Douglas Allaire[1]
e-mail: dallaire@mit.edu

Qinxian He

John Deyst

Karen Willcox

Aerospace Computational Design Laboratory,
Department of Aeronautics and Astronautics,
Massachusetts Institute of Technology,
Cambridge, MA 02139

An Information-Theoretic Metric of System Complexity With Application to Engineering System Design

System complexity is considered a key driver of the inability of current system design practices to at times not recognize performance, cost, and schedule risks as they emerge. We present here a definition of system complexity and a quantitative metric for measuring that complexity based on information theory. We also derive sensitivity indices that indicate the fraction of complexity that can be reduced if more about certain factors of a system can become known. This information can be used as part of a resource allocation procedure aimed at reducing system complexity. Our methods incorporate Gaussian process emulators of expensive computer simulation models and account for both model inadequacy and code uncertainty. We demonstrate our methodology on a candidate design of an infantry fighting vehicle. [DOI: 10.1115/1.4007587]

1 Introduction

Over the years, engineering systems have become increasingly complex, with astronomical growth in the number of components and their interactions. With this rise in complexity comes a host of new challenges, such as the adequacy of mathematical models to predict system behavior, the expense and time to conduct experimentation and testing, and the management of large, globally distributed design teams. These obstacles contribute uncertainties to system design, which can have serious, often distrastrous, implications for program outcome. A notable example is the Hubble Space Telescope which, when first launched, failed its resolution requirement by an order of magnitude. A Shuttle repair mission, costing billions of additional dollars, was required to remedy the problem [1]. The V-22 Osprey tilt-rotor aircraft is another example: over the course of its 25-year development cycle, the program was fraught with safety, reliability, and affordability challenges, resulting in numerous flight test crashes with concomitant loss of crew and passenger lives [2]. More recently, the Boeing 787 Dreamliner transport aircraft program has experienced a number of major prototype subsystem test failures, causing budget overruns of billions of dollars and service introduction delays of about 3 yr. One major source of blame for Boeing's setbacks is its aggressive strategy to outsource component design and assembly, which created heavy program management burdens and led to unforeseen challenges during vehicle integration [3].

In these cases and numerous others, the design program was unaware of the mounting risks in the system, and was surprised by one or more unfortunate outcomes. Although these examples are extreme, they are suggestive that current system design practices are unable to recognize performance, cost, and schedule risks as they emerge. Such unanticipated or emergent behavior is often attributed to the complexity of the underlying system [4]. This has led to a desire to measure system complexity in a manner that will enable design trades and improve parameterization of cost and schedule. Thus, our objectives are to quantitatively define system complexity in terms of system quantities of interest and to formulate a complexity-based sensitivity analysis. The resulting methodology identifies the key contributors to system complexity and provides quantitative guidance for resource allocation decisions aimed at reducing system complexity.

We define system complexity as the potential for a system to exhibit unexpected behavior in the quantities of interest. A background discussion on complexity metrics, uncertainty sources in complex systems, and related work presented in Sec. 2. We measure this complexity as the exponential information entropy of the probability distribution of the quantities of interest associated with a given system. Exponential entropy has been established by Ref. [5] as a rigorous measure of the extent of a probability distribution and is described in more detail in Sec. 3, which also includes the development of our sensitivity analysis procedure, which may be used to direct a design refinement process [6]. We apply our methodology to a design of an infantry fighting vehicle (IFV). The quantity of interest for the application is the range of the vehicle. The application is described in more detail in Sec. 4. A demonstration of the use of our methodology is presented in Sec. 5, where two IFV options are considered and general conclusions are drawn in Sec. 6.

2 Background

Complexity in system design is an elusive concept for which many definitions have been proposed, though none formally adopted. Early work in the field of *complexity science* by Warren Weaver posited complexity as the nebulous middle ground between order and chaos, a region in which problems require "dealing simultaneously with a sizeable number of factors which are interrelated in an organic whole" [7]. Another interpretation of this idea considers a set of "phase transitions" during which the fundamental features of a system undergo drastic changes [8]. As an illustrative example, consider the phase transitions of water [9]. On one end of the spectrum, water is frozen into a simple lattice of molecules whose structure and behavior are straightforward to understand. At the other extreme, water in gaseous form consists of millions of molecules vibrating at random, and the study of such a system requires methods of statistical mechanics or probability theory [10]. In between the two lies the complex liquid state, wherein water molecules behave in a manner neither orderly nor chaotic, but at once enigmatic and mesmerizing, which has captured the imagination of fluid dynamicists past and present.

[1]Corresponding author.

Contributed by Design Automation Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received February 16, 2012; final manuscript received July 26, 2012; published online September 21, 2012. Assoc. Editor: Bernard Yannou.

Though the above example makes the idea of complexity relatable to a large audience, the debate over its definition still persists. However, many researchers agree that there are several properties that complex systems tend to share [11–15]: (1) they consist of many parts; (2) there are many interactions among the parts; (3) the whole exceeds the sum of the parts, that is, the parts in combination produce synergistic effects that are not easily predicted and may often be novel, unexpected, even surprising; and (4) they are difficult to model and to understand.

In addition to qualitative descriptions of complexity, there have also been many attempts to explain complexity using quantitative measures. These definitions can be classified into two general categories, structure-based metrics and process-based metrics. Structure-based metrics quantify the complexity associated with the physical representation of a system [16]. They typically involve counting strategies: in software engineering, the source lines of code (SLOC) can be used to describe a computer program [17]; in mechanical design, analogous measures include the number of parts [18], functions [19], or core technologies [20] embodied in a product. Though appealing, these counting metrics may be susceptible to different interpretations of what constitutes a distinct component—depending on the level of abstraction, a component may be as high-level as an entire subsystem, or as basic as the nuts and bolts holding it together. More sophisticated structure-based metrics also attempt to address the issue of component interactions through an analysis of the topology and connectivity of the system [21,22]. For example, McCabe proposed the idea of cyclomatic complexity in software engineering, which uses graph theory to determine the number of control paths through a module [23]. Numerous others have also recommended approaches to estimate system complexity by characterizing the number, extent, and nature of component interactions, which govern the interconnectedness and solvability of the system [15,24–26]. Overall, structure-based complexity metrics are usually easy to understand and to implement, but they may not be meaningful except in the later stages of design, after most design decisions have been made, and the system is well-characterized [27].

A second class of complexity metrics quantifies system uncertainty in terms of processes required to realize the system. One metric in this category is algorithmic complexity, or Kolmogorov complexity, which measures the compactness of an algorithm needed to specify a particular message [28–30]. Similar definitions include the number of basic operations required to solve a problem (computational complexity), or the amount of effort necessary to design, modify, manufacture, or assemble a product [16,27,31,32]. Another possible interpretation of complexity is related to the information content of a system. The concept of *information entropy* was originally proposed by Claude E. Shannon to study lossless compression schemes for communication systems [33]. Information entropy, or Shannon entropy, measures the uncertainty associated with a random variable. It also has an intuitive and appealing analogy to entropy in the thermodynamic sense, as a measure of a system's tendency toward disorder [34]. In this work, we propose a complexity metric based on exponential information entropy, which is described in Sec. 3. It is important to note here that there are many different metrics of complexity and each can be useful in different ways and thus, all are important. We intend our complexity metric to be used in simulation-based design activities where limited information is known about quantities of interest relevant to the design of a complex system. This is a context that has also been considered by Refs. [35] and [36] in terms of life cycle design and design optimization, respectively. Given our context, our metric is based on the information content in our estimates of quantities of interest. Thus, our metric reflects a correspondence between uncertainty in a system and the complexity of the system, as consistent with our complexity definition stated in Sec. 1. This correspondence does not exist for many of the other complexity metrics noted, particularly the structure-based metrics.

