Selection-Integrated Optimization (SIO) Methodology for Optimal Design of Adaptive Systems

Ritesh A. Khire

Achille Messac

Corresponding Author

Achille Messac, Ph.D.
Distinguished Professor and Department Chair
Mechanical and Aerospace Engineering
Syracuse University, 263 Link Hall
Syracuse, New York 13244, USA

Email: messac@syr.edu
Tel: (315) 443-2341
Fax: (315) 443-3099
https://messac.expressions.syr.edu/

Bibliographical Information
Selection-Integrated Optimization (SIO) Methodology for Optimal Design of Adaptive Systems

Many engineering systems are required to operate under changing operating conditions. A special class of systems called adaptive systems has been proposed in the literature to achieve high performance under changing environments. Adaptive systems acquire this powerful feature by allowing their design configurations to change with operating conditions. In the optimization of the adaptive systems, designers are often required to select (i) adaptive and (ii) nonadaptive (or fixed) design variables of the design configuration. Generally, the selection of these variables and the optimization of adaptive systems are performed sequentially, thus being a source of suboptimality. In this paper, we propose the Selection-Integrated Optimization (SIO) methodology, which integrates the two key processes: (1) the selection of the adaptive and fixed design variables and (2) the optimization of the adaptive system, thereby eliminating a significant source of suboptimality from adaptive system optimization problems. A major challenge to integrating these two key processes is the selection of appropriate fixed and adaptive design variables, which is discrete in nature. We propose the Variable-Segregating Mapping-Function (VSMF), which overcomes this challenge by progressively approximating the discreteness in the design variable selection process. This simple yet effective approach allows the SIO methodology to integrate the selection and optimization processes and helps avoid one significant source of suboptimality from the optimization procedure. The SIO methodology finds its applications in a variety of other engineering fields, such as product family optimization. However, in this paper, we limit the scope of our discussion to adaptive system optimization. The effectiveness of the SIO methodology is demonstrated by designing a new air-conditioning system called Active Building Envelope (ABE) system.

DOI: 10.1115/1.2965365

1 Introduction

Change can be a pervasive aspect of many engineering systems. In practice, many engineering products or systems are required to operate under changing operating conditions or design requirements. For example; an air-conditioning system may need to operate under changing weather conditions, and an aircraft is often subjected to diverse flight conditions during take-off, cruise, and landing. Under changing environments, a fixed configuration system, or a fixed system, may not offer an overall optimal performance. For example, a truss containing bars of fixed cross-sectional areas may not be the lightest possible truss at all loading conditions. A special class of systems called adaptive systems attempts to overcome the nonoptimal behavior of fixed systems, by maintaining near optimal performance under changing operating conditions. Adaptive systems acquire this powerful feature by changing critical aspects of their design configurations. In the optimization of adaptive systems, the designer is often required to select (i) adaptive and (ii) nonadaptive (or fixed) design variables of the design configuration. As we discuss later, this selection is typically based on two conflicting criteria: (i) adaptivity, which measures the benefits of changing the design configuration, and (ii) penalty, which measures the consequences of this change. In this paper, we propose the Selection-Integrated Optimization (SIO) methodology to design adaptive systems. The SIO methodology integrates two key processes that affect the above criteria: (i) the selection of the design variables and (ii) the optimization of the adaptive system. This integration helps avoid a significant source of suboptimality from the adaptive system optimization formulations, which is in contrast with the prevailing approaches in the literature.

1.1 Adaptive System.

In the literature, adaptive systems are referred to by different names such as flexible systems [1,2] and reconfigurable systems [3]. Adaptive systems have recently been receiving greater attention, because of their ability to offer near optimal performance under changing scenarios. NASA’s morphing aircraft project is conceptually based on utilizing the benefits of adaptive systems [4]. These aircraft are designed to change their configuration to maintain a high level of performance under radically different flight conditions [5]. Robots that adapt to their changing surroundings and applications [6], and adaptive engines that can effectively operate under changing driving requirements [7] are some of the other recent industrial applications of adaptive systems.

As mentioned earlier, adaptive systems can change their design configuration—a key feature that distinguishes them from fixed systems. Typically, a design configuration is defined in terms of pertinent design variables. To change their design configuration, adaptive systems have a special class of design variables that can change their magnitudes as needed. We call these design variables adaptive design variables [2]. Consider a simplified design of an adaptive wing for a morphing aircraft (Fig. 1). This wing will be subjected to changing operating conditions such as take-off, cruise, and landing. For simplicity, assume that the two design variables for this wing are the wing area, $S$, and the sweep angle, $\theta$ (see Fig. 1(a)). The objective is to minimize the total drag at every stage of flight. We note that the design of the adaptive wing shown here is for discussion purposes only and does not reflect the complexity of the actual wing design.

As shown in Fig. 1(b), the sweep angle can be adjusted as per...
the operating requirements and hence, it is called an adaptive design variable. We can make an interesting observation from Fig. 1. Based on the selection of the fixed and adaptive design variables, this wing can have four design alternatives as follows:

1. Both, wing area and sweep angle fixed, Fig. 1(a)
2. Fixed wing area and adaptive sweep angle, Fig. 1(b)
3. Fixed sweep angle and adaptive wing area, Fig. 1(c)
4. Both wing area and adaptive sweep angle, Fig. 1(d)

1.2 Selection of Appropriate Design Alternative. Different design alternatives of an adaptive system differ in terms of their level of adaptivity and penalty associated with each alternative. Here, we define the level of adaptivity as a measure of the ability of the system to offer optimal performance under a given changing environment (minimum drag at all stages of the flight). On the other hand, we define the penalty as a measure of such factors as (i) increased cost and (ii) complexity of operation—because of the inclusion of the adaptive design variables. For example, alternative (a) being a fixed wing design will have the lowest level of adaptivity along with the lowest penalty of all four design alternatives. On the other hand, alternative (d) will have the highest level of adaptivity along with the highest penalty.

We observe that the selection of a design alternative for an adaptive system involves resolving the trade-off between the level of adaptivity and the penalty. In the optimization community, the process of resolving a trade-off is called multi-objective optimization. For simplicity, we call it the optimization of an adaptive system. From the above discussion, we infer that the selection of the design alternative and the optimization of the adaptive system are interdependent processes. However, most existing methods address them separately—thus leaving a likelihood of designing a suboptimal system. A review of pertinent existing methods is presented in the next subsection.

1.3 Literature Review. The literature review is divided into two parts. In the first part, the literature published under the umbrella of adaptive, flexible, and reconfigurable systems is discussed. Some research problems from the product family optimization area are similar to those of adaptive system optimization. This warrants the discussion of the existing methods for product family optimization and is included in the second part. In fact, the proposed SIO methodology is applicable to some of the product family optimization problems as well.

1.3.1 Adaptive, Flexible, and Reconfigurable Systems. Researchers from different disciplines have studied a variety of adaptive system design problems [8–10]. In the design optimization community, Olewnik et al. [1,11] have presented a framework for developing optimization based methods for flexible systems. Important contributions in these papers include the discussion on the current challenges in the development of optimization based methods for flexible systems and the potential areas of future research. Recently, we proposed an optimization based methodology [2] that addresses the issue of adaptive and fixed design variables selection, with a potential limitation of generating suboptimal designs. Nadir et al. [3] investigated the manufacturing cost benefits resulting from introducing reconfigurability in structural designs.

Parrish et al. [12] used the design-under-uncertainty method to design a layout of an adaptive truck-cabin to accommodate the variation within drivers. In this method, the design variables are modeled as deterministic, and the variation in drivers is modeled as a source of uncertainty. Here, the adaptive and fixed design variables were assumed known a priori. Roser and Kazmer [13] proposed a “flexible design methodology” that designs flexible systems to address design variable and model uncertainties, when a system is subjected to fixed operating conditions. It is important to note that robust design optimization methods are typically used to account for such uncertainties or variations. These methods tend to optimize the design objectives, while at the same time minimize its sensitivity to parameter and design variable variations [14,15]. However, the design that is obtained using such methods is typically fixed (with certain allowed variation) and not adaptive.

Ferguson and Lewis [16] have used state-feedback-control law to determine the needed change in the adaptive design variables, when a system is subjected to time-dependent changes. In the above publication, it is assumed that the selection of the fixed and adaptive design variables is made a priori. When designing time-dependent adaptive systems in which the selection of the fixed and adaptive design variables is assumed known, the designer can use the methods available for simultaneous design and control optimization, see Refs. [17,18].

