd ed., McGraw–Hill, New York, 2001.

[7] Anon., "Future Flight: A Review of the Small Aircraft Transportation System Concept—Special Report 265," *National Academy Press*, Washington, D.C., 2002.

[8] Mavris, D. N., and DeLaurentis, D. A., "Methodology for Examining the Simultaneous Impact of Requirements, Vehicle Characteristics, and Technologies on Military Aircraft Design," *Proceedings of the 22nd Congress of the International Council of Aeronautical Sciences*, International Council of Aeronautical Sciences, Harrogate, England, U.K., Aug. 2000, pp. 19–31.

[9] Sterman, J. D., *Business Dynamics: Systems Thinking and Modeling for a Complex World*, Irwin/McGraw–Hill, Boston, 2000.

[10] Stanley, D. O., "A Space Transportation Architecture for the Future," *Acta Astronautica*, Vol. 47, Dec. 2000, pp. 265–274. doi:10.1016/S0094-5765(00)00066-7

[11] Stanley, D. O., "NASA Exploration Systems Architecture: Study Results" Space Systems Engineering Conference, Atlanta, GA, Georgia Inst. of Technology Paper SSEC-Plenary 05, Oct. 2005.

[12] Stanley, D. O., Cook, S., Connolly, J., and Hanley, J., "Exploration Systems Architecture Study: Overview of Architecture and Mission Operations Approach," *Journal of Spacecraft and Rockets* (submitted for publication); also *SpaceOps 2006*, AIAA, Reston, VA, June 2006, pp. 76–87.

[13] Stanley, D. O., and Wilhite, A. W., "Technology Engineering: The Concurrent Development of Space Transportation Systems and Technology," *Engineering Management Journal* (to be published); also *Space 2006*, AIAA, Reston, VA, Sept. 2006, pp. 139–152.

[14] Sobieszczanski-Sobieski, J., Agte, J. S., Sandusky, R. R., Jr., "Bilevel Integrated System Synthesis," *AIAA Journal*, Vol. 38, No. 1, pp. 164–172.

[15] Agte, J. S., Sobieszczanski-Sobieski, J., and Robert R. Sandusky, Jr., "Supersonic Business Jet Design Through Bi-Level Integrated System Synthesis," SAE World Aviation Congress, San Francisco, Society of Automotive Engineers Paper 1999-01-5622, Oct. 1999.

[16] Sobieszczanski-Sobieski, J., Altus, T., Phillips, M., and Sandusky, R. Jr., "Bi-Level Integrated System Synthesis for Concurrent and Distributed Processing (BLISS 2000)," *AIAA Journal*, Vol. 41, No. 10, Oct. 2003, pp. 1996–2003; 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, AIAA

Paper 2002-5409, Sept. 4–6, 2002.
[17] Woyak, S., Kim, H., Mullins, J., and Sobieszczanski-Sobieski, J., "A Web-Centric Architecture for Deploying Multi-Disciplinary Engineering Design Process," 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, AIAA Paper 2004-4498, 2004.
[18] Kim, H., Mulllins, J., Soremekun, G., Haisma, M., and Sobieszczanski-Sobieski, J., "A Parallel Trade Study Architecture for Design Optimization of Complex Systems," 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Austin, TX, AIAA Paper 2005-2201, 18–21 Apr. 2005.
[19] Steward, D. V., *Systems Analysis and Management: Structure, Strategy and Design*, Petrocelli, New York, 1981.
[20] Brown, N. F., and Olds, J. R., "Evaluation of Multidisciplinary Optimization (MDO) Techniques Applied to Reusable Launch Vehicle," 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 2005-0707, 10–13 Jan. 2005.
[21] Mavris, D., and Osburg, J., "Design Methodology for Revolutionary Aerospace Concepts," Aerospace Systems Design Lab., Guggenheim School of Aerospace Engineering, Georgia Inst. of Technology, Rept. NAG-1-02023, Atlanta, GA, Dec. 31, 2004.
[22] Sobieszczanski-Sobieski, J., Venter, G., "Imparting Desired Attributes in Structural Design by Means of Multi-Objective Optimization," *Structural and Multidisciplinary Optimization*, Vol. 29, No. 6, June 2005, pp. 432–444.

A. Messac
Associate Editor