3 Methodology

In this section, we define our complexity metric and develop a quantitative measure of it. We then develop a sensitivity analysis procedure designed to identify the key contributors to system complexity in an effort to identify how to best allocate resources for complexity reduction.

3.1 Complexity Metric. We define complexity as the potential of a system to exhibit unexpected behavior in the quantities of interest, which are the quantities characterizing the performance, cost, schedule, and other relevant attributes of the system. Thus, we wish to characterize the amount of knowledge we have with respect to our quantities of interest. To measure this amount of knowledge, or level of information, we define a metric of complexity based on exponential information entropy. For a discrete random variable Y with probability mass function $p(y)$, the information entropy of Y is defined as

$$H(Y) = -\sum_i p(y_i)\log p(y_i) \tag{1}$$

where $y_1, y_2, \ldots$ are values of y such that $p(y) \neq 0$. For a continuous random variable X with probability density function $f_X(x)$, the differential information entropy of X is defined as

$$h(X) = -\int_{-\infty}^{\infty} f_X(x)\log f_X(x)dx \tag{2}$$

where the integrand is taken to be zero when $f_X(x) = 0$. Our work here focuses on continuous random variables. For both the discrete and continuous cases, the base of the logarithm is chosen by the user. We will deal exclusively in this work with the natural logarithm. Thus, our quantitative metric of system complexity is given as

$$C(Q) = \exp\left\{-\int_{-\infty}^{\infty} f_Q(q)\ln f_Q(q)dq\right\} \tag{3}$$

where Q is the random variable associated with a quantity of interest of a given system.

Exponential entropy was first introduced in Ref. [5] as a measure of the extent of a probability distribution. Following Ref. [5], we can express the range of a random variable as

$$\mathcal{R}(Q) = \int_{\mathbb{Q}} dq \tag{4}$$

$$\mathcal{R}(Q) = \int_{\mathbb{Q}} \frac{1}{f_Q(q)} f_Q(q)dq \tag{5}$$

where Q is a continuous random variable, $f_Q(q)$ its associated probability density function, $\mathcal{R}(Q)$ is the range of Q, and $\mathbb{Q} = \{q \in Q : f_Q(q) > 0\}$. We can also express the generalized mean of order t of $f_Q(q)$ over $\mathbb{Q}$ as

$$M_t(Q) = \left\{\int_{\mathbb{Q}} \left[\frac{1}{f_Q(q)}\right]^t f_Q(q)dq\right\}^{1/t} \quad (t \neq 0) \tag{6}$$

$$M_0(Q) = \exp\left\{\int_{\mathbb{Q}} \ln\left[\frac{1}{f_Q(q)}\right] f_Q(q)dq\right\} \quad (t = 0) \tag{7}$$

It can be seen that the range as defined in Eq. (5) is the generalized mean of order 1. Furthermore, when $t = 0$, $M_0(Q)$ is the geometric mean of $f_Q(q)$ over $\mathbb{Q}$ [37]. For a uniform random variable $Q \sim \mathcal{U}[a, b]$, $M_t(Q) = \mathcal{R}(Q) = b - a$ for all t. For arbitrary orders s and t, where $s < t$, the relationship between $M_s(Q)$ and $M_t(Q)$ can be described by the generalized mean inequality [38]

$$M_s(Q) \leq M_t(Q) \tag{8}$$

where equality holds if and only if Q is a uniform random variable. Setting $s=0$ and $t=1$, Eq. (8) can be used to obtain a general relationship between $M_0(Q)$ and $\mathcal{R}(Q)$

$$M_0(Q) \leq M_1(Q) \tag{9}$$

$$M_0(Q) \leq \mathcal{R}(Q) \tag{10}$$

The quantity $M_0(Q)$ is referred to as the *intrinsic extent* of Q, which as shown in Eq. (10) is always less than or equal to the range of Q. This intrinsic extent may be rewritten as

$$M_0(Q) = \exp\left\{-\int_{\mathbb{Q}} f_Q(q)\ln f_Q(q)dq\right\} \tag{11}$$

$$= \exp\{h(Q)\} \tag{12}$$

Thus, the intrinsic extent of a distribution is the exponential of the information entropy of the distribution, which is termed the *exponential entropy*. We propose that the exponential entropy of the quantities of interest of a given system is a quantitative measure of the complexity of the system, which we write as

$$C(Q) = \exp\{h(Q)\} \tag{13}$$

The exponential entropy of a uniform random variable can be interpreted as the length of the support of the random variable (and area, volume, and hypervolume for 2, 3, and n-dimensional jointly distributed uniform random variables). To this end, the exponential entropy of any arbitrarily distributed random variable can be related to the length of the support of an information-entropy-equivalent uniform distribution.

3.2 Complexity Estimation. Defining complexity in terms of exponential entropy implies that we are concerned with uncertainty associated with quantities of interest. In modeling a potential system, which is typically done with numerical simulation models, there are many potential sources of uncertainty that can impact quantities of interest, and thus system complexity. Among these are parametric uncertainty, parametric variability, code uncertainty (CU), observation error, and model inadequacy. Following Ref. [39], parametric uncertainty refers to uncertain inputs or parameters of a model, parametric variability refers to uncontrolled or unspecified conditions in inputs or parameters, code uncertainty refers to the uncertainty associated with not knowing the output of a computer model given any particular configuration until the code is run, observation error is uncertainty associated with actual observations and measurements, and model inadequacy relates to the fact that no model is perfect. For the application considered here, we do not incorporate experimental data; therefore, our focus is on parametric variability, parametric uncertainty, code uncertainty, and model inadequacy.

A simulation model, or simulator, is a function $g(\cdot)$ that maps inputs $\mathbf{x}$ into an output $q=g(\mathbf{x})$. In our work, we incorporate the presence of simulator model inadequacy by adding noise to simulator output. Thus, the true value of a quantity of interest that has been estimated by a simulator is in the form

$$q = g(\mathbf{x}) + \varepsilon(\mathbf{x}) \tag{14}$$

where $\varepsilon(\mathbf{x})$ is additive noise that is permitted to vary throughout the input space. In the demonstrations provided in Secs. 4 and 5, we notionally account for model inadequacy by assuming normally distributed noise. The purpose of this is to ensure that we are taking into account some form of model inadequacy in the complexity estimation process and the sensitivity analysis methodology. However, our approach does not require that the model inadequacy term be normally distributed. The need to quantify model inadequacy in simulation models was originally addressed in Ref. [39]. More general approaches to the quantification of model inadequacy that incorporate both data and expert opinion are important topics of future work.

When analyzing quantities of interest with computer models, it is often necessary to approximate the input/output relationships of expensive simulators using less expensive surrogate models. For this, we employ the well-known technique of Gaussian process regression [39–43]. Gaussian process regression is based on emulating a simulator with a stochastic process model. Emulating with a stochastic process ensures there is a complete statistical approximation of the simulator, which enables the code uncertainty associated with the use of the emulator in place of the simulator to be quantified. This is essential for situations where the code uncertainty of the emulator is a key driver of complexity.

When using an emulator, the true value of a quantity of interest is in the form

$$q = \mathcal{G}(\mathbf{x}) + \varepsilon(\mathbf{x}) \tag{15}$$

where $\mathcal{G}(\mathbf{x}) \sim \mathcal{GP}(\mathrm{m}(\mathbf{x}), k(\mathbf{x},\mathbf{x}'))$, $\mathrm{m}(\mathbf{x})$ is the mean function of the Gaussian process $\mathcal{GP}(\cdot,\cdot)$, and $k(\mathbf{x},\mathbf{x}')$ is the covariance kernel of the Gaussian process. A Gaussian process emulator is built with a set of training runs of the simulation model, $\mathcal{D}=\{(\mathbf{x}_i,q_i)|i=1,\ldots,n\}$. This training set is treated as data that are used to estimate the simulation model. An example of one-dimensional Gaussian process regression is shown graphically in Fig. 1, where three data points from a simulator have been used as training points for the emulator. The underlying simulator is the function $q(x)=x+3\sin(x/2)$. The emulator itself is a stochastic process, which is represented on the figure as a mean function (dashed line) and plus and minus two standard deviation bounds (grayed area). The grayed area is a representation of the code uncertainty associated with the use of this emulator in place of the underlying simulator. The fitting of such an emulator is a machine learning task that involves the estimation of several hyperparameters. Details on how this may be accomplished can be found in Ref. [42].