As stated in Sec. 1.2, one of the challenges in adaptive system design involves selecting appropriate fixed and adaptive design variables such that the resulting design is optimal. On a slightly different note, there are a number of design methods available in the literature [19,20] that attempt to simultaneously select and optimize designs for fixed systems. Since the design of fixed systems is not within the scope of the current paper, such design methods are not discussed.

1.3.2 Product Family Optimization. A product family consists of multiple products that share a common platform. Different products in the family are developed by introducing additional features (design variables) on the platform [21]. We explore the class of product family problems where different products in the family are developed by scaling the added design variables such that each product satisfies a unique requirement [21]. The common platform in the product family is analogous to the fixed design variables in the adaptive systems, and the scalable design variables are analogous to the adaptive design variables. The methods that have been proposed in the literature to optimally design product families generally follow a two-step approach. First, multiple products are designed such that (1) they are not forced to share a common platform, and (2) each product optimally satisfies a specific requirement. Second, depending on the change in the optimal design variable values required to optimally satisfy all product requirements, the designer then selects the design variables for the platform (fixed) and those for scaling (adaptive). The robust concept exploration method proposed by Simpson et al. [21], and the variation based method proposed by Nayak et al. [22] transform the product family problem into a robust design problem. By designing product families from a robust design perspective, the authors attempt to minimize the sensitivity of the product platform to the variation because of scaling design variables. This method may require approximating the standard deviations and the means of the objective function and
the constraints and may also require sampling methods such as Monte Carlo simulation. The methods proposed by Simpson et al. [21] and Nayak et al. [22] use the compromise decision support problem formulation proposed by Mistree et al. [23].

Fellini et al. [24] used the performance bound constraints in the product family optimization problem. In this constraint, an upper limit is set on the deviation between the performance of the product family and that of the product family with no common platform. Fellini et al. [25] also proposed a sensitivity-based commonality strategy applicable when mild variations are required in the scaled design variables to obtain different product variants. In all of the above methods, the design variable selection is performed separately from the optimization of the product family.

There is another class of methods available in the literature that follows an exhaustive search technique. In an exhaustive search technique, multiple product families, each containing a unique set of platform design variables, are individually optimized and compared. In such methods, the number of possible product families (and the number of optimization problems) increases with the number of design variables (n design variables = 2^n possible product families). Hence, these methods may become computationally prohibitive for systems with a large number of design variables. D’Souza and Simpson [26] used a genetic algorithm to optimally design product families, where design variables were divided in small groups to reduce the number of possible product families. The use of a genetic algorithm requires extensive fine tuning, which in many cases is problem dependent. Gonzalez-Zugasti et al. [27] used a team based negotiation model to design product families. In this model, the designers manually select the platform design variables from the preselected alternatives, which is primarily based on the level of expertise and the prior knowledge of the individuals. Many methods proposed in the literature [28, 29] assume that the selection of the platform and scalable design variables is made a priori. As such these methods are not directly relevant to the adaptive system optimization problem addressed in this paper. However, these methods can be used to optimize individual product families under the exhaustive approach, if one should decide to use them.

1.4 Motivation for the SIO Methodology. The literature review suggests that the identification of the fixed and adaptive design variables is an important research topic in the product family and adaptive systems fields. Most existing methods separate the process of design variable selection from that of the optimization of the adaptive system—thus leaving a likelihood of designing a suboptimal system. The combinatorial and discrete nature of the fixed and adaptive design variables selection process is a major challenge to its integration with the adaptive system optimization process. In this paper, we propose the SIO methodology, which integrates (1) the selection of the fixed and adaptive design variables and (2) the optimization of the adaptive system. This unique feature of the SIO methodology is expected to avoid one significant source of suboptimality from the adaptive system optimization procedure. To integrate the two key processes, we propose the Variable-Segregating Mapping-Function (VSMF) that progressively approximates the inherent discreteness involved in the selection of design variables.

The remainder of this paper is organized as follows. In the next section, we present the research problem addressed in this paper. The overview of the SIO methodology is presented in Sec. 3. The mathematical details are presented in Sec. 4. In Sec. 5, we present a numerical example solved using the SIO method. Finally, we provide concluding remarks in Sec. 6.

2 Research Problem and Background for VSMF

2.1 Optimization Problem. We use the optimization problem as a basis for the development of the SIO methodology. A typical design optimization problem is given as

\[ \min \mu(x, P) \]

subject to

\[ g(x, P) \leq 0 \]

\[ x_{\text{min}} \leq x \leq x_{\text{max}} \]

where \( \mu \) is a measure of the system performance and is either an objective function of a single objective problem or an aggregate objective function (AOF) of a multiobjective problem [30], \( x = [x_1, x_2, \ldots, x_n] \) is a vector of \( n \) design variables, \( g \) is a vector of constraints, and \( P \) is a set of design parameters that represents the operating condition or design requirement (referred to here and onwards as operating condition).

Solely for the sake of presentational simplicity and visual clarity, we will assume that \( P \) is a scalar. We note that this assumption does not present any limitations to the application of the SIO methodology. We also note that we do not address the issue of AOF formation in this paper. The SIO methodology presumes that an appropriate AOF is available in the case of multiobjective problems. Next, let us provide a generic description of adaptive systems.

2.2 Changing Operating Conditions. The solution of the above optimization problem is a design variable vector \( x^* = [x_1^*, x_2^*, \ldots, x_n^*] \) that minimizes the objective function, \( \mu \), while satisfying all the constraints—for a fixed operating condition, \( P \). However, in this paper, we consider the case where the operating conditions deterministically change between predefined minimum and maximum values \( P_{\text{min}} \) and \( P_{\text{max}} \), respectively. To design a system that performs optimally at all operating conditions, one would theoretically need to solve the optimization problem given in Eqs. (1)–(3) for all values of \( P \) between \( P_{\text{min}} \) and \( P_{\text{max}} \). Consider Fig. 2(a) that shows the optimal objective function values for a hypothetical two-design variable system operating between \( P_{\text{min}} \) and \( P_{\text{max}} \).

2.2.1 Fully Adaptive System. For a system to perform optimally all the time, it needs to adapt itself to the changing operating conditions by somehow changing its design configuration. The change in the design configuration implies that certain design variables take on different values at different operating conditions. Figure 2(b) shows the optimum values of two-design variables \( x_1 \) and \( x_2 \), across changing operating conditions. As shown in Fig. 2(b), the two-design variables change their magnitudes (or adapt themselves to the changing operating conditions) and are termed adaptive design variables. We call a system fully adaptive if all its design variables are adaptive.

2.2.2 Adaptive System. In spite of its potential benefits, fully adaptive systems may not be practical from either an economic or a complexity of operation point of view. Instead, a decision needs to be made regarding which design variables need to be fixed, and which ones should be kept adaptive. We call a system that comprises both the fixed and adaptive design variables an adaptive
system. In the SIO methodology, we realize that the segregation of the fixed from the adaptive design variables involves resolving the trade-off between performance and penalty, as discussed in the previous section. To better understand this trade-off, consider a generic optimization problem that is often solved in the adaptive system optimization.

2.3 Optimization of Adaptive System. In the fully adaptive system, the design variables are allowed to change largely without restriction, such that the objective function, \( \mu \), is minimized at a set of operating conditions. For the fully adaptive system shown in Fig. 2(b), the change in the design variables \( x_1 \) and \( x_2 \) is denoted by \( \Delta x_1 \) and \( \Delta x_2 \), respectively. To design adaptive systems, we use the optimization problem formulation typically used in product family optimization, where one often simultaneously minimizes the objective, \( \mu \), and the change in the design variables, \( \Delta x_{th} \) [22]. Doing so alleviates the consequences of the change in the design variables. Here, \( \mu \) is the performance measure of the adaptive system and minimizing \( \mu \) leads to maximizing its performance. The change, \( \Delta x_{th} \), in the generic design variable, \( x_{th} \), is a measure of the penalty associated with this design variable. Thus, minimization of each \( \Delta x_{th} \) corresponds to minimizing its penalty on the adaptive system.

Unfortunately, performance and penalty are generally conflicting in nature—improvement in one leads to worsening of other. To resolve this trade-off, a bi-objective optimization problem is often formulated, as shown in Fig. 3 (Step 1) [22]. Mathematical details of this problem formulation are presented in the implementation section. We note that the mathematical formulation of the problem shown in Fig. 3 (Step 1) may involve some thoughtful heuristic choices (e.g., handling of the two objectives). In the SIO methodology, we do not address the selection or elimination of the heuristic choices involved in the problem formulation. We assume that the formulation of the optimization problem is appropriate to the extent desired by the designer. The motivation of the SIO methodology is to eliminate a significant source of suboptimality involved in the adaptive system optimization procedure caused by the separation of the (i) selection and (ii) optimization processes, as discussed next.

