STOCHASTIC PROCESS DECISION METHODS FOR COMPLEX CYBER-PHYSICAL SYSTEMS

K. Willcox, D. Allaire, J. Deyst, C. He, and G. Sondecker

Massachusetts Institute of Technology

JULY 2011
Final Report

TABLE OF CONTENTS

Section **Page**

List of Figures

Figure **Page**

1.0 SUMMARY

The primary objective of our effort is to develop a fundamental theory to quantify the inherent uncertainties and risks in complex system design and development processes. These theoretical developments will help enable the achievement of the META goal of devising, implementing, and demonstrating in practice a radically different approach to the design, integration/manufacturing, and verification of complex systems. Our approach to meeting this objective is: to adapt the entropy concepts of information theory to create a metric for system complexity; to apply estimation theory to characterize inherent uncertainty in system development processes; and to utilize this theoretical base to develop efficient methods for resource allocation so as to manage uncertainty and mitigate risk in complex system developments.

Our specific innovative claims for this project are as follows:

1. Viewing system development as a problem of Bayesian estimation leads to a theoretical framework for complex system development.
2. Quantifying complexity in terms of information theoretic concepts permits the treatment of the complexity metric with the tools of estimation theory. This enables a systematic approach to quantitative modeling of system development as a resource investment procedure in the presence of uncertainty.
3. A stochastic model for system development facilitates quantification of the uncertainty reduction that is necessary for success and can be used as a tool to monitor the actual development process.
4. Our proposed theoretical framework for uncertainty quantification provides the bedrock upon which the methods and tools, enabling orders of magnitude improvement in complex system developments, can be built.

In this research we achieved our objectives by defining a complexity metric, quantifying that complexity, and developing a sensitivity analysis procedure for directing resource allocation in complexity reduction exercises.

2.0 INTRODUCTION

Over the course of a successful system development many large initial uncertainties are systematically reduced, until great certainty is achieved when the actual system is realized. Viewed in this manner the system development process becomes a problem of Bayesian estimation. All decision processes along the way are re-cast as discovery procedures by which models and tests are used, in concert, to conduct a series of observations of the key parameters. A Bayesian characterization of key design parameters is used to represent the level of uncertainty in each parameter at any point during the development process.

Our Bayesian estimation viewpoint provides a natural mathematical/computational framework for integrating statistical and deterministic analysis methods. The overall framework encapsulates uncertainty assessment methods, such as Markov chain Monte Carlo methods for statistical inference, Monte Carlo sampling for forward propagation of uncertainties, global sensitivity analysis for identifying key contributors to total variance in overall system metrics, and stochastic optimization for decision under uncertainty. By viewing the process as part of an estimation problem, we can exploit many deterministic design methods, such as system decomposition, optimization, multifidelity approaches to modeling uncertainty, reduced order modeling, etc.

Figure 1 depicts how an evolving estimate of the system---which comprises both the system state and its current level of uncertainty---is continually fed by data from experiments and simulations as the decision process advances. The evolving system estimate in turn feeds information back to aid in experimental design as well as in development and management of multifidelity models. For example, model parameter estimates may be updated as new design or test data are assimilated. Propagation of uncertainties in parameters through to outputs of interest might indicate high levels of uncertainties in model outputs that require additional experiments or model refinement. Sensitivity analysis indicates important sources of uncertainty and thus specific components of models or parameters that require higher fidelity. The optimization/control task provides specific goals that also inform model selection and parameterization. Evolution of the system through the elements depicted around the cylinder in the figure typically repeats many times over the duration of the project. As the probabilistic estimate of the system state evolves, it contributes towards the different stages of the decision process. These decisions may include resource allocations—perhaps to experiments, or to improving physical modeling, or to improved simulation model capabilities, or to hardware acquisitions—which in turn impact the estimation process.

The rest of this report proceeds as follows. Section 2 presents our project objectives and our approach to meeting them. Section 3 develops our complexity definition and our complexity metric. Section 4 develops our sensitivity analysis methodology for resource allocation. Section 5 demonstrates our methodology on an RLC circuit. Section 6 describes the application of our methodology to a satellite system design program.