To estimate complexity with respect to a quantity of interest, we require an estimate of the probability density function of the quantity of interest, $f_Q(q)$. For situations where a simulator is used to estimate the distribution of a quantity of interest as in Eq. (14), we estimate $f_Q(q)$ using Monte Carlo simulation followed by kernel density estimation. Each evaluation of q in the Monte Carlo simulation consists of randomly sampling $\mathbf{x}$ according to its joint distribution, evaluating $g(\mathbf{x})$, randomly sampling from $\varepsilon(\mathbf{x})$, and adding the simulator and noise values. Once we have an estimate of $f_Q(q)$, we compute the information entropy of Q via a numerical integration technique. Following Ref. [30], assuming that $f_Q(q)$ is Riemann-integrable, $h(Q)$ can be estimated by discretizing $f_Q(q)$ into N equal bins of size Δ. The quantity $h(Q^\Delta)$

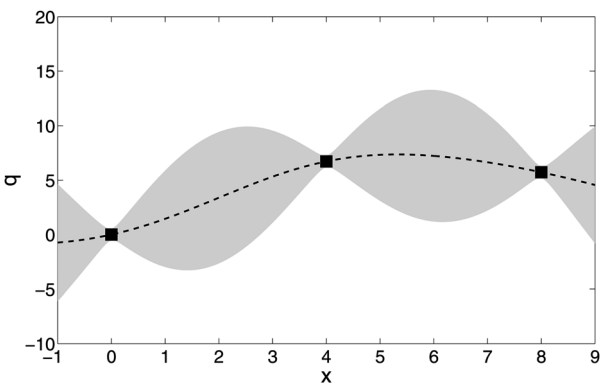

Fig. 1 Example of Gaussian process emulation with three training points. The dashed line is the mean function of the emulator. The grayed area is the ±2 standard deviation confidence interval for the emulator.

represents the estimate of $h(Q)$ computed using numerical approximation

$$h(Q^{\Delta}) = -\sum_{j=1}^{N} [f_Q(q^j)\Delta] \log[f_Q(q^j)\Delta] + \log \Delta \tag{16}$$

Note that $h(Q^{\Delta}) \to h(Q)$ as $\Delta \to 0$. An estimate of complexity is then given by $\hat{C}(Q) = \exp\{h(Q^{\Delta})\}$.

For situations where an emulator must be used in place of a simulator to compute quantities of interest, the complexity estimate must also account for code uncertainty. In this case, the procedure described in the preceding paragraph is conducted for each sample of the emulator stochastic process. Letting the complexity estimated for a particular sample of the emulator be written as $\hat{C}(Q|\mathcal{G} = G)$, the complexity estimate in the presence of code uncertainty is defined here as

$$\tilde{C}(Q) = \max_{G}(\hat{C}(Q|\mathcal{G} = G)) \tag{17}$$

which we estimate using Monte Carlo simulation. Equation (17) reveals that our definition of complexity in the presence of code uncertainty is conservative in the sense that the largest complexity estimate that results from different emulator samples is taken to be the complexity.

3.3 Sensitivity Analysis. For situations where the system complexity is large, it is desirable to identify factors of the system, which include inputs, parameters, components, subsystems, simulators, and emulators that are the largest contributors to the complexity. Thus, we have developed a rigorous sensitivity analysis procedure for indentifying the most significant factor contributors to the system complexity associated with the quantities of interest. The development of such a procedure has been achieved with respect output dispersion measures such as variance [44], Kullback–Liebler divergence of output distributions [45], and moment independent importance indicators [46]. The approach taken here is similar to that of variance-based sensitivity analysis as described in Ref. [44]. In the variance-based case, the goal is to apportion the variance of a quantity of interest among its various factor contributors. This apportionment is based on the law of total variance, which for a given quantity of interest Q and a given factor X_i is written as

$$\mathbb{V}\text{ar}(Q) = \mathbb{E}[\mathbb{V}\text{ar}(Q|X_i)] + \mathbb{V}\text{ar}(\mathbb{E}[Q|X_i]) \tag{18}$$

From this, a main effect sensitivity index, S_i, for factor X_i can be written as

$$S_i = \frac{\mathbb{V}\text{ar}(\mathbb{E}[Q|X_i])}{\mathbb{V}\text{ar}(Q)} \tag{19}$$

which is the expected fraction of the variance of Q that is removed if the true value of X_i was known.

We consider the expected complexity of the system that would remain if the true value of some factor X_i was known. This quantity is given as $\mathbb{E}[C(Q|X_i)]$, where the random variable associated with the quantity of interest for the system is Q. Thus, to identify the expected fraction of complexity that can be removed if the true value of a given factor X_i is known, we define complexity-based sensitivity indices as

$$\eta_i = \frac{C(Q) - \mathbb{E}[C(Q|X_i)]}{C(Q)} \tag{20}$$

where here the uncertainty associated with X_i is attributable to either parametric variability or parametric uncertainty. In Sec. 3.4, we consider the numerical estimation of complexity-based sensitivity indices, which includes the potential presence of model inadequacy and code uncertainty.

The information gained from our sensitivity analysis procedure can be used as part of a resource allocation strategy aimed at reducing system complexity. It is important to note here that the system complexity we are referring to is that of our proposed definition based on the potential for unexpected behavior. For other definitions of system complexity different means should be taken for complexity reduction. For example, if structural complexity is a concern for a particular design, increased modularity could be a viable means for complexity reduction. In this work, we deal exclusively with our proposed definition, and hence aim to increase knowledge of the system quantities of interest via identification of key sources of uncertainty in the system.

3.4 Estimation of Sensitivity Indices. We consider first the estimation of sensitivity indices in the case for which code uncertainty is not present. In order to compute $\mathbb{E}[C(Q|X_i)]$ as given in Eq. (20), it is necessary to evaluate the complexity of Q given X_i for each possible value of X_i on its domain. If X_i is approximated by m samples with values $x_i^1, \ldots, x_i^m$, then $\hat{C}(Q|X_i = x_i^j)$ must be computed for each of $j = 1, \ldots, m$. By the Law of Large Numbers, the mean of the N evaluations approximates the expected value $\mathbb{E}[C(Q|X_i)]$, and allows η_i to be estimated as

$$\hat{\eta}_i = \frac{\hat{C}(Q) - \mathbb{E}_{X_i}[\hat{C}(Q|X_i = x_i^j)]}{\hat{C}(Q)} \tag{21}$$

In general, estimating η_i is computationally intensive. For k factors, if the $\hat{C}(Q|X_i = x_i^j)$ estimates involve N simulator evaluations and $j = 1, \ldots, m$, then we require kmN simulator evaluations, which can be prohibitively expensive. Thus, it is often necessary to employ an emulator of the simulator model, which results in the addition of code uncertainty and the need to modify the sensitivity index estimates for the different system factors. The sensitivity indices for factors associated with parametric uncertainty or parametric variability are estimated as

$$\tilde{\eta}_i = \mathbb{E}_{\mathcal{G}}[\hat{\eta}_i(\mathcal{G})] \tag{22}$$

which is estimated via Monte Carlo simulation. The indices are computed for each sample of the emulator and then averaged to give the expected sensitivity index for each factor with associated parametric variability or parametric uncertainty.