2.4 Existing Approaches to Design Adaptive Systems in a Nutshell. Most existing methods generally, follow a two-step approach to designing adaptive systems, as typified in Fig. 3. In the first step, the changes in the design variables, \( \Delta x \)'s, are determined by solving the bi-objective problem, which simultaneously considers \( m \) operating conditions (i.e., for different values of \( P \)). In the second step, the segregation of the fixed from the adaptive design variables is made based on the \( \Delta x \) values. As shown in Fig. 3, under these approaches, if the change, \( \Delta x_{th} \), is smaller than a prespecified threshold change [24], \( \Delta x_{th} \), the corresponding design variable, \( x_{th} \), is fixed; otherwise it is made adaptive. Let us examine the strategy (for Step 2) of selecting design variables from a graphical perspective in Fig. 4. As shown in Fig. 4, the design variable \( x_{th} \) is fixed if \( \Delta x_{th} \) lies between points \( O \) and \( a \); otherwise it is made adaptive. For the fixed design variables, \( \Delta x_{th} \) is made zero, and for the adaptive variables, \( \Delta x_{th} \) is not changed. Mathematically, Fig. 4 represents a mapping of \( \Delta x_{th} \) onto itself. We call the change prior to this mapping the actual change, \( \Delta x_{act} \), and that after the mapping the mapped-change, \( \Delta x_{map} \).

2.5 Background for Variable-Segregating Mapping-Function (VSMF). One of the limitations of existing two-step approaches is that they do not assess the effect of design variable selection on the adaptive system—thus inherently introducing a possible source of suboptimality. One intuitive approach to avoid this significant source of suboptimality is to incorporate the mapping shown in Fig. 4 in the optimization problem formulation. Conceptually, the proposed SIO methodology is based on this realization. That is, it integrates (i) the selection of design variables and (ii) the optimization problem, as shown in Fig. 5(a). Importantly, it does not assume a priori knowledge of which variable should be fixed and which should be adaptive. This critical decision results from the actual aggregate objective function and not from ad hoc heuristic procedures that are fraught with unquantifiable uncertainties.

However, we note that the mapping shown in Fig. 4 entails two important consequences. First, this mapping is not continuous and therefore is not suitable for gradient based optimizers. We may have to use nongradient based optimizers such as genetic algo-

---

**Fig. 3 Existing two-step approach of segregating design variables**

**Fig. 4 Selecting design variables based on \( \Delta x_{th} \)**

**Fig. 5 Optimization of adaptive system using SIO methodology**
3 Overview of the Selection-Integrated Optimization (SIO) Methodology

In this section, we present an overview of the SIO methodology to help the reader appreciate its premise. As stated earlier, we propose the VSMF to overcome the two consequences of the discontinuous mapping mentioned in the previous section. As its name suggests, the VSMF facilitates the segregation of the fixed design variable. By including the VSMF in the optimization problem, the SIO methodology facilitates the discontinuous mapping mentioned in the previous section. As its name suggests, the VSMF facilitates the segregation of the fixed design variable as a result of the optimization process (see Fig. 5). In this section, we discuss the properties of a generic VSMF, followed by its implementation in adaptive system optimization.

3.1 Variable-Segregating Mapping-Function (VSMF). The VSMF is a family of continuous functions that progressively approximate the discontinuous mapping shown in Fig. 4. We define a generic VSMF that is used for all design variables. Details of the VSMF are provided in Fig. 4(b). We use two normalized n-dimensional variables to define the VSMF: (i) $\Delta x_{\text{act}}$ for actual change, and (ii) $\Delta x_{\text{map}}$ for mapped-change.

The VSMF is defined such that it satisfies the following properties. (1) The VSMF is a monotonically increasing smooth function (continuous first derivative). (2) The threshold value is set at $\Delta x_{\text{act}}=1$, at point $a$ in Fig. 5(b). (3) $\Delta x_{\text{map}}(x_{\text{act}}=0)=0$ (point $O$). (4) $\Delta x_{\text{map}}(x_{\text{act}}=1)=1$ (point $b$). (5) $\Delta x_{\text{map}}(x_{\text{act}}=\alpha)=\Delta x_{\text{act}}$ (segment $b$-$c$). (6) The VSMF contains a point $s$ between $\Delta x_{\text{act}}=0$ and $\Delta x_{\text{act}}=1$ that has an interesting property, defined next.

Point $s$ divides the VSMF into two parts, shown by $O$-$s$ and $s$-$b$ in Fig. 5(b). The coordinates of point $s$ are governed by a special parameter $\alpha$ as

$$s = \left[0.5(2 - \alpha), 0.5\alpha\right]$$

By changing $\alpha$ between 1 and 0, we can obtain a family of VSMFs—a property exploited for segregating the fixed design variables. We note that $\alpha$ is not a design variable in the SIO methodology and is instead a VSMF parameter that facilitates the progressive approximation of the discreteness involved in the design variable selection process. For $\alpha=1$, the VSMF follows the straight line $O$-$a_1$-$b$-$c$ in Fig. 5(b). If we progressively lower the value of $\alpha$ toward zero, point $s$ travels from point $a_1$ to point $a$, thereby causing the VSMF to progressively approximate the original discontinuous mapping shown by $O$-$a$-$b$-$c$ in Fig. 5(b). This progression here has a significant similarity to homotopy based approaches [31]. We make some important observations regarding the discontinuous mapping case.

3.1.1 Segregating Design Variables Using VSMF. From Fig. 5(b) we can observe that as $\alpha$ approaches 0, the VSMF converges to the discontinuous mapping $O$-$a$-$b$-$c$. This implies that segment $O$-$s$ of the VSMF converges to segment $O$-$a$ of the discontinuous mapping. Therefore, for a small value of $\alpha=0$, if the mapped-change $\Delta x_{\text{map}}$ for a design variable $x_{\text{act}}$ lies between points $O$ and $s$, the corresponding design variable converges to a fixed status. Otherwise, the design variable is viewed as converging to the adaptive status and is made so. Because of this property of point $s$, we call it the segregating point. Based on this observation, we develop the following segregation criterion.

$$\text{if } \lim_{\alpha \to 0} (\Delta x_{\text{map}}(x)) = 0.5\alpha \text{ then } x_{\text{act}} \text{ is fixed}$$

where $0.5\alpha$ is the coordinate of the separating point $s$ along the vertical axis, which vanishes as $\alpha$ goes to zero.

3.2 Implementing VSMF in the Optimization Problem. In this subsection, let us discuss the most powerful feature of the SIO methodology—the design variable segregation within the optimization problem. Figure 6 depicts the process of design variable segregation when the VSMF is implemented in the adaptive system optimization problem.

We customize the VSMF for each design variable as follows. We solve the optimization problem shown in the first step of Fig. 3. The solution of this problem is used in two ways. First, for each design variable, we customize the normalized variables $\Delta x_{\text{act}}$ and $\Delta x_{\text{map}}$ such that the solution lies at point $a_1$ on the VSMF, as shown in Fig. 6. (Mathematical details of this customization procedure are given in Sec. 4.) Second, the solution is used as the starting point for the optimization that involves the VSMF. In short, we have completely defined point $a_1$ on the VSMF as the starting point, as shown in Fig. 6.

After customizing the VSMF for all design variables, we formulate the adaptive system optimization problem, as shown in Fig. 5(a). In this problem formulation, we use the normalized value of the mapped-change, $\Delta x_{\text{map}}$, as the measure of the penalty associated with the adaptive design variables. We solve this optimization problem with a value of the VSMF parameter $\alpha=1$. During each optimization iteration, the design variables are mapped using the VSMF. The solution of this optimization problem becomes the starting point for the next one that uses a reduced value of $\alpha$. This procedure is repeated by progressively lowering the value of $\alpha$ in each repetition. In Fig. 6, the stars show typical locations of the optimal solutions obtained from each repetition of the optimization problem. As shown in Fig. 6(a), as the value of $\alpha$ is lowered in each repetition, the design variables that are going to be fixed move closer to segment $O$-$a_0$ on their corresponding VSMFs. We note that on segment $O$-$a_0$, the mapped-change in the design variable is zero and therefore represents a fixed design variable. Also, with each repetition, the adaptive design variables...
move closer to or further than point \( b \) (see Fig. 6(b)). Thus, the SIO methodology segregates the fixed from the adaptive design variables within the optimization problem, which, as discussed earlier, leads to the elimination of a significant source of suboptimality from the optimization procedure.