Section 7 describes the application of our methodology to a notional infantry fighting vehicle. Section 8 develops theoretical material related to model inadequacy. Section 9 discusses the scalability of our methods. Section 10 discusses future work and Section 11 presents our conclusions.

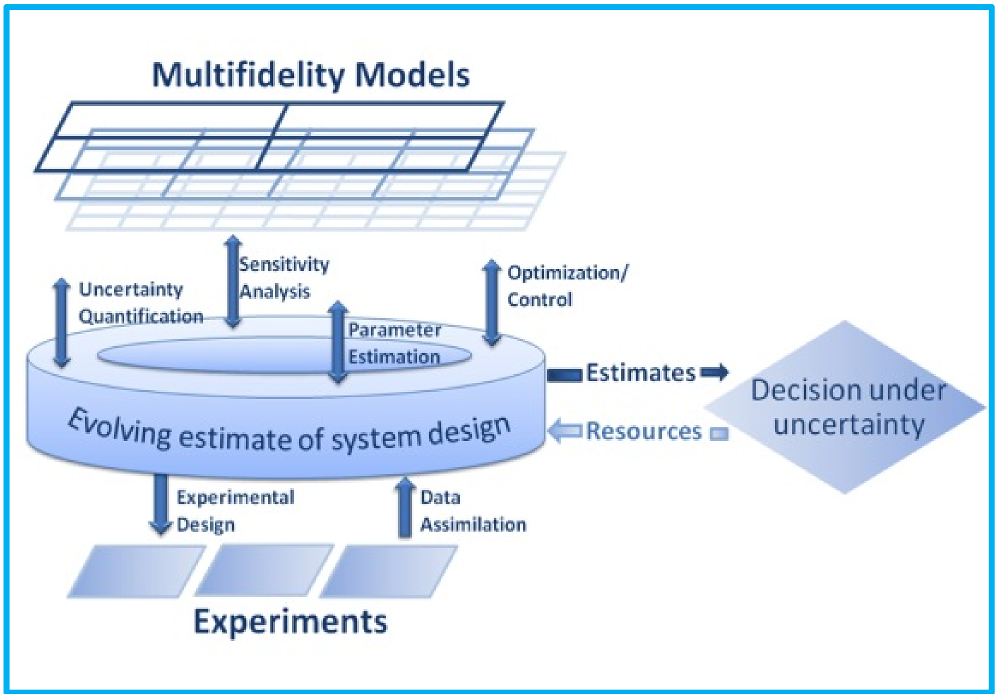


Figure 1: Experimental and Simulation Data Combination

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

In this section we discuss our complexity metric theoretical development, our estimation procedure for the complexity metric, and our development of a sensitivity analysis procedure for resource allocation purposes.

3.1 Complexity Metric Theoretical Development

In the following subsections we present our definition of complexity and our complexity metric, background material on information entropy, a discussion of quantities of interest, and circuit based examples that reveal the uses and interpretation of the metric.

3.1.1 Definition of Complexity and Our Complexity Metric. To successfully manage complexity throughout system design, development, and deployment, a proper definition of complexity is essential. We define complexity as the potential of a system to exhibit unexpected behavior in the quantities of interest, regardless of whether or not that behavior is detrimental to achieving system requirements. Complexity defined in this manner captures the notion of emergent behavior and nonlinear interaction phenomena characteristic of what are typically termed complex systems. This definition also permits a system that does not necessarily exhibit complex behavior to still be defined as complex if there is potential for such behavior. This potential may be measured in terms of unknown possible states that quantities of interest may lie in (which is essentially emergent behavior). Here, quantities of interest refer to anything a decision-maker is interested in and thus, vary by application, level of abstraction being considered, the current stage in the development process, etc. These could be parameters that define the system dynamics and performance, the complete set of state variables for the system, or even cost and schedule information.

3.1.2 Complexity Metric. We claim that an appropriate metric for measuring complexity as defined above can be defined on the basis of information entropy. The reason for this is that we are essentially referring to how uncertain we are about the values our quantities of interest may take, which leads to the notion of emergent behavior. Specifically, our complexity metric is defined as

$$C(Q) = \exp(h(Q)), \tag{1}$$

where Q is the joint distribution of the quantities of interest and h(Q) is the differential entropy of Q. This metric takes values on $(0,\infty)$. As we learn more about our quantities of interest, the complexity metric tends to zero.