It is important to also estimate the sensitivity indices of both model inadequacy and code uncertainty. For example, if model inadequacy is a key contributor to complexity, it may be possible to incorporate higher fidelity simulation models. If code uncertainty is a key contributor, it may be possible to better train the emulator with more samples from the simulator. The sensitivity index for model inadequacy is defined as

$$\tilde{\eta}_{\text{MI}} = \frac{\tilde{C}(Q) - \tilde{C}(Q|\varepsilon(\mathbf{x}) = 0)}{\tilde{C}(Q)} \tag{23}$$

The sensitivity index for code uncertainty is defined as

$$\tilde{\eta}_{\text{CU}} = \frac{\tilde{C}(Q) - \mathbb{E}_{\mathcal{G}}[\hat{C}(Q|\mathcal{G} = G)]}{\tilde{C}(Q)} \tag{24}$$

While computing estimates of the sensitivity indices defined here involve Monte Carlo simulations, it is important to note that the simulations are conducted with the Gaussian process emulator, which is computationally inexpensive. Indeed, the most expensive aspect of the complexity analysis is typically the training of the emulator as a result of the expense incurred from running the simulator.

3.5 Complexity Estimation Algorithm. We present here algorithms for computing the complexity and the complexity-based sensitivity indices of a system for a given quantity of interest. We assume that we are using an emulator we have already trained. The training of an emulator is a machine learning task that is beyond the scope of this work. The algorithms do not require an emulator be used if the simulator computational costs are reasonable. The use of a simulator results in minor modifications to the algorithms.

Algorithm 1 provides a procedure for the complexity estimation. Since an emulator is being used, we assume we can afford to evaluate it a large number of times m, and that we can sample a large number of emulators n, from the underlying Gaussian process. Therefore, we do not consider any specific convergence criteria in the algorithm.

Algorithm 1: Complexity Estimation

1: Generate m input samples $\{\mathbf{x}^1, \ldots, \mathbf{x}^m\}$
2: Generate m model inadequacy samples $\{\varepsilon^1, \ldots, \varepsilon^m\}$
3: Generate n emulator samples $\{G^1, \ldots, G^n\}$
4: For $i = 1$ to n
  For $j = 1$ to m
    Evaluate $q_{ij} = G^i(\mathbf{x}^j) + \varepsilon^j$
  End
 End
5: For $i = 1$ to n
  Estimate $f_Q^i(q)$ using kernel density estimation and the q values in $\{q_{ij}\}_{j=1}^m$
  Estimate $h^i(Q^\Delta)$ according to Eq. 16 with $f_Q^i(q)$
  Estimate $\hat{C}^i(Q) = \exp\{h^i(Q^\Delta)\}$
 End
6: Estimate $\tilde{C}(Q)$ as the maximum $\hat{C}^i(Q)$ from Step 5

Algorithm 2 provides a procedure for estimating the complexity-based sensitivity indices for inputs with either parametric uncertainty or variability. The complexity-based sensitivity index for model inadequacy can be computed according to Eq. (23) by using Algorithm 1 to compute $\tilde{C}(Q)$ and Algorithm 1 without incorporating the model inadequacy samples to compute $\tilde{C}(Q|\varepsilon(\mathbf{x}) = 0)$. The complexity-based sensitivity index for code uncertainty can be computed according to Eq. (24) by using the $\tilde{C}(Q)$ calculated in Step 6 of Algorithm 1 and the average of the $\hat{C}^i(Q)$ values computed in Step 5 of Algorithm 1.

Algorithm 2: Estimation of Sensitivity Index for Factor i

1: Generate m samples of x_i, $\{x_i^1, \ldots, x_i^m\}$
2: Generate N samples of $\mathbf{x}_{\sim i}$, $\{\mathbf{x}_{\sim i}^1, \ldots, \mathbf{x}_{\sim i}^N\}$, where $\mathbf{x}_{\sim i} = [x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k]^T$
3: For $p = 1$ to N
  Let $\mathbf{x}^{pj} = [\mathbf{x}_{\sim i}^p, x_i^j]^T$
 End
4: Generate p model inadequacy samples $\{\varepsilon^1, \ldots, \varepsilon^p\}$
5: Generate n emulator samples $\{G^1, \ldots, G^n\}$
6: For $r = 1$ to n
  For $j = 1$ to m
    For $p = 1$ to N
      Evaluate $q_{rjp} = G^r(\mathbf{x}^{pj}) + \varepsilon^p$
    End
  End
 End
7: For $r = 1$ to n
  For $j = 1$ to m
    Estimate $f_Q^{rj}(q)$ using kernel density estimation and the q values in $\{q_{rjp}\}_{p=1}^N$
    Estimate $h(Q^\Delta)$ according to Eq. (16) with $f_Q^{rj}(q)$
    Estimate $\hat{C}(Q|X_i = x_i^j) = \exp\{h(Q^\Delta)\}$
  End
  Average over the $\hat{C}(Q|X_i = x_i^j)$ to estimate $\hat{\eta}_i^r$
 End
8: Average over the $\hat{\eta}_i^r$ to estimate $\tilde{\eta}_i$ as in Eq. (22)

4 Application of the Methodology

We demonstrate the use of the complexity metric and sensitivity analysis developed in Sec. 3 on a simulation-based design of an infantry fighting vehicle. The quantity of interest for this demonstration is the range of the vehicle. The simulation models for the IFV, the sources of uncertainty considered, and the emulation models for the IFV simulator are discussed in the Secs. 4.1–4.6.

4.1 The IFV Bond Graph Model. An IFV is a complicated system consisting of many different subsystems and thousands of components. The simulation of such a vehicle requires several different disciplinary analyses, such as electrical, mechanical, and hydrodynamical. A common strategy for modeling the dynamics of such multidisciplinary systems is through bond graphs [47–49]. Bond graphs were first proposed in Ref. [50]. Bond graphs are based on energy conservation in a system and provide a discipline-independent means of describing the dynamics of a physical system. Further details on bond graphs can be found in Refs. [47–49].

The specific IFV we are considering here comes from a set of IFV designs developed by the Vanderbilt University Institute for Software Integrated Systems (ISIS). These designs were developed as part of an attempt to redefine the systems engineering process as part of the defense advanced research projects agency's (DARPA) adaptive vehicle make plan to develop better complex vehicles more quickly and at lower cost [51]. A goal of our work is thus to demonstrate the use of the complexity metric developed here on these important complex vehicles.

The IFV designs consist of many subsystems, such as the complete powertrain of the vehicle, the vehicle hull and chassis, as well as the integrated starter generator. Figure 2 shows the bond graph representation of the driveline of the IFV designs. The source of power in the system is the engine, which is represented as a source of effort, which it produces as torque. Inductive elements of the model consist of the inertia of the engine, the gear box, the wheels, and the vehicle mass. The only capacitive element of the model is the compliance of the drive shaft. The resistive elements of the model are the clutch resistance, the wind drag, and the road resistance on the wheels. Though this model is simple in many respects, it is still capable of obtaining good

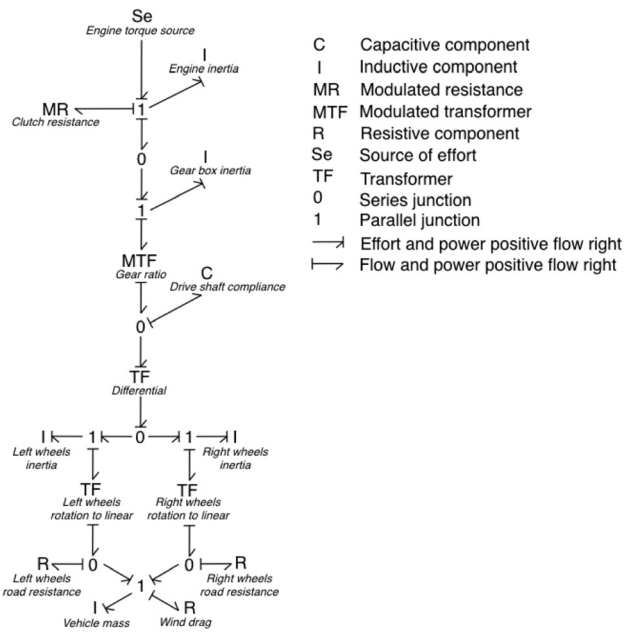

Fig. 2 Bond graph representation of the driveline of the IFV candidate designs (adapted from Ref. [49])