We can exploit the design variable segregation feature in two ways as follows. (1) We can repeat the optimization problem as many times as required for \( \alpha \) to become sufficiently close to zero. The solution obtained at the end of this procedure is selected as the final design of the adaptive system. In the numerical example presented in this paper, we follow this approach. (2) We can stop the repetitions when the fixed design variables show a “trend” of converging to segment \( O-a_b \). For example, in Fig. 6(a)—Path 1, the stars on the VSMFs labeled \( \alpha = 1 \) and \( 0 < \alpha < 1 \) show a trend that the corresponding design variable is going to be fixed. In this approach, we may have to solve another optimization problem with appropriate fixed and adaptive design variables and obtain the optimal adaptive system design. The second approach requires some judgment regarding when to stop the repetitions of the optimization problem.

We note that in the SIO methodology, the segregation of the fixed from the adaptive design variables is done based on the trade-off between penalty and performance. It is indeed possible that the SIO methodology may select a fully adaptive system. This constraint is called the PLC. In adaptive systems, this deviation is constrained by an upper limit. In such cases, the designer can use metamodeling techniques such as response surfaces to reduce the computational cost.

For the \( i \)th operating condition, \( P_i \), the objective function, design constraint, and design variable vectors are denoted as \( \mu^i \), \( g^i \), and \( x^i \), respectively. Here, the design variable vector, \( x^i \), is given as

\[
x^i = \{x_{1,1}^i, x_{1,2}^i, \ldots, x_{1,m}^i\}, \quad \forall i = 1, \ldots, m
\]

Figure 8 shows the optimization problem formulations at \( m \) operating conditions. After each optimization, we record the optimal objective function value. For the \( i \)th operating condition, the optimal objective function value of the fully adaptive system is denoted as \( \mu_{fa}^i \). In adaptive systems, some design variables are fixed, which is likely to result in a performance loss [24]. We account for this loss using the performance loss constraints (PLC) [24].

4.2 Performance Loss Constraint (PLC). Figure 9 contrasts the fully adaptive and adaptive systems in terms of their optimal objective function values (here, \( m = 4 \)). In Fig. 9, the vertical distance between the black dots (adaptive system) and the gray dots (fully adaptive system) represents the deviation in the objective function value, or the performance loss. In the optimization of the adaptive systems, this deviation is constrained by an upper limit. This constraint is called the PLC [24]. The upper limit is specified in terms of a fraction of the optimal objective function value of the fully adaptive system [24]. At the \( i \)th operating condition, the PLC is given as

\[
\mu^i(x^i, P) - \mu_{fa}^i = K_{i}^L \mu_{fa}^i \quad \forall i = 1, \ldots, m
\]

where \( \mu^i(x^i, P) \) is the objective function value of the adaptive system at the \( i \)th operating condition, and \( K_i^L \) is called the performance loss factor [24]. We have \( m \) PLCs—each corresponding to

---

**Fig. 7 Main steps in the SIO methodology**

**Fig. 8 Optimization of the fully adaptive system**

**Fig. 9 Comparison of objective function values**
one operating condition. All the PLCs are included in the optimization of the adaptive system.

4.3 Optimization Problem for the Adaptive System. We formulate the adaptive system optimization problem such that it simultaneously (1) maximizes the performance and (2) minimizes the penalty. We discuss these two objectives followed by the optimization problem formulation.

4.3.1 Performance Objective. We measure the performance of the adaptive system in terms of a set of m defined optimal objective function values. We wish the adaptive system to have the defined optimal objective function values (performance) as close as possible to those of the fully adaptive system. Accordingly, we minimize the difference between the objective function values of the adaptive system and those of the fully adaptive (benchmark) system. Specifically, the performance objective function, \( f_{\text{per}} \), is given as

\[
 f_{\text{per}} = \sum_{i=1}^{m} \mu_i(x^i, P^i) - \mu_{i_a}
\]

Minimizing the above objective function results in maximizing the performance of the adaptive system.

4.3.2 Penalty Objective. We assume that the change in the adaptive design variable value adds to the complexity of operation and/or manufacturing cost. Accordingly, we define the penalty as a monotonically increasing function of the change in the design variable values. We use an expression similar to that used in Ref. [24] to denote the penalty objective function as

\[
 f_{\text{pen}} = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \left[ \sum_{k=i+1}^{n} D_j(x_k^j - x_k^i)^2 \right]
\]

where

\[
 D_j(x_k^j - x_k^i) = 1 - \frac{1}{\left( \frac{x_k^j - x_k^i}{\gamma} \right)^2 + 1}
\]

Here, \( x_k^j \) represents the value of the kth design variable at the ith operating condition, and \( \gamma = 0.05 \) [24]. Minimizing the above objective function results in minimizing the penalty. We note that the above penalty function estimates penalty based on the change in the design variable values. However, in practice, the cost/complexity of making some design variables adaptive is likely to be higher than that of others. In such cases, the penalty function should be sufficiently sophisticated to differentiate between (i) the cost/complexity due to the changes in adaptive design variable values and (ii) that due to the nature of a given adaptive design variable. However, the motivation of the SIO methodology is to integrate the selection and optimization processes, and its applicability is not diminished by the particulars of the penalty function formulation. Hence, we do not address the issue of penalty function formulation in this paper. Reference [32] by Thevenot et al. compares various penalty function formulations (called commonality index) and should provide some insight in this issue to the interested reader. Based on the above two objectives, we formulate a bi-objective optimization problem.

4.3.3 Bi-Objective Optimization Problem. We formulate the bi-objective optimization problem for the adaptive system as

\[
 \min_{x_1 \ldots x_n} (f_{\text{per}}, f_{\text{pen}})
\]

subject to

\[
 g^i(x^i, P^i) \leq 0, \quad \forall i = 1 \ldots m
\]

\[
 \mu(x^i, P^i) - \mu_{i_a} \leq \mu_{i_a} - \mu_{i_a}, \quad \forall i = 1 \ldots m
\]

After solving the above optimization problem, we record the optimum design variable values. At the ith operating condition, the optimum design variable vector is given by

\[
 x^i = (x_1^i, x_2^i, \ldots, x_n^i), \quad \forall i = 1, \ldots, m
\]

These optimal design variable values are used to determine the penalty values in the design variables. For the kth design variable, \( x_k \), the change, \( \Delta x_k \), is determined as

\[
 \Delta x_k = \max(s_k^i) - \min(s_k^i), \quad \forall k = 1, \ldots, n
\]

where vector \( s_k^i \) stores the optimum values of the design variable \( x_k \) at various operating conditions, and

\[
 s_k^i = (x_1^k, x_2^k, \ldots, x_n^k), \quad \forall k = 1, \ldots, n
\]

The changes in the design variable values obtained from Eq. (17) are used to map the design variables using the VSMF.

4.4 Variable-Segregating Mapping-Function (VSMF). We construct the VSMF using two normalized nondimensional variables: (i) \( \Delta x_{\text{act}} \) for the actual change and (ii) \( \Delta x_{\text{map}} \) for the mapped-change. In this paper, we use cubic splines available in MATLAB to construct the VSMF shown in Fig. 5(b). In Table 1, we give the coordinates and the slopes at the key points that we used to construct the VSMF. We note that the cubic spline is but one of the functions that can potentially be used to define the VSMF. The choice of the constituent function of the VSMF may affect its computational properties (e.g., computational cost). A detailed investigation of different functions that can be used to define the VSMF is a good topic for the future work.

4.4.1 Redefined Penalty Objective Function. From Eq. (5) we recall that the design variable, \( x_k \), is fixed if the normalized mapped-change, \( \Delta x_{\text{map}} \), is smaller than 0.5\( \sigma \) as \( \sigma \to 0 \). For computational reasons, it is a good idea to use the redefined penalty objective function as

\[
 \tilde{f}_{\text{pen}} = \sum_{k=1}^{n} \left[ (\Delta x_{\text{map}}^k - 0.5\sigma) \right]
\]

Table 1 Properties of the points on generic VSMF

<table>
<thead>
<tr>
<th>Point</th>
<th>( \Delta x_{\text{act}} )</th>
<th>( \Delta x_{\text{map}} )</th>
<th>Slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O )</td>
<td>0</td>
<td>0</td>
<td>–</td>
</tr>
<tr>
<td>( s )</td>
<td>0.5(2–( \alpha ))</td>
<td>0.5(( \alpha ))</td>
<td>( \alpha )</td>
</tr>
<tr>
<td>( b )</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( c )</td>
<td>10</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

We note that Eqs. (10) and (19) are practically equivalent. In other words, Eqs. (10) and (19) both estimate the penalty based on the change in the design variable values. We observed that the penalty function defined in Eq. (19) segregates the fixed from the adaptive design variables (makes \( \Delta x_{\text{map}}^k \) smaller than does Eq. (10). Such behavior is contributed to the fact that the penalty function defined in Eq. (19) is a direct variant of the design variable segregation condition defined in Eq. (5).