3.1.3 Background Material on Information Entropy. Here we present some brief background material on information entropy. For the discrete case, consider a random variable Y with probability mass function p(y). The entropy of Y is then defined as

$$H(Y) = -\sum_i p(y_i) \log p(y_i), \tag{2}$$

where y_1,y_2,… are the values of y such that p(y) does not equal zero. For the continuous case, consider a random variable X with probability density function $f_X(x)$. The differential entropy of X is then defined as

$$h(X) = -\int_{\Omega_X} f_X(x) \log f_X(x) dx, \tag{3}$$

where Ω_X is the support of X. Examples of the information entropy for typical distributions are as follows:

Normal Distribution: $$h(N(\mu,\sigma^2)) = \frac{1}{2}\ln(2\pi e\sigma^2), \tag{4}$$

Uniform Distribution: $$h(U[a,b]) = \ln(b-a), \tag{5}$$

Triangular Distribution: $$h(T(a,b,c)) = \frac{1}{2} + \ln\left(\frac{b-a}{2}\right), \tag{6}$$

where μ is the mean and σ^2 is the variance of the normal distribution, a is the minimum and b is the maximum of the uniform distribution, and a is the minimum, b is the maximum, and c is the mode of the triangular distribution.

3.1.4 Examples of Complexity Calculations. To demonstrate how the metric is calculated and to build intuition as to how to interpret results, we have computed the metric for several cases of the adder circuit shown in Figure 2.

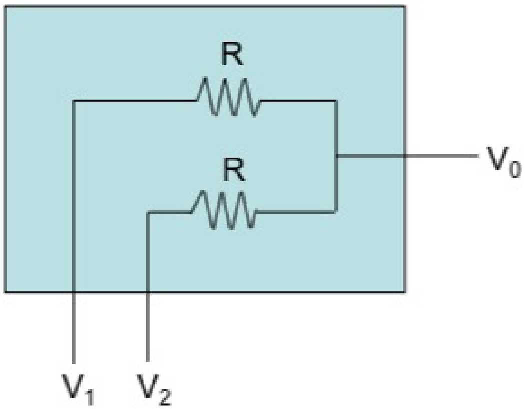


Figure 2: Adder Circuit

Here voltages V_1 and V_2 are input, voltage V_0 is output, and R is resistance. By superposition, we have

$$V_0 = \frac{1}{2}V_1 + \frac{1}{2}V_2. \qquad (7)$$

If we assume V_1~U[0,5]V, V_2~U[0,5]V, and V_1 and V_2 are independent, then V_0~T(0,5,2.5). Given this, we will consider three cases where we look at different quantities of interest and have differing knowledge of the inner workings of the system. The cases are: 1) we know the component is an adder and our quantity of interest is V_0; 2) we know nothing about the component and our quantities of interest are V_0 and V_1; 3) we know the component is an adder and our quantities of interest are V_0 and V_1.

Case 1. The first case, where we know the component is an adder and our quantity of interest is V_0, is represented in Figure 3.

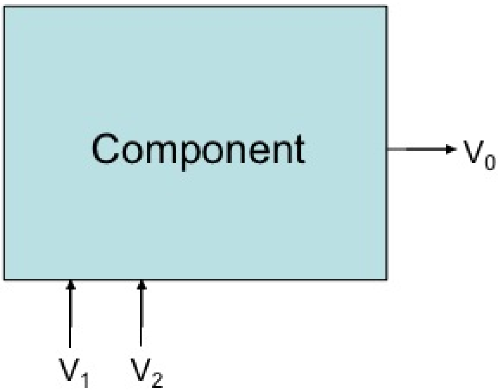


Figure 3: System for Case 1

Here we know that V_1 and V_2 are independent and uniformly distributed between 0 and 5 volts, thus, V_0~T(0,5,2.5)V. Since V_0 is our quantity of interest, we estimate the complexity of the system by estimating the entropy of V_0. We do this using Equation 5 and Equation 1, which yields a system complexity of 4 nats, where nats are the units of entropy when the natural logarithm is used in the entropy calculation.