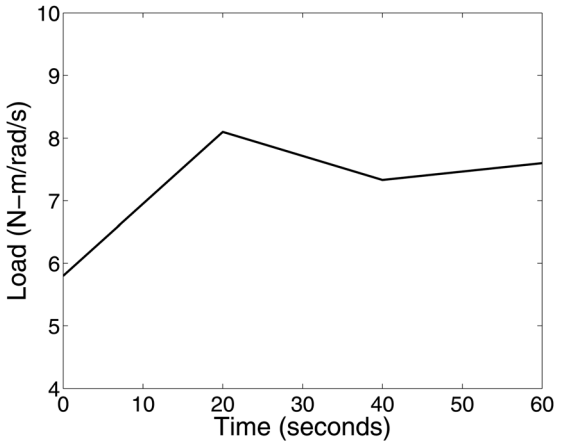

Fig. 3 First 60 s of the simulated terrain used in the estimate of the range of the IFV

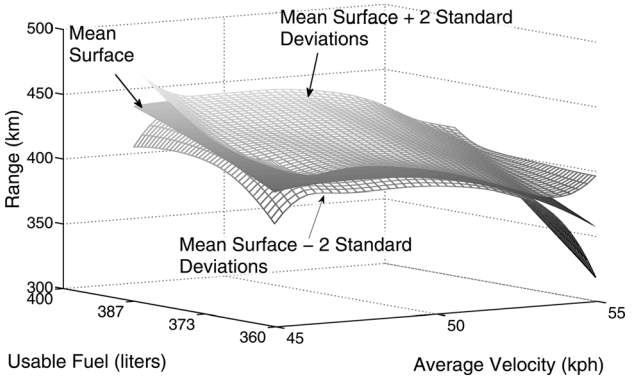

Fig. 4 Gaussian process model of a candidate IFV design. The lighter gray surfaces represent the ±2 standard deviation surfaces from the mean surface, which is shown as the darker mesh between the top and bottom surfaces.

estimates of many aspects of vehicle behavior [49]. There are of course many other subsystems of the IFV designs; however, for the sake of brevity, we do not provide a complete description of them here.

It is important to note that the bond graph representation of the IFV configuration depicted in Fig. 2 is more than just an architecture for the driveline of the vehicle. The diagram also fully encapsulates the dynamics of the vehicles driveline, which consists of a set of differential equations derived from the bond graph. These differential equations are solved numerically to provide estimates of quantities of interest of the vehicle (e.g., the range of the vehicle), which are then used in the estimation of the complexity of the system with respect to our proposed definition of complexity.

4.2 IFV Range Calculation. The quantity of interest for this demonstration of our methodology is the range of a candidate IFV design. The range of the vehicle is estimated by simulating vehicle on a typical terrain course. The terrain used for this demonstration is shown in Fig. 3. Here, the vertical axis represents the load experienced by the vehicle as a result of an undulating terrain. The terrain was generated from Gaussian noise.

4.3 IFV Simulation Emulators. A single simulation of an IFV design for the range calculation takes approximately 1500 s on a standard laptop computer. The estimation of the complexity metric and the subsequent sensitivity analysis involves the estimation of several potentially high dimensional integrals, which could require thousands of function evaluations if Monte Carlo simulation is employed. Thus, for the IFV application, we wish to generate Gaussian process models of the candidate IFV design to emulate the simulation of the vehicle. The Gaussian process model of the potential IFV design constructed here is shown in Fig. 4. The Gaussian process was trained with 20 training points from the bond graph simulation model.

4.4 IFV Sources of Uncertainty. As noted previously, there are many sources of uncertainty that affect estimates of quantities of interest for a complex system. For the IFV range application, we are considering parametric uncertainty, parametric variability, code uncertainty, and model inadequacy. Thus, for the stages of complex system design that involve computer simulation models, we have included all sources of uncertainty.

The parametric uncertainty we consider here is the result of an uncertain amount of trapped fuel that cannot be used by the IFV. The uncertainty in the amount of trapped fuel is captured by considering the available fuel at the beginning of the mission to be uniformly distributed from 360 to 400 l. Thus, we are assuming between 0% and 10% of the fuel will be unusable. In general, such information should be obtained from expert opinion or historical data [52]. Here, we have assigned the distribution for demonstration purposes only. The parametric variability we consider here is the result of different possible human operators of the IFV driving at different speeds. We assume that each operator is attempting to operate the tank at 50 kph; however, each operator may be more or less skilled at achieving this objective. To account for this, we allow the target velocity of the vehicle to be uniformly distributed between 45 and 55 kph. If this uncertainty is found to be a major contributor to complexity, an obvious next step in the design process is to ensure adequate feedback information to the operator to ensure the operator is capable of maintaining the vehicle at the target velocity. The model inadequacy we consider here is assumed to be normally distributed with mean 0 and a standard deviation of 10 km. This uncertainty is added to the output of the emulator. We have assumed that the model inadequacy is constant throughout the input space. The code uncertainty we consider here is captured by the variability between training points in the Gaussian process model. There are of course many other parameters that would be uncertain at an early stage of the design of a complex vehicle such as the IFV considered here. However, our goal is to demonstrate our methodology rather than perform a complete complexity analysis of the IFV design.

4.5 IFV Complexity Estimation. Following the procedure outlined in Sec. 3.2, we estimate the complexity of the IFV design using Eq. (17). The result is $\tilde{C}(Q) = 104$ km, where Q denotes the random variable associated with the range of the IFV. Distributions of the range of the IFV are shown in Fig. 5. Here, two distributions are shown in solid black lines that were estimated using two different samples of the Gaussian process emulator shown in Fig. 4. The dashed gray lines are the output distributions from the same two samples of the Gaussian process emulator; however, for these distributions, model inadequacy has been included.

4.6 IFV Sensitivity Analysis. Following the procedure outlined in Sec. 3.4, we estimate the sensitivity indices of the average velocity (AV), usable fuel (UF), model inadequacy (MI), and code uncertainty with respect to the quantity of interest, IFV range. The results of the sensitivity analysis are shown in Fig. 6. As shown on the figure, the sensitivity indices are

$$\tilde{\eta}_{\mathrm{AV}} = 0.46$$

$$\tilde{\eta}_{\mathrm{UF}} = 0.44$$

$$\tilde{\eta}_{\mathrm{MI}} = 0.15$$

$$\tilde{\eta}_{\mathrm{CU}} = 0.16$$

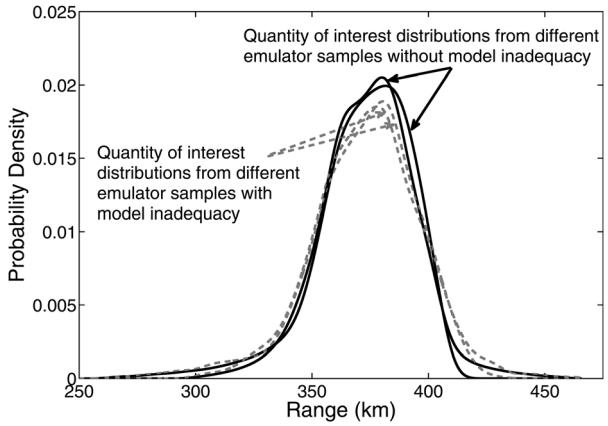

Fig. 5 Probability density functions of the range of the IFV. The solid lines represent two distributions that were estimated using two different samples of the Gaussian process emulator. The dashed gray lines are the output distributions from the same two samples of the Gaussian process emulator; however, for these distributions, model inadequacy has been included.