4.5 Adaptive System Optimization Problem Definition. Based on the above penalty function and the VSMF, we define the adaptive system optimization problem as

\[
 \min_{x_1 \ldots x_n} (\tilde{f}_{\text{per}}, \tilde{f}_{\text{pen}})
\]

such that

\[
 x_{\text{min}} \leq x^i \leq x_{\text{max}}, \quad \forall i = 1 \ldots m
\]
(\(x^1, \ldots, x^m\)) = VSMF(\(\alpha, x^1, \ldots, x^m\))  \text{ map design variables}  
\text{(21)}

subject to 
\[g'(\tilde{x}^i, P') \leq 0, \quad \forall \ i = 1 \ldots m\]  
\text{(22)}

\[\mu'(\tilde{x}^i, P') - \mu'_i \leq k'_i \mu'_i, \quad \forall \ i = 1 \ldots m\]  
\text{(23)}

\[x_{\min} \leq \tilde{x}^i \leq x_{\max}, \quad \forall \ i = 1 \ldots m\]  
\text{(24)}

\[\alpha = 0\]  
\text{(25)}

In the above problem, \(\tilde{x}^i\) represents the mapped value of the design variable vector \(x^i\). The solution to the above problem yields the optimal design for the adaptive systems discussed in this paper. Because \(\alpha = 0\) yields a discontinuous problem statement, we have developed a systematic procedure that allows us to use the gradient based optimization methods for the above problem.

4.6 Solution Procedure for Solving Optimization Problem With VSMF. The final task in the SIO methodology involves solving the adaptive system optimization problem given by Eqs. (20)–(25). The procedure to do so involves the following eight steps, of which Steps 4–8 are repeatedly executed. These repeating steps are pictorially described in the large box in Fig. 7.

1. Use the optimum design variable values from Eq. (16) as the starting point.
2. For each design variable, \(x_k\), define the threshold change as
\[(\Delta x_k)_h = 2 \Delta x^*_k, \quad \forall \ k = 1, \ldots, n\]  
\text{(26)}

where \(\Delta x^*_k\) is obtained from Eq. (17). This threshold change is a constant in the SIO methodology and is used to map the design variables using the VSMF. The procedure of mapping the design variables is discussed in the next subsection.
3. Set \(\alpha = 0.9\).
4. Construct the VSMF using the current value of \(\alpha\).
5. Solve the following optimization problem:
\[
\min \left\{ f_{\text{per}}, f_{\text{pen}} \right\} \quad \text{subject to} \quad \tilde{x}^i - x^i \leq \Delta x^*_i, \quad \forall \ i = 1 \ldots m
\]
\text{(27)}

such that
\[(\tilde{x}^1, \ldots, \tilde{x}^m) = VSMF(\alpha, x^1, \ldots, x^m) \quad \text{map design variables} \]  
\text{(28)}

6. Replace the starting point with the solution of the above optimization problem.
7. Lower the value of \(\alpha\) as follows: \(\alpha = \alpha - \Delta \alpha\), where \(\Delta \alpha = 0.1\).
8. Repeat the entire procedure from Steps 4 to 7 until \(\alpha = 0.2\).

In the above procedure, we use \(\Delta \alpha = 0.1\) in Step 7. We note that judicial selection of \(\Delta \alpha\) is important from (1) computational cost and (2) convergence perspectives. Choosing a small \(\Delta \alpha\) value can be computationally prohibitive, and choosing a large value is likely to negatively impact the convergence of the solution. As such, an argument can be made that parameter \(\Delta \alpha\) is heuristic in nature and so is the SIO methodology. However, we also like to state that the selection of \(\Delta \alpha\) is not arbitrary. Our considerable experience to date with the SIO methodology suggests that 0.05 \(\leq \Delta \alpha \leq 0.1\) seems to satisfy both of the above considerations. However, a more comprehensive evaluation of this parameter in the future would be helpful.

At the end of the above procedure, the design variables that satisfy the condition given in Eq. (5) become the fixed design variables, and the remainders are made adaptive. A relatively straightforward way to confirm the convergence of the design obtained using \(\Delta \alpha = 0.1\) in the above procedure is to repeat it with \(\Delta \alpha < 0.1\) (e.g., 0.07). Convergence to the same design would increase confidence in its optimality. This approach may seem to involve additional computational cost. However, the designer need not actually continue the SIO methodology repetitions until \(\alpha\) becomes 0.2 in Step 8. Instead, he/she can terminate the methodology once the design variables show a trend of segregating the fixed from the adaptive design variables, as discussed in Sec. 3.2. Future work in establishing increased confidence in the optimality of the obtained solution would be helpful. As such, a formal mathematical evaluation of the SIO methodology to evaluate its behavior in generating optimal solutions is in the order in the future.

In the above procedure, the mapping of the design variables is performed in Eq. (28), the details of which are as follows.

4.6.1 Mapping the Design Variables Using Generic VSMF. To map the change in the design variables using the VSMF, we normalize the former. For the design variable \(x_k\), we perform the normalization using the threshold change \((\Delta x_k)_h\) determined in Eq. (26). For each design variable, the mapping involves the following process.

1. Calculate the actual change \(\Delta x_k\) for each design variable \(x_k\) as
\[
\Delta x_k = \max(s_k) - \min(s_k)
\]
\text{(32)}

where
\[s_k = [x_k^1, x_k^2, \ldots, x_k^m]\]
\text{(33)}

2. Determine the normalized value of the actual change \((\Delta x_k)_{\text{act}}\).
\[
(\Delta x_k)_{\text{act}} = \frac{\Delta x_k}{(\Delta x_k)_h}
\]
\text{(34)}

3. Substitute the actual change \((\Delta x_k)_{\text{act}}\) in the VSMF and obtain the mapped-change \((\Delta x_k)_{\text{map}}\).
4. The mapped value of the design variable values is determined as
\[
\tilde{x}_k = \min[(s_k) + (\Delta x_k)_{\text{map}}(s_k - \min(s_k)), \quad \forall \ i = 1, \ldots, m
\]
\text{(35)}

In the above equation, \(\tilde{x}_k\) represents the mapped value of the design variable \(x_k\).

This completes the discussion of the implementation of the SIO methodology. Next, we apply the SIO methodology to an air-conditioning system called active building envelope (ABE) system. We design the ABE system for the changing temperature between spring and summer. We note that this is a hypothetical case study, intended to demonstrate the application of the SIO methodology.

5 Numerical Example: Active Building Envelope (ABE) System

An ABE system is a new technology that proposes to actively use solar energy to maintain a comfortable indoor environment in buildings [33]. In ABE systems, solar radiation energy is converted into electrical energy by means of a photovoltaic unit (PV unit). Subsequently, this electrical energy is used to power a ther-
moelectric heat pump unit (TE unit). Among the key differences between an ABE system and conventional thermal control technologies are that the former (i) is intended to operate using solar energy [34], (ii) is made of solid-state devices and operates silently with no moving parts, (iii) can be used for heating as well as cooling applications, (iv) uses little or no fossil energy sources, and (v) should result in important long-term environmental benefits. In this paper, we design an ABE system that maintains a constant indoor temperature of 20°C, when the outside temperature changes with season.

5.1 Active Building Envelopes. A brief description of the ABE system is provided here (Fig. 10), for more details, see Refs. [33–36]. As shown in Fig. 10, the ABE system comprises a PV unit and a TE unit. The PV unit consists of photovoltaic cells (solar cells), which are solid-state devices that convert solar radiation energy into electrical energy. The TE unit consists of thermoelectric heaters/coolers (referred to here onwards as TE coolers), which are solid-state devices that convert electrical energy into thermal energy, or the reverse. The gap between the PV unit and the external wall acts as an external heat dissipation zone for the TE unit (see Fig. 10). The external walls of the ABE system consist of two layers. The external layer facing the PV unit is made of a good thermal insulating material, and the internal layer is made of a material with high heat storage capacity.