Case 2. The second case, where we know nothing about the component and our quantities of interest are V_0 and V_1, is represented in Figure 4. Here, since we know nothing about the component, we assume that V_1, V_2, and V_0 are all independent. We know that V_1 and V_2 are independent and uniformly distributed between 0 and 5 volts, and we assume that V_0~T(0,5,2.5)V, which perhaps we ascertained through experimentation. Since our quantities of interest are V_0 and V_1, we must estimate the entropy of their joint distribution. This results in a system complexity of 20 nats.

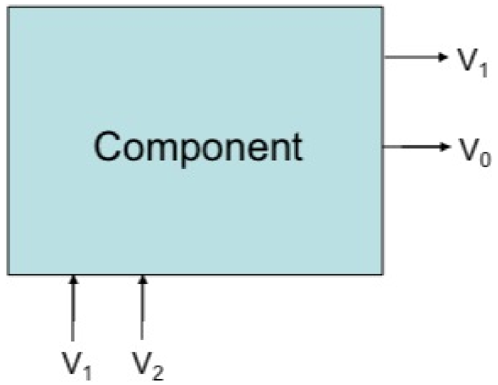


Figure 4: System for Case 2

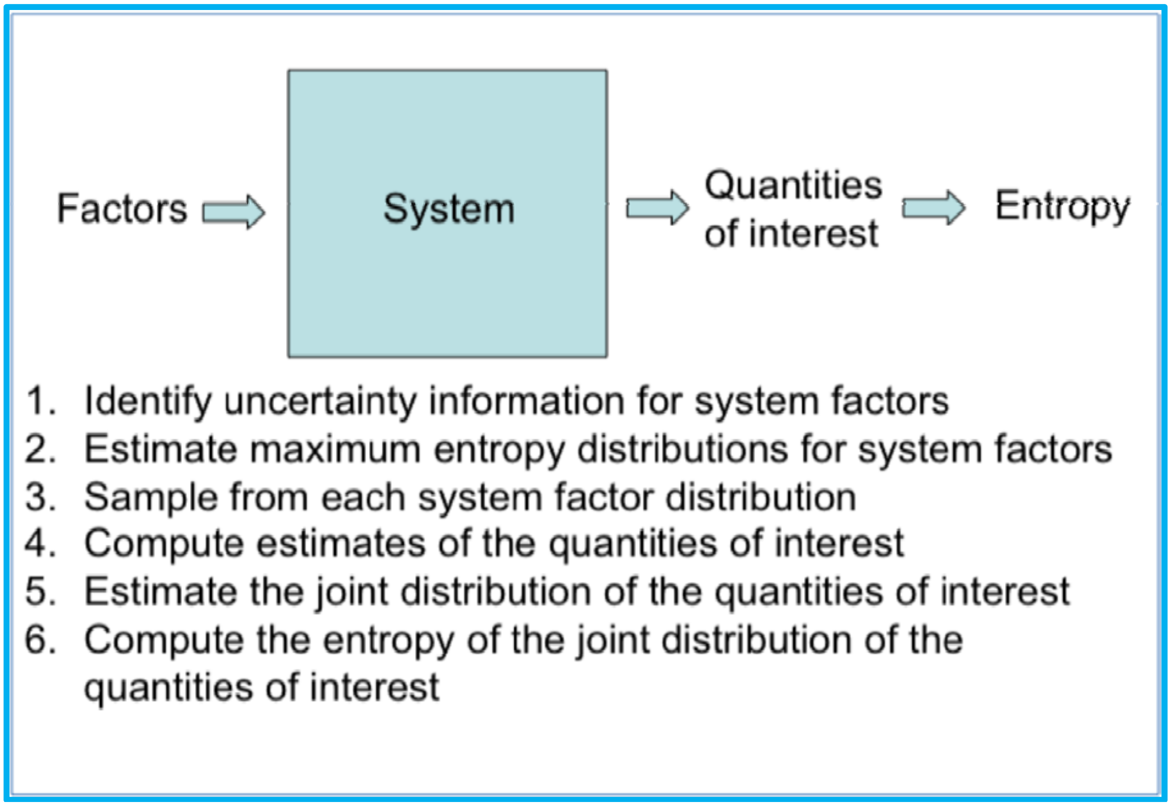
Case 3. The third case, where we know the component is an adder and our quantities of interest are V_0 and V_1, is represented as in Figure 4, however now we know the component is an adder circuit. Here we know that V_1 and V_2 are independent and uniformly distributed between 0 and 5 volts, thus, V_0~T(0,5,2.5)V. Since our quantities of interest are V_0 and V_1, we must estimate the entropy of their joint distribution. In this case, the system complexity is 12 nats. This result and the results from the other cases are discussed in the following subsection.