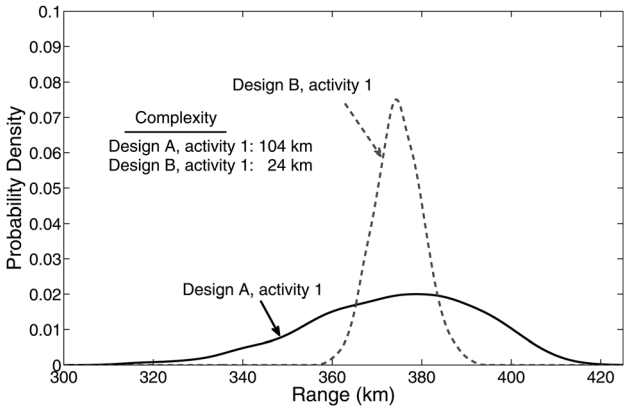

Fig. 7 Sampled distributions and complexity estimates of the range quantity of interest of the two different IFV design options. Design A uses newer technology for the vehicle velocity control and fuel systems, whereas design B uses older technology for both systems.

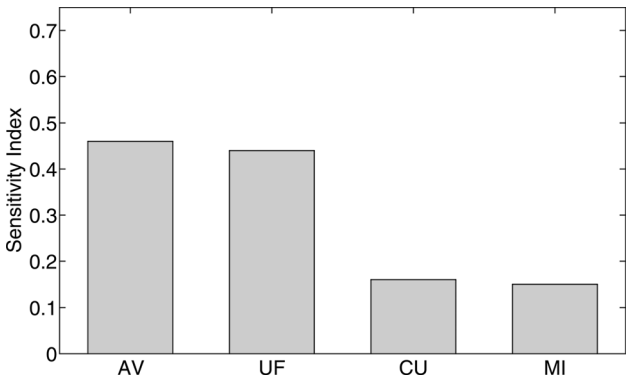

Fig. 6 Sensitivity indices of the four factors that impact the complexity of the IFV design with respect to range. From left to right the indices are for average velocity (AV), usable fuel (UF), code uncertainty (CU), and model inadequacy (MI).

where $\tilde{\eta}_{\mathrm{AV}}, \tilde{\eta}_{\mathrm{UF}}, \tilde{\eta}_{\mathrm{MI}}$, and $\tilde{\eta}_{\mathrm{CU}}$ are the sensitivity indices for the average velocity, usable fuel, model inadequacy, and code uncertainty, respectively. Here we note that global sensitivity analyses, such as the variance-based work of Refs. [44,53] typically consider the percentage of output variance of which a particular factor is responsible. Thus, sensitivity indices are typically reported to two significant figures. We follow this convention here.

If the complexity, with respect to range, of the IFV design is considered too large, resources should be allocated to the various contributors of that complexity. There are many possible situations where system complexity may be deemed too large. Examples include comparisons among candidate designs with overlapping distributions of quantities of interest, designs with unacceptably high probability of not meeting a critical design constraint, and conceptual design processes where the goal is simply to better understand the capabilities of a given system. For the case of this IFV design, without consideration of cost and available resources, the most efficient means of reducing complexity would be to allocate resources to learn more about the average velocity or the usable fuel, which have the largest sensitivity indices.

In general the resource allocation problem is made more complicated by the fact that available and required resources may be to some extent uncertain, and it is unlikely that any given activity be able to identify the true value of a given parameter or fully reduce model inadequacy or code uncertainty. Nevertheless, the complexity-based sensitivity indices provide essential information regarding the key contributors to system complexity and will greatly contribute to the ability of a system designer to reduce system complexity. It is a topic of future work to consider the more general resource allocation problem under uncertainty.

5 Design Comparison Demonstration

To demonstrate the use of our methodology when multiple design options are available, we consider here a notional example where design A consists of new technologies for the vehicle velocity control system and fuel system, and design B consists of older technologies for those same systems. For this demonstration, design A is the IFV considered in Sec. 4 and design B is the same general IFV configuration only with different input distributions for the average velocity and the usable fuel. Hence, the emulator developed in Sec. 4.3 is used throughout this demonstration. Here, again, the range of the vehicles are the quantities of interest and the objective at this point in the design process is to select the design concept that results in the longest range for the IFV.

5.1 Activity 1: Initial Complexity Estimation. The first design activity is the estimation of the complexity of each design with respect to the range of the vehicles. The probability distribution of the average velocity for design A, as noted in Sec. 4.4 is given as $\sim U[45, 55]$ kph and that of the usable fuel is given as $\sim U[360, 400]$ l. The probability distribution of the average velocity for design B is given as $\sim U[50, 51]$ kph, and the probability distribution of the usable fuel is given as $\sim U[376, 384]$ l. The larger extent of the design A input distributions with respect to extent of the design B input distributions reflects the fact that the new technologies are less well-understood. For each design, Algorithm 1 was used to estimate the complexity and output distributions for the range of the vehicles. Sample output distributions, which represent output distributions computed using specific emulator samples that are representative of the ensemble of output distributions computed by sampling over the emulators, and complexity estimates are shown in Fig. 7. At this point in the design process, it is not clear whether the range of design A will be longer or shorter than that of design B. Also, the complexity of design A is much larger than that of design B, which reflects the fact that the extent of the potential values the range of design A can take is larger than that of design B. The implication being that we know much less about the impact of choosing design A on vehicle range than the impact of choosing design B.

When this situation occurs in the design process, a decision must be made regarding whether to allocate resources towards learning more about the impact of the new technology on range, abandon the new technology in favor of the lower complexity older technology, or move forward with the new technology with the understanding that the incorporation of the new technology leads to a more complex system at this time. Here, we assume resources are available to consider the impact of the new technology more carefully. To best allocate these resources, we use the sensitivity analysis procedure developed in Sec. 3.3 to identify the key contributors to the complexity of design A. The results of this analysis were given previously in Sec. 4.6, where it was established that the key contributors to the complexity of this design are the average velocity and usable fuel.

5.2 Activity 2: Refine Average Velocity Estimate for Design A. Since average velocity is the largest contributor to the complexity of design A, resources are allocated to a notional design activity that improves our understanding of the average velocity that would result if design A were implemented. Examples of what this activity could be include gathering data from a designed experiment of IFV operators using the new velocity control system in a simulator or even constructing a prototype of the design A IFV and measuring the range of the vehicle. The activities that can be performed will depend on the level of resources available. We assume here that the activity chosen to learn more about the velocity control system produced a new estimate of the distribution of the average velocity $\sim U[46, 46.5]$ kph. With this new input information, Algorithm 1 was used to estimate the complexity of design A. Figure 8 presents the results in terms of a representative sample output distribution of design A after the activity, which is labeled as *design A, activity 2*. The complexity results for the initial design A and design B range estimates are also shown on the figure along with representative distributions for each estimate. As can be seen on the figure, the complexity of design A following activity 2 has been reduced owing to the improved understanding of the average velocity of the design. However, it is also clear from the figure that there is still some potential for design A to have a shorter range than design B as evidenced by the left tail of the representative range distribution for design A after activity 2 and the overlapping region of this distribution with the design B representative distribution. Here, another design decision must be made regarding whether to allocate further resources to design A to reduce the complexity of the design and learn whether or not design A is preferred to design B, to proceed with the lower complexity design B, or to proceed with the higher complexity design A. We assume here that resources are still available to be allocated for complexity reduction of design A.