In Fig. 10, the words “Thermal Insulation” and “Thermal Mass” pertain to the external and the internal layers of the ABE wall, respectively. TE coolers are dispersed inside the openings provided in the insulating layer. Each TE cooler consists of two heat sinks. The internal heat sink either absorbs or dissipates heat to the thermal mass layer. The external heat sink either absorbs heat from or dissipates heat to the surrounding air, through natural or forced convection. Next, we describe the computationally favorable approximate analytical model of the ABE system that we developed for this study.

5.2 Model of ABE System. Figure 11 shows the model of the ABE system that integrates those of the PV unit, the TE unit, and the house. We assume that the outside temperature changes with season. The changing temperature affects the amount of heat entering the house, also called the cooling load, and is determined by the model of the house. The TE unit is designed to absorb the cooling load. The TE unit consists of TE coolers placed in a grid formation, see Fig. 11. We determine the input voltage and the input current required to operate the TE unit from the number of TE coolers connected in series, $H_S$ and in parallel $H_P$, as shown in Fig. 11.

The TE unit is powered by the PV unit, which is also a grid of solar cells (see Fig. 11). By using the appropriate number of solar cells in series $S_S$ and in parallel $S_P$, we satisfy the input voltage and current requirements of the TE unit. Next, we briefly describe the three models shown in Fig. 11.

5.2.1 Model of TE Unit. The TE unit is a collection of TE coolers. The basic element of a TE cooler is called a thermocouple [37]. The amount of heat absorbed by one TE cooler, $Q_{pc}$, that contains $n$ thermocouples is given as follows [38]:

$$Q_{pc} = n[S_i T_c - \frac{1}{2} \sum_{i=1}^{n} R_i + (Q \cdot v_i) R_{sh} - T_o]$$

where

$$v_i = S_i (T_o + Q + v_i R_{sh} - T_c) + i R$$

In Eqs. (36) and (37), $S$, $K$, and $R$ are the Seebeck coefficient, the thermal conductance, and the electrical resistance of the thermocouple, respectively, $T_i$ is the temperature of the cold junctions of the TE cooler, $v_i$ and $i$ are the input voltage and the input current of one TE cooler, respectively, and $R_{sh}$ is the thermal resistance of the heat sink attached to each TE cooler. To determine the heat absorbed by the entire TE unit, we multiply Eq. (36) with the product of $H_S$ and $H_P$. We use CP-1.0-07-06 type of off-the-shelf TE coolers in this example. The properties of this type of TE coolers is available in the manufacturer’s catalog [39]. The input voltage $V_{TE}$ and the input current $I_{TE}$ required to operate the TE unit are determined as

$$V_{TE} = H_S V_t, \quad I_{TE} = H_P I_t$$

5.2.2 Model of PV Unit. For the PV unit that contains $S_S$ solar cells in series and $S_P$ solar cells in parallel, the governing equation is given as [40]

$$\frac{I_{PV}}{S_P} = \frac{I_{sh}}{S_P} - \left[ \exp \left( \frac{V_{PV}}{A k_B T_0} \right) \frac{V_{PV}}{S_S} + \frac{I_{PV} R_{sh}}{S_P} \right] - 1$$

where $V_{PV}$ and $I_{PV}$ are the voltage and the current generated by the PV unit, respectively, $I_{sh}$, $I_0$, $R_s$, $R_{sh}$, and $A$ are the photocurrent, the reverse saturated current, the series resistance, the shunt resistance, and the ideality factor of the solar cell, respectively, $q$ is the electron charge, $k_B$ is Boltzmann’s constant, and $T_0$ is the solar cell temperature. We note that in this paper, we use the solar cell data provided in Ref. [40]. In Eq. (39), the photocurrent and the solar cell temperature depend on the weather conditions. To determine the photocurrent and the solar cell temperature, we use the following models given in Ref. [40].

$$I_{ph} = -0.00607 + 0.0000671 E_o + 0.984 I_{sc}$$
The other two quantities that depend on the short circuit current $E_e$ where $E_e$ is the solar radiation per unit area and $I_0$ is the short circuit current $E_e$. where $E_e$ is the solar radiation per unit area and $I_0$ is the short circuit current $E_e$ are given in the third and fourth rows of Table 2.

5.2.3 Model of House. We assume that all the walls of the house considered in this example contain the ABE system, and the conduction through the ABE walls is the only mode of heat transfer (no windows or doors in the house). The amount of heat conducted through the ABE wall, or the cooling load, is determined by Fourier’s law as

$$ Q_{\text{load}} = k_{\text{ABE}} A_{\text{ABE}} (T_i - T_a) / T_{\text{ABE}} $$

where $k_{\text{ABE}}$, $A_{\text{ABE}}$, and $T_{\text{ABE}}$ are the thermal conductivity, the surface area, and the thickness of the ABE wall, respectively, and $T_i$ is the temperature inside the house.

5.3 Changing Weather Conditions. We assume that between spring and summer, the outside temperature $T_a$ changes from 21°C to 42°C, and the solar radiation $E_r$ changes from 300 W/m² to 900 W/m². We select the seven operating conditions ($m=7$) shown in the first two rows of Table 2. Based on these seven operating conditions, we design the ABE system that maintains a constant indoor temperature of $T_i = 20°C$. As mentioned in the previous subsection, the reverse saturation current $I_0$ and the short circuit current $I_{sc}$ of the solar cell depend on the solar radiation $E_r$. These quantities are given in the last two rows of Table 2.

According to Eq. (42), the cooling load changes with the outside temperature. To absorb the changing cooling load, the input power required to operate the TE unit also changes. In order to design a power efficient ABE system, we minimize the TE unit input power at a set of operating conditions. Preliminary analysis suggests that the design configuration of the ABE system considered in this paper needs to change with the outside temperature, for it to operate with the minimum input power. For the ABE system example, the change in the configuration implies changing the number of TE coolers and solar cells placed in series and parallel ($H_S, H_P, S_S,$ and $S_P$). Accordingly, we select these quantities as design variables. To change the design variables with the outside temperature, we need a controlling mechanism. Addition of the controlling mechanism contributes to the penalty that we measure in terms of the change in the design variables.

Based on the above discussion, we formulate the ABE system optimization problem such that it simultaneously minimizes the TE input power and the penalty associated with the changing design variables. We include two constraint variables as follows. At a set of operating conditions, (1) the total heat absorbed by the TE unit should match the cooling load and (2) the voltage and the current generated by the PV unit should match those required by the TE unit. Finally, we set the upper and lower limits on the design variables as $1$ and $50$, respectively. We use the proposed SIO methodology to design this ABE system.

5.4 Design of ABE System Based on SIO Methodology

5.4.1 Benchmark for ABE System. We develop the fully adaptive ABE system as a benchmark. To do so, we solve the following optimization problem at the seven operating conditions given in Table 2. For the benchmark ABE system, the TE unit input power is minimized alone as

$$ \min_{x^i} \mu^i = V_{\text{TE}} I_{\text{TE}} $$

subject to

$$ (H_S H_P) Q_{\text{PE}} = Q_{\text{load}} \quad \forall i $$

$$ V_{\text{TE}} = V_{\text{PV}} \quad \forall i $$

$$ P_{\text{TE}} = P_{\text{PV}} \quad \forall i $$

$$ 1 \leq S_S, S_P, H_S, H_P \leq 50 \quad \forall i $$

In the above formulation, the objective function $\mu^i$ represents the input power for the TE unit, and $i$ changes from $1$ to $7$. After each optimization, we record the minimum input power, $\mu_1$. The minimum TE unit input power values for the fully adaptive ABE system are given in Table 3 under the benchmark ABE system.