3.1.5 Adder Circuit Example Discussion. In the case where we knew everything about the system and only cared about V_0 (case 1), our complexity was the smallest. When we knew everything about the system but cared about V_0 and V_1 (case 3), our complexity was larger than when we cared about just V_0 alone. When we knew the least information about our system (case 2), our complexity was the largest. Clearly the choice of the quantities of interest impacts the complexity of the system. Also, the more we know about a component, the smaller its entropy and hence complexity in terms of the quantities of interest. This is the result of the following property of information entropy

$$h(X,Y) \leq h(X) + h(Y), \tag{8}$$

where X and Y are random variables. Here we have equality only when X and Y are independent. In the adder case, between case 2 and case 3 we learned the component was an adder, and thus learned about some dependence between V_0 and V_1. This dependence reduced the entropy in their joint distribution, and thus the complexity in terms of those quantities of interest.

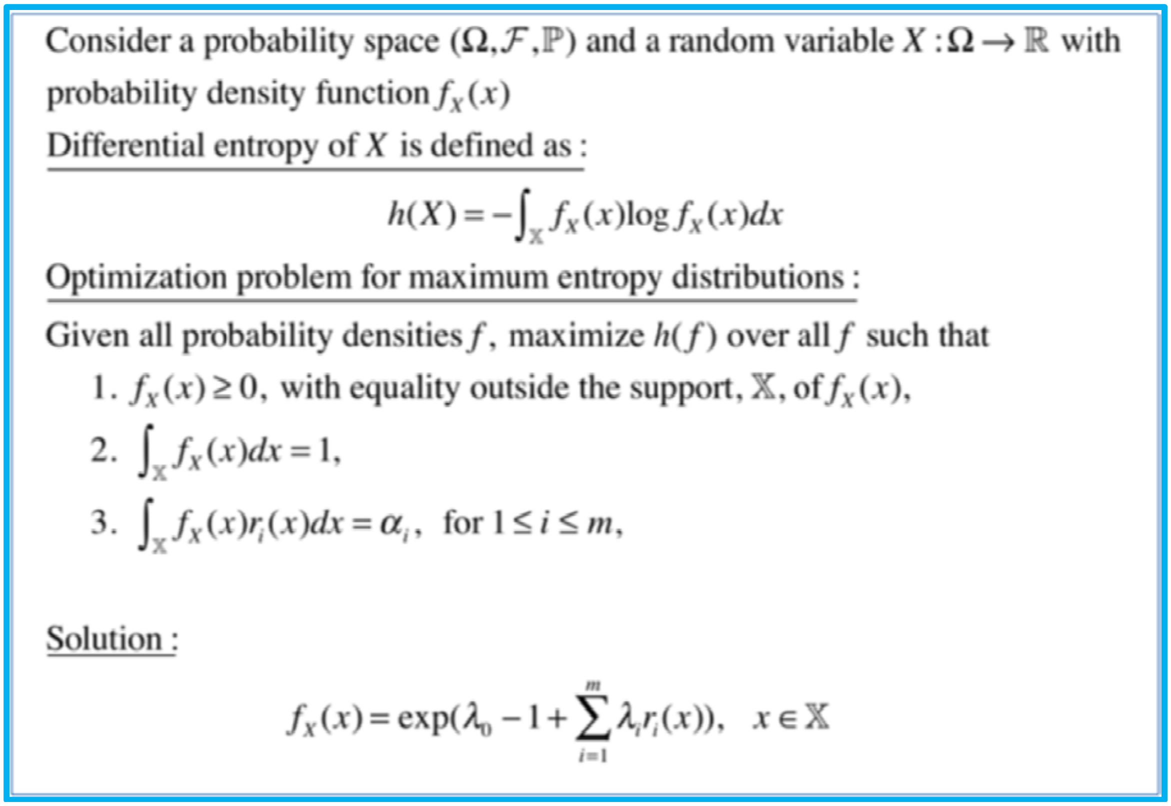
3.1.6 Computation of the Complexity Metric. The computation of our complexity metric follows the procedure shown in Figure 5.