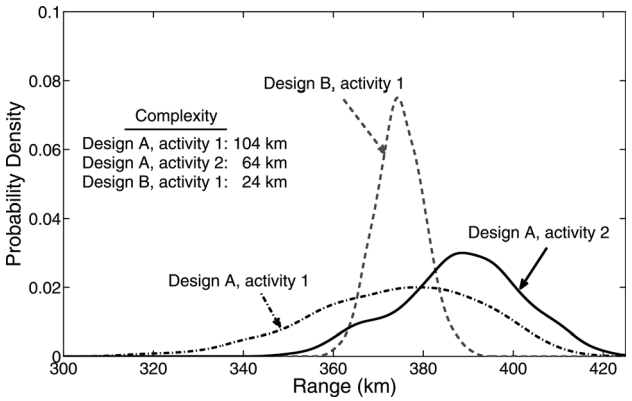

Fig. 8 Sampled distributions and complexity estimates of the range quantity of interest of design A both before and after activity 1, as well as design B results

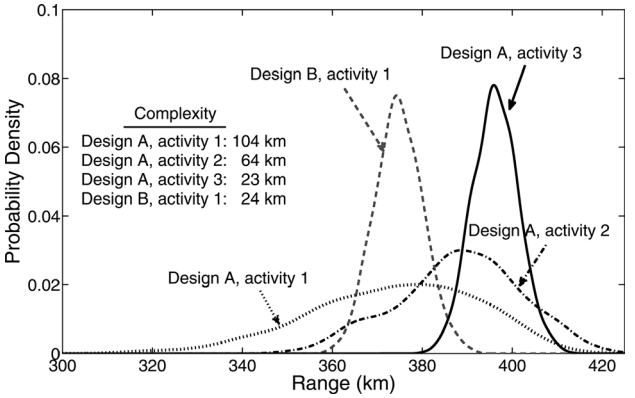

Fig. 9 Sampled distributions and complexity estimates of the range quantity of interest of design A both before and after activity 1, as well as design B results

5.3 Activity 3: Refine Usable Fuel Estimate for Design A. Since usable fuel is the next largest contributor to the complexity of design A, resources are allocated to a notional design activity that improves our understanding of the usable fuel that would result if design A were implemented. We assume that the result of the activity is a new estimate of the distribution of the usable fuel that is now $\sim U[386, 390]$ l. With this new input information, Algorithm 1 was used to estimate the current complexity of design A. Figure 9 presents the results in terms of a representative sample output distribution of design A after the activity, which is labeled as *design A, activity 3*. The complexity results for the initial design A, the design A after the second activity, and design B range estimates are also shown on the figure along with representative distributions for each estimate. Once again, the complexity of design A has been reduced and is now slightly less than that of design B. Thus, we now have essentially the same knowledge regarding the range of both designs. Further, though there is still potential for design A to produce a shorter range vehicle than design B owing to the small overlapping regions of the distributions, based on the current level of knowledge it is likely that design A will produce the longer range vehicle.

5.4 Discussion. The demonstration presented here provides an indication of the usefulness of the complexity metric and sensitivity analysis methodology as enabling technologies for the management and control of uncertainty in system design processes. In this particular example, an older set of technologies comprising design B was shown to initially be a lower complexity design choice due to a lack of knowledge about the newer technologies of design A. Through the use of sensitivity analysis, key contributors to the lack of knowledge surrounding the performance of design A were identified and notional activities were conducted to learn about these contributors. The end result was a reduction of the complexity of design A that revealed that the design was superior to the older technology design. It is important to note here that the end result could have been that the newer technology was inferior to the older technology design and that only through activities designed to learn more about the performance design A were we able to reduce the complexity of design A and distinguish between the different options. Had these activities not taken place, the complexity of design A would not have been reduced, and though in this case the design was superior, this would not have been discovered until the vehicle was fielded.

6 Conclusions

We have developed an demonstrated a methodology for estimating system complexity with respect to quantities of interest, as well as estimating sensitivity indices designed to indicate key

contributors to system complexity. Our complexity metric can be used to compare and rank different candidate designs of complex systems with respect to quantities of interest. In situations where designs are too complex, our sensitivity analysis methodology can be used to identify key contributors to the complexity, which may then be used to inform a resource allocation process. The incorporation of model inadequacy in our approach ensures that complexity arising from the use of low fidelity models be accounted for, and provides direction, in a resource sense, for a multifidelity approach to complex system design. The incorporation of code uncertainty ensures that uncertainty associated with the use of inexpensive surrogate models be accounted for, and the sensitivity index associated with code uncertainty can potentially be used in the future as part of an adaptive approach to train the emulators.

The work described here assumed the existence of quantified uncertainty in the form of parametric variability, parametric uncertainty, model inadequacy, and code uncertainty. In general, it is critical in the design of complex systems that these uncertainties be rigorously quantified. Systematic methods for achieving this goal are an important topic of future work. Once such methods exist, the use of metrics such as the complexity metric described here, as well as the sensitivity analysis developed here, can be used as part of a design verification strategy aimed at producing probabilistic certificates of correctness for designs through simulation.

Acknowledgment

This work was funded in part by the DARPA META program through AFRL Contract No. FA8650-10-C-7083 and Vanderbilt University Contract No. VU-DSR #21807-S7, and by the Singapore University of Technology and Design. The authors would also like to acknowledge the Vanderbilt University Institute for Software Integrated Systems for providing the IFV model used in this work.

Nomenclature

$C(\cdot)$ = complexity
$\mathcal{D}$ = training set
$\mathbb{E}[\cdot]$ = expectation
$f_X(x)$ = probability density function
$\mathcal{GP}(\cdot,\cdot)$ = Gaussian process model
$H(\cdot)$ = information entropy
$h(\cdot)$ = differential entropy
IFV = infantry fighting vehicle
$M_t(\cdot)$ = generalized mean of order t
$p(\cdot)$ = probability mass function
Q = quantity of interest random variable
$\mathbb{Q}$ = support of a random variable Q
$\mathcal{R}(\cdot)$ = range of a random variable
S_i = main effect sensitivity index of factor i
$\mathbb{V}\mathrm{ar}(\cdot)$ = variance
Δ = bin size
η_i = complexity-based sensitivity index of factor i