### Table 2 Operating conditions selected for the ABE system

<table>
<thead>
<tr>
<th>$T_a$, °C</th>
<th>$E_r$, W/m²</th>
<th>$I_0$, 10⁻⁷ A</th>
<th>$I_{sc}$, A</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>24.5</td>
<td>0.901</td>
<td>0.72</td>
</tr>
<tr>
<td>28</td>
<td>31.5</td>
<td>1.20</td>
<td>0.92</td>
</tr>
<tr>
<td>35</td>
<td>38.5</td>
<td>1.72</td>
<td>1.24</td>
</tr>
<tr>
<td>42</td>
<td>47</td>
<td>2.34</td>
<td>1.49</td>
</tr>
</tbody>
</table>

$T_{\text{PV}} = 30.006 + 0.0175(E_r - 300) + 1.14(T_a - 273)$

5.4.2 Design of ABE System Using SIO Methodology

### Table 3 Results obtained at various steps of the proposed SIO methodology

<table>
<thead>
<tr>
<th>$p^1$</th>
<th>$p^2$</th>
<th>$p^3$</th>
<th>$p^4$</th>
<th>$p^5$</th>
<th>$p^6$</th>
<th>$p^7$</th>
<th>$\Delta x^i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark ABE system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mu_{SIO}^i$</td>
<td>0.349</td>
<td>7.337</td>
<td>24.082</td>
<td>51.751</td>
<td>91.702</td>
<td>145.525</td>
<td>215.088</td>
</tr>
<tr>
<td>$S_S^*$</td>
<td>5.5663</td>
<td>5.5708</td>
<td>5.5700</td>
<td>5.5801</td>
<td>5.6069</td>
<td>5.5243</td>
<td>5.6094</td>
</tr>
<tr>
<td>$S_P^*$</td>
<td>1.0525</td>
<td>3.8082</td>
<td>5.1744</td>
<td>6.3503</td>
<td>7.2401</td>
<td>8.0717</td>
<td>8.8676</td>
</tr>
<tr>
<td>$H_S^*$</td>
<td>38.0898</td>
<td>38.4140</td>
<td>39.2330</td>
<td>39.3636</td>
<td>40.0363</td>
<td>40.6821</td>
<td>41.3531</td>
</tr>
<tr>
<td>$\mu_{SIO}^i$</td>
<td>5.6632</td>
<td>5.6632</td>
<td>5.6632</td>
<td>5.6632</td>
<td>5.6632</td>
<td>5.6632</td>
<td>5.6632</td>
</tr>
<tr>
<td>$S_P^*$</td>
<td>40.2463</td>
<td>40.2463</td>
<td>40.2463</td>
<td>40.2463</td>
<td>40.2463</td>
<td>40.2463</td>
<td>40.2463</td>
</tr>
<tr>
<td>$H_P^*$</td>
<td>0.3495</td>
<td>7.3374</td>
<td>24.0821</td>
<td>51.7510</td>
<td>91.7024</td>
<td>145.5253</td>
<td>215.2635</td>
</tr>
</tbody>
</table>

101401-10 / Vol. 130, OCTOBER 2008 Transactions of the ASME
5.4.2 Bi-Objective Formulation for Adaptive ABE System. In the bi-objective problem of the adaptive ABE system, we construct a weighted sum AOF to combine the performance and penalty objective functions, \( f_{\text{per}} \) and \( f_{\text{pen}} \) respectively. To avoid potential scaling mismatch issues between the two objectives, we normalize them according to the normalization scheme proposed by Messac et al. [41]. The optimization problem is formulated as:

\[
\min_{x_1, \ldots, x_7} w_1 f_{\text{per}} + w_2 f_{\text{pen}}
\]

subject to

\[
(HS, HP)Q_i = Q_{\text{load}}, \quad \forall i = 1, \ldots, 7
\]

\[
V_{\text{TE}} = V_{\text{PV}}, \quad \forall i = 1, \ldots, 7
\]

\[
I_{\text{TE}} = I_{\text{PV}}, \quad \forall i = 1, \ldots, 7
\]

\[
\mu_i - \mu_{\text{ref}} \leq 0.1 \mu_{\text{ref}} \quad \forall i = 1, \ldots, 7
\]

\[
1 \leq S_i S_{\text{ref}} H_i H_{\text{ref}} \leq 50, \quad \forall i = 1, \ldots, 7
\]

where \( w_1 \) and \( w_2 \) are the weights for the performance and penalty objectives, respectively. These weights represent the designer’s preferences regarding the two objectives. We use equal weights \( (w_1 = w_2 = 0.5) \) for the two objectives. This nominally implies that we give equal importance to both of them.

In the above problem, we determine \( f_{\text{per}} \) and \( f_{\text{pen}} \) using Eqs. (9) and (10), respectively. In the PLCs, denoted in Eq. (52), we do not allow the performance of the adaptive ABE system to deviate more than 10% from that of the benchmark ABE system. Table 3 shows the optimal design variable values obtained from the above problem under the title ABE system without SIO methodology. We determine the changes in the optimal design variables using Eq. (17), and these are shown in the last column of Table 3. We note that the simple weighted sum AOF used in the current example may be inadequate in some situations. A detailed discussion regarding limitation of weighted sum AOF and possible alternatives is presented in Ref. [30].

5.4.3 Implementing VSMF to Design Adaptive ABE System. The last step in the SIO methodology involves implementing the VSMF in the bi-objective optimization problem. We follow the procedure described in the earlier Sec. 4.6. This procedure essentially involves repeatedly solving the optimization problem given by Eqs. (48)–(53) with two modifications. First, the penalty is redefined as per in Eq. (19). Second, the design variables are mapped using the VSMF. In each repetition of the optimization problem, the value of \( \alpha \) is lowered. We start with \( \alpha = 0.9 \) and stop at \( \alpha = 0.2 \). Table 3 shows the optimal design variable values obtained at the end of this repetitive procedure under the heading ABE System with SIO Methodology. Next, let us evaluate these results in more detail.

5.5 Results and Discussion. From Table 3, we observe that the SIO methodology has converged to a design in which the optimum values of the two-design variables \( S_i \) and \( H_i \) do not change with operating conditions. Thus, the SIO methodology has converged to an ABE system with \( S_i \) and \( H_i \) as the fixed design variables and \( S_{\text{ref}} \) and \( H_{\text{ref}} \) as the adaptive design variables. The SIO methodology also provided the magnitudes of the respective design variables. The segregation of the fixed from the adaptive design variables and the optimization of the ABE system is performed simultaneously. Throughout the implementation of the SIO methodology, the designer is not required to arbitrarily select the fixed and adaptive design variables, nor to transform the optimization problem into an approximated one. Also, we successfully demonstrated that the gradient based optimization methods can be used for solving the original problem, which is combinatorial in nature. The simple yet powerful VSMF is at the center of these new possibilities. To further explore the effectiveness of the SIO methodology, we discuss the effect of the \( \alpha \) iterations and the validation of the results.

5.5.1 Effect of \( \alpha \) Iterations. As shown in Fig. 6(a), with every repetition of the optimization problem, the fixed design variables move closer to segment \( O_{\text{opt}} \). This implies that for the fixed design variables, the magnitude of the normalized mapped-change, \( \Delta_{\text{mapp}} \), converges to zero as the value of \( \alpha \) is lowered. Table 4 provides the magnitudes of the normalized mapped-change for all the design variables, as \( \alpha \) is lowered. It is observed that all the fixed and adaptive design variables are segregated in the very first repetition of the optimization problem. We have observed a similar trend in the other engineering problems as well. That is, the SIO methodology is able to segregate the design variables within three to four repetitions of the optimization problem. However, this observation is based on numerical experimentation to date. Theoretically, it is possible to encounter a situation where not all the design variables are segregated at \( \alpha = 0.2 \) or lower. In such cases, because of the numerical conditioning issues involved as \( \alpha \) tends to zero, the designer may be required to use a nongradient based optimizer.

5.5.2 Comparison of the Numerical Results. In this section, we compare the ABE system design obtained from the SIO methodology with that obtained from a prominent existing method; specifically, that proposed by Fellini et al. [24]. In Table 3, the design given under heading “ABE system without SIO methodology” is obtained using the formulation similar to that proposed by Fellini et al. (see the formulation shown in Eqs. (48)–(53)). Clearly, for none of the design variables, the change has become zero (see the last column in Table 3). Hence, this method has not segregated the fixed from the adaptive design variables. Whereas for the design obtained from the SIO methodology, the fixed from the adaptive design variables are clearly segregated (see the design given under the title ABE system with SIO methodology in Table 3).

Next, we generate 16 (all) possible alternate ABE system designs by manually changing the set of fixed and adaptive design variables, see Table 5. In Table 5, for any design (row), the letter “f” indicates a fixed design variable and “a” an adaptive design variable. For each design, we solve the optimization problem proposed by Fellini et al. (Eqs. (48)–(53)), with one change—the magnitudes of the fixed design variables are not allowed to change with operating conditions. Table 5 shows the optimal values of the performance objective \( f_{\text{per}} \), the penalty objective \( f_{\text{pen}} \), and the AOF for the 16 designs.

From Table 5, we observe that the designs that have \( S_i \) as the fixed design variable are not feasible. Comparing the remaining designs, we observe that Design 16 results in the lowest AOF value. We note that this is the same design that we discussed at the beginning of this subsection. By comparing Design 16 with the highlighted Design 10, we observe that there is not a significant difference between their AOF values. However, Design 10 has two fixed design variables and two adaptive design variables. Since the AOF values of the Designs 16 and 10 are almost the

```
<table>
<thead>
<tr>
<th>Iteration</th>
<th>( \alpha )</th>
<th>( \Delta S_i )</th>
<th>( \Delta S_{\text{ref}} )</th>
<th>( \Delta H_i )</th>
<th>( \Delta H_{\text{ref}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9</td>
<td>0.0000</td>
<td>0.5081</td>
<td>0.0000</td>
<td>0.5027</td>
</tr>
<tr>
<td>2</td>
<td>0.8</td>
<td>0.00006</td>
<td>0.5050</td>
<td>0.0017</td>
<td>0.4663</td>
</tr>
<tr>
<td>3</td>
<td>0.7</td>
<td>0.00000</td>
<td>0.5078</td>
<td>0.0000</td>
<td>0.4955</td>
</tr>
<tr>
<td>4</td>
<td>0.6</td>
<td>0.0002</td>
<td>0.5082</td>
<td>0.0000</td>
<td>0.4928</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>0.00000</td>
<td>0.5181</td>
<td>0.0115</td>
<td>0.5976</td>
</tr>
<tr>
<td>6</td>
<td>0.4</td>
<td>0.00000</td>
<td>0.5196</td>
<td>0.0140</td>
<td>0.6065</td>
</tr>
<tr>
<td>7</td>
<td>0.3</td>
<td>0.00000</td>
<td>0.5180</td>
<td>0.0073</td>
<td>0.5069</td>
</tr>
<tr>
<td>8</td>
<td>0.2</td>
<td>0.00000</td>
<td>0.5129</td>
<td>0.0088</td>
<td>0.5973</td>
</tr>
</tbody>
</table>
```


same, the latter has an advantage because it is not fully adaptive. It is interesting to observe that Design 10 has the same fixed and adaptive design variables that are given by the SIO methodology (δS and HS as fixed and SP and HP as adaptive). We also observed that for Design 10, the fixed and adaptive design variables take on the same optimal values that we obtained from the SIO methodology (see Table 3). This observation demonstrates the effectiveness of integrating selection and optimization processes in the SIO methodology. Because of this feature, the SIO methodology is able to uncover the ABE system design that is almost equivalent to the fully adaptive design, with an advantage of not having to make all the design variables adaptive.

6 Concluding Remarks

In this paper, we presented the SIO methodology that avoids a significant source of suboptimality from the adaptive system optimization procedure. The SIO methodology integrates two critical aspects of the adaptive system optimization process: (1) the selection of the fixed and adaptive design variables and (2) the optimization of the adaptive system. A bi-objective optimization problem is formulated for adaptive systems that attempts to maximize the overall performance of the adaptive system and to minimize the penalty associated with the changing design variables. The VSMF is introduced in the bi-objective problem. The VSMF progressively approximates the inherent discreteness present in the selection of the fixed design variables. The bi-objective problem, that includes the VSMF, is repeatedly solved by lowering the key VSMF parameter, α, in every repetition. As the repetitive procedure progresses, the VSMF segregates the fixed from the adaptive design variables, by making the change in these design variables converge to zero.

The effectiveness of the proposed SIO methodology was demonstrated by designing a new air-conditioning system called ABE system to operate under changing weather conditions. The analytical model of the ABE system was developed by combining those of its components. The SIO methodology required only one repetition of the VSMF-integrated bi-objective optimization problem to segregate the fixed from the adaptive design variables. Similar trends were observed for other numerical applications of the SIO methodology. Throughout the adaptive system optimization process, the designer was only required to introduce a simple yet powerful VSMF. Furthermore, no arbitrary rules were invoked to segregate the fixed from the adaptive design variables nor to transform the original problem into some approximate on—as is typically required by other approaches. These promising results suggest that the SIO methodology could prove to be a highly effective approach for optimizing complex adaptive systems.

Table 5 Evaluation of alternate ABE designs (N/F = not feasible)

<table>
<thead>
<tr>
<th>Design</th>
<th>S₁</th>
<th>S₂</th>
<th>H₁</th>
<th>H₂</th>
<th>( f_{\text{pen}} )</th>
<th>( f_{\text{act}} )</th>
<th>AOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>—</td>
<td>—</td>
<td>N/F</td>
</tr>
<tr>
<td>2</td>
<td>a</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>—</td>
<td>—</td>
<td>N/F</td>
</tr>
<tr>
<td>3</td>
<td>f</td>
<td>a</td>
<td>f</td>
<td>f</td>
<td>0.0196</td>
<td>255.812</td>
<td>0.05618</td>
</tr>
<tr>
<td>4</td>
<td>f</td>
<td>f</td>
<td>a</td>
<td>f</td>
<td>—</td>
<td>—</td>
<td>N/F</td>
</tr>
<tr>
<td>5</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>a</td>
<td>—</td>
<td>—</td>
<td>N/F</td>
</tr>
<tr>
<td>6</td>
<td>a</td>
<td>a</td>
<td>f</td>
<td>f</td>
<td>0.0196</td>
<td>260.436</td>
<td>0.05633</td>
</tr>
<tr>
<td>7</td>
<td>a</td>
<td>f</td>
<td>f</td>
<td>a</td>
<td>—</td>
<td>—</td>
<td>N/F</td>
</tr>
<tr>
<td>8</td>
<td>a</td>
<td>f</td>
<td>f</td>
<td>a</td>
<td>—</td>
<td>—</td>
<td>N/F</td>
</tr>
<tr>
<td>9</td>
<td>f</td>
<td>a</td>
<td>a</td>
<td>f</td>
<td>2.539 \times 10^{-5}</td>
<td>1250.5</td>
<td>0.03294</td>
</tr>
<tr>
<td>10</td>
<td>f</td>
<td>a</td>
<td>f</td>
<td>a</td>
<td>4.401 \times 10^{-4}</td>
<td>504.716</td>
<td>0.00997</td>
</tr>
<tr>
<td>11</td>
<td>f</td>
<td>f</td>
<td>a</td>
<td>a</td>
<td>—</td>
<td>—</td>
<td>N/F</td>
</tr>
<tr>
<td>12</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>4.107 \times 10^{-5}</td>
<td>1187.4</td>
<td>0.0309</td>
</tr>
<tr>
<td>13</td>
<td>a</td>
<td>a</td>
<td>f</td>
<td>a</td>
<td>3.693 \times 10^{-4}</td>
<td>551.643</td>
<td>0.0112</td>
</tr>
<tr>
<td>14</td>
<td>a</td>
<td>f</td>
<td>a</td>
<td>a</td>
<td>4.643 \times 10^{-4}</td>
<td>588.7699</td>
<td>0.0119</td>
</tr>
<tr>
<td>15</td>
<td>f</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>1.797 \times 10^{-4}</td>
<td>463.6040</td>
<td>0.05633</td>
</tr>
<tr>
<td>16</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>1.948 \times 10^{-4}</td>
<td>463.6040</td>
<td>0.00793</td>
</tr>
</tbody>
</table>

Acknowledgment

Support from the National Science Foundation, Award No. CMS-033568 and the US Department for Housing and Urban Development is much appreciated.

Nomenclature

- \( f_{\text{pen}} \): penalty objective function
- \( f_{\text{act}} \): performance objective function
- \( a \): inequality constraints
- \( g \): used to denote generic operating condition
- \( k \): used to denote generic design variable
- \( k_L \): performance loss factor
- \( m \): number of operating conditions
- \( n \): number of design variables
- \( s \): vector used to store design variable values
- \( x \): design variable vector
- \( \alpha \): design variable
- \( \bar{x} \): mapped value of the design variable
- \( P \): operating condition

Greek Symbols

- \( \alpha \): mapping function parameter
- \( \mu \): objective function
- \( \Delta x_k \): change in \( k^{th} \) design variable
- \( \Delta x_k \): normalized value of change in \( k^{th} \) design variable
- \( \Delta \alpha \): change in \( \alpha \)

Superscripts

- \( i \): value at \( i^{th} \) operating condition
- \( k \): \( k^{th} \) design variable
- \( * \): optimal value for a particular variable

Subscripts

- \( \text{act} \): actual value
- \( \text{fa} \): corresponding to the fully adaptive system
- \( \text{map} \): mapped value
- \( \text{max} \): maximum value for a particular variable
- \( \text{min} \): minimum value for a particular variable
- \( \text{th} \): threshold value

References