1. Identify uncertainty information for system factors
2. Estimate maximum entropy distributions for system factors
3. Sample from each system factor distribution
4. Compute estimates of the quantities of interest
5. Estimate the joint distribution of the quantities of interest
6. Compute the entropy of the joint distribution of the quantities of interest

Figure 5: Complexity Metric Computation Procedure

The overall process is to identify the quantities of interest for a particular system and then identify all of the factors of the system that impact the quantities of interest. Once this is complete, uncertainty information is identified for each system factor. This information may come from historical data, expert opinion, simulation exercises, etc. With this information, maximum entropy distributions are computed following Cover and Thomas [1]. The general form of the maximum entropy optimization problem is presented in Figure 6.



Consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and a random variable $X : \Omega \to \mathbb{R}$ with probability density function $f_X(x)$

Differential entropy of X is defined as :

$$h(X) = -\int_{\mathbb{X}} f_X(x) \log f_X(x) dx$$

Optimization problem for maximum entropy distributions :

Given all probability densities f, maximize $h(f)$ over all f such that

1. $f_X(x) \geq 0$, with equality outside the support, $\mathbb{X}$, of $f_X(x)$,
2. $\int_{\mathbb{X}} f_X(x) dx = 1$,
3. $\int_{\mathbb{X}} f_X(x) r_i(x) dx = \alpha_i$, for $1 \leq i \leq m$,

Solution :

$$f_X(x) = \exp(\lambda_0 - 1 + \sum_{i=1}^{m} \lambda_i r_i(x)), \quad x \in \mathbb{X}$$

Figure 6: Maximum Entropy Optimization Problem Solution

Once maximum entropy distributions have been constructed, the next step is sampling from each system factor distribution. For this we first sample uniformly using quasirandom sequences and them employ an empirical inverse cumulative distribution function (CDF) method for converting uniform samples into samples from the maximum entropy distributions. Figure 7 presents a comparison of samples from two uniform random variables, factor 1 and factor 2, using a standard pseudorandom number approach (left) and a quasirandom sequence (right). For high dimensional problems, quasirandom sequences tend to have better convergence properties than pseudorandom numbers. The error convergence rate for Monte Carlo integration using pseudorandom numbers is $O(1/\sqrt{N})$, whereas it is $O((\log N)^s / N)$ for quasirandom numbers.