References

[1] Allen, L., Angel, R., Mangus, J. D., Rodney, G. A., Shannon, R. R., and Spoelhof, C. P., 1990, "The Hubble Space Telescope Optical Systems Failure Report," Technical Report No. NASA-TM-103443, National Aeronautics and Space Administration.
[2] Bolkcom, C., 2005, "V-22 Osprey Tilt-Rotor Aircraft: Congressional Research Service Report for Congress," Technical Report, Congressional Research Service. Available at http://www.history.navy.mil/library/online/v-22%20osprey.htm
[3] Hiltzik, M., 2011, *787 Dreamliner Teaches Boeing Costly Lesson on Outsourcing*, Los Angeles Times, Los Angeles, CA.
[4] Brown, O., and Eremenko, P., 2006, "The Value Proposition for Fractionated Space Architectures," AIAA Space 2006, San Jose, CA, Paper No. AIAA-2006-7506.
[5] Campbell, L., 1966, "Exponential Entropy as a Measure of Extent of a Distribution," Z. Wahrscheinlichkeitstheorie verw. Geb., **5**, pp. 217–225.
[6] Takeda, H., Veerkamp, P., Tomiyama, T., and Yoshikawa, H., 1990, "Modeling Design Processes," AI Mag., **11**(4), pp. 37–48.
[7] Weaver, W., 1948, "Science and Complexity," Am. Sci., **36**, pp. 536–544.
[8] Huberman, B. A., and Hogg, T., 1987, "Phase Transitions in Artificial Intelligence Systems," Artif. Intell., **33**, pp. 155–171.
[9] Johnson, G., 1997, "Researchers on Complexity Ponder What It's All About," The New York Times: Technology, May 6, 1997.
[10] Balestrini-Robinson, S., 2009, "A Modeling Process to Understand Complex System Architectures," Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA.
[11] von Bertalanffy, L., 1950, "An Outline of General System Theory," Br. J. Philos. Sci., **1**(2), pp. 134–165.
[12] Eriksson, D., 1997, "A Principal Exposition of Jean-Louis Le Moigne's Systemic Theory," Cybern. Human Knowing, **4**(2).
[13] Corning, P. A., 1998, "Complexity is Just a Word!," Technol. Forecast. Soc. Change, **59**, pp. 197–200.
[14] Bankes, S. C., 2002, "Tools and Techniques for Developing Policies for Complex and Uncertain Systems," Proc. Natl. Acad. Sci. U.S.A., **99**, pp. 7263–7266.
[15] Summers, J. D., and Shah, J. J., 2010, "Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability," J. Mech. Des., **132**(2), p. 021004.
[16] Braha, D., and Maimon, O., 1998, "The Measurement of a Design Structural and Functional Complexity," IEEE Trans. Syst. Man Cybern., Part A. Syst. Humans, **28**(4), pp. 527–535.
[17] Albrecht, A. J., and Gaffney, J. E., Jr., 1983, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation," IEEE Trans. Softw. Eng., **SE-9**(6), pp. 639–648.
[18] Murmann, P. A., 1994, "Expected Development Time Reductions in the German Mechanical Engineering Industry," J. Prod. Innov. Manage., **11**(3), pp. 236–252.
[19] Griffin, A., 1997, "The Effect of Project and Process Characteristics on Product Development Cycle Time," J. Mark. Res., **34**(1), pp. 24–35.
[20] Meyer, M. H., and Utterback, J. M., 1995, "Product Development Cycle Time and Commercial Success," IEEE Trans. Eng. Manage., **42**(4), pp. 297–304.
[21] Braha, D., and Bar-Yam, Y., 2004, "Topology of Large-Scale Engineering Problem-Solving Networks," Phys. Rev., **69**(1), p. 016113.
[22] Braha, D., and Bar-Yam, Y., 2007, "The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results," Manage. Sci., **53**(7), pp. 1127–1145.
[23] McCabe, T. J., 1976, "A Complexity Measure," IEEE Trans. Softw. Eng., **2**(4), pp. 308–320.
[24] Tatikonda, M. V., and Rosenthal, S. R., 2000, "Technology Novelty, Project Complexity, and Product Development Project Execution Success: A Deeper Look at Task Uncertainty in Product Innovation," IEEE Trans. Eng. Manage., **47**(1), pp. 74–87.
[25] Novak, S., and Eppinger, S. D., 2004, "Sourcing by Design: Product Complexity and the Supply Chain," Manage. Sci., **47**(1), pp. 189–204.
[26] Kim, J., and Wilemon, D., 2009, "An Empirical Investigation of Complexity and Its Management in New Product Development," Technol. Anal. Strategic Manage., **21**(4), pp. 547–564.
[27] Hölttä, K. M. M., and Otto, K. N., 2005, "Incorporating Design Effort Complexity Measures in Product Architectural Design and Assessment," Des. Stud., **26**(5), pp. 463–485.
[28] Kolmogorov, A. N., 1965, "Three Approaches to the Quantitative Definition of Information," Problems Inf. Transm., **1**(1), pp. 1–7.
[29] Chaitin, G. J., 1969, "On the Simplicity and Speed of Programs for Computing Infinite Sets of Natural Numbers," J. ACM, **16**(3), pp. 407–422.
[30] Cover, T. M., and Thomas, J. A., 1991, *Elements of Information Theory*, John Wiley & Sons, Inc., New York, NY.
[31] Suh, N. P., 1990, *The Principles of Design*, Oxford University Press, New York, NY.
[32] Rodriguez-Toro, C. A., Tate, S. J., Jared, G. E. M., and Swift, K. G., 2003, "Complexity Metrics for Design (Simplicity + Simplicity = Complexity)," Proc. Inst. Mech. Eng., Part B: J. Eng. Manuf., **217**, pp. 721–725.
[33] Shannon, C. E., 1948, "A Mathematical Theory of Communication," Bell Syst. Tech. J., **27**, pp. 379–423 and 623–656.
[34] Lloyd, S., and Pagels, H., 1988, "Complexity as Thermodynamic Depth," Ann. Phys., **188**, pp. 186–213.
[35] Coatanea, E., 2005, "Conceptual Design of Life Cycle Design: A Modelling and Evaluation Method Based on Analogies and Dimensionless Numbers," Doctoral Dissertation, Helsinki University of Technology, Helsinki, Finland.
[36] Krus, P., and Andersson, J., 2004, "An Information Theoretical Perspective on Design Optimization," Proceedings of DETC'04, ASME.
[37] Dannon, H. V., 2008, "Power Means Calculus, Product Calculus, Harmonic Mean Calculus, and Quadratic Mean Calculus," Gauge Inst. J., **4**(4).
[38] Hardy, G. H., Littlewood, J. E., and Pólya, G., 1934, *Inequalities*, Cambridge University Press, Cambridge, UK.
[39] Kennedy, M., and O'Hagan, A., 2001, "Bayesian Calibration of Computer Models." J. R. Stat. Soc. B, **63**(3), pp. 425–464.
[40] O'Hagan, A., 1978, "Curve Fitting and Optimal Design for Prediction," J. R. Stat. Soc. B (Methodol.), **40**, pp. 1–42.
[41] O'Hagan, A., 2006, "Bayesian Analysis of Computer Code Outputs: A Tutorial," Reliab. Eng. Syst. Saf., **91**, pp. 1290–1300.

[42] Rasmussen, C., and Williams, K., 2006, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, Massachusetts.
[43] Sacks, J., Welch, W., Mitchell, T., and Wynn, H., 1989, "Design and Analysis of Computer Experiments," Stat. Sci., **4**(4), pp. 409–435.
[44] Homma, T., and Saltelli, A., 1996, "Importance Measures in Global Sensitivity Analysis of Nonlinear Models," Reliab. Eng. Syst. Saf., **52**, pp. 1–17.
[45] Liu, H., Chen, W., and Sudjianto, A., 2006, "Relative Entropy Based Method for Global and Regional Sensitivity Analysis in Probabilistic Design," ASME J. Mech. Des., **128**(2), pp. 1–11.
[46] Borgonovo, E., 2006, "Measuring Uncertainty Importance: Investigation and Comparison of Alternative Approaches," Risk Anal., **26**(5), pp. 1349–1361.
[47] Broenink, J., 1999, "Introduction to Physical Systems Modeling With Bond Graphs," *The SiE Whitebook on Simulation Methodologies*, available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.124.3518
[48] Borutzky, W., 2006, "Bond Graph Modelling and Simulation of Mechatronic Systems An Introduction into the Methodology," 20th European Conference on Modeling and Simulation.
[49] Das, S., 2009, *Mechatronic Modeling and Simulation Using Bond Graphs*, Taylor & Francis Group, LLC, Boca Raton, Florida.
[50] Paynter, H., 1961, *Analysis and Design of Engineering Systems*, MIT Press, Cambridge, Massachusetts.
[51] Belfiore, M., 2012, "Adaptive Vehicle Make: DARPA's Plan to Revolutionize Auto Manufacturing," Popular Mechanics, January.
[52] Oakley, J., and O'Hagan, A., 2007, "Uncertainty in Prior Elicitations: A Non-parametric Approach," Biometrika, **94**, pp. 427–441.
[53] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S., 2008, *Global Sensitivity Analysis: The Primer*, John Wiley & Sons, Ltd., West Sussex, England.