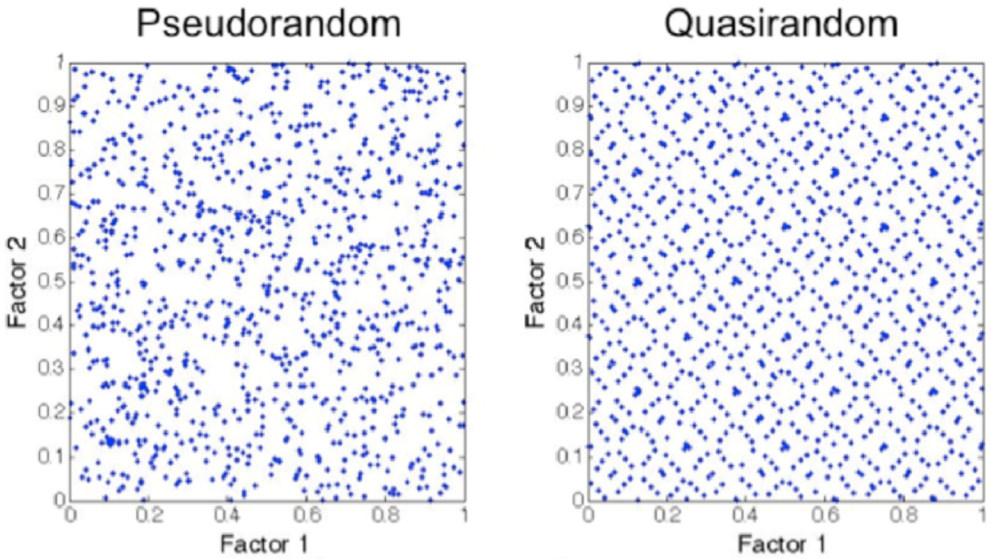


Figure 7: Comparison of Uniform Pseudorandom and Quasirandom Samples

Once uniform samples have been obtained, the empirical CDF method shown in Figure 8 is used to obtain samples of the maximum entropy distributions. In this figure, u is a a uniform sample and the resulting x is a sample form the desired distribution. The CDF for the maximum entropy distributions is arrived at empirically by discretizing the maximum entropy probability density functions.

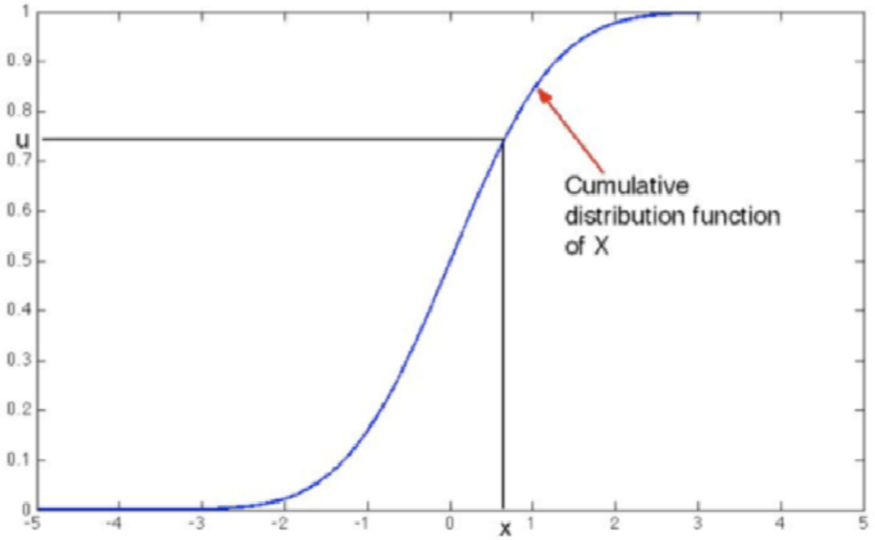


Figure 8: Cumulative Distribution Function Method for Sample Generation

Once samples of the system factors have been obtained, Monte Carlo simulation may be used to obtain samples of the quantities of interest. Following this, we estimate the joint distribution of the quantities of interest using kernel density estimation techniques as shown for one-dimension in Figure 9, where the red dashed lines are kernel density functions that are used to estimated the probability density function given by the blue line.

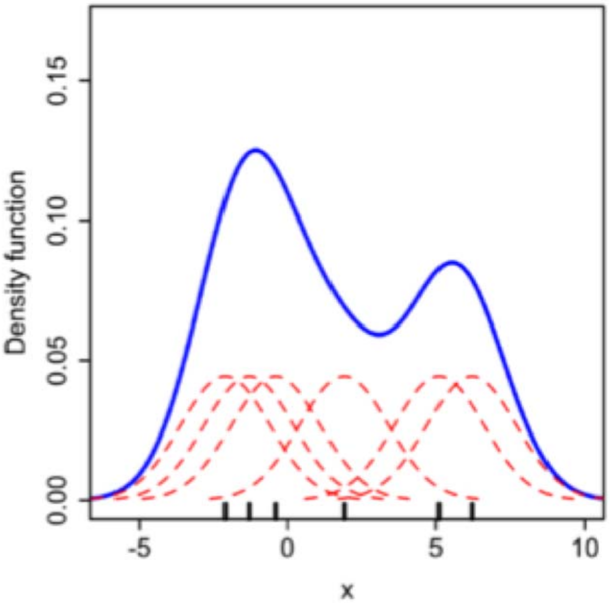


Figure 9: Notional Kernel Density Estimation of a Probability Density Function

Once we have estimated the joint distribution of our quantities of interest, we compute the entropy using the method shown in Figure 10.

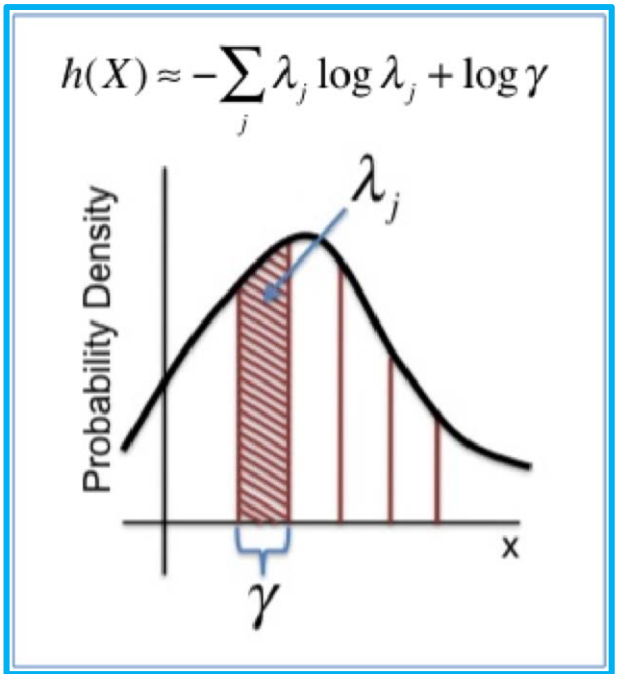

$h(X) \approx -\sum_j \lambda_j \log \lambda_j + \log \gamma$
Probability Density
λ_j
x
γ

Figure 10: Entropy Estimation Method

Once the entropy has been estimated the exponential of it is taken to yield an estimate of the system complexity.

3.2 Resource Allocation

Having defined complexity and a metric for estimating it for a given system, the question arises regarding what to do when system complexity is too high. That is, our system is deemed too complex as currently designed. To deal with this situation, we have developed a resource allocation procedure based on complexity sensitivity indices that quantitatively identify key factor, component, subsystem, discipline, etc. contributors to system complexity. In the following subsections we present our sensitivity analysis procedure, a computational method for computing the indices, and a similar sensitivity analysis methodology aimed at meeting reliability requirements.

3.2.1 Sensitivity Analysis for Resource Allocation. Consider Figure 5, where we have a set of factors that interact with a system to impact a set of quantities of interest. We estimate complexity from the joint distribution of these quantities of interest. If our complexity is to large, we would like to know how to best allocate our resources to effectively and efficiently reduce that complexity. Thus, we would like to know which of the factors that impact the complexity are the key drivers. For this we have developed a sensitivity index that enables a rigorous, quantitative ranking of the factors based on the expected amount of system complexity that could be reduced if everything about a given factor could be known. The derivation of our sensitivity indices is as follows. Let Q be the quantities of interest and let X_i for $i \in \{1,\ldots,k\}$ be the factors of our system that impact Q. Here all components of Q and all X_i are random variables. Then the complexity of our system is defined as

$$C_Q = \exp(h(Q)). \tag{9}$$

The expected complexity of the system that would be removed if factor X_i were known is given as

$$\exp(h(Q)) - \exp(h(Q \mid X_i)). \tag{10}$$

We may normalize this by the initial system complexity to arrive at

$$\frac{\exp(h(Q)) - \exp(h(Q \mid X_i))}{\exp(h(Q))}, \tag{11}$$

which is now an index giving the proportion of system complexity that is expected to be reduced if factor X_i is known. Since $h(Q) - h(Q \mid X_i)$ is the mutual information, $I(Q; X_i)$, between Q and X_i, we may rearrange Equation 3 and define our complexity sensitivity indices for resource allocation as

$$\eta_i = 1 - \exp(-I(X_i; Q)). \tag{12}$$

The range of these indices is the interval [0,1].

3.2.2 Computational Approach to Computing Sensitivity Indices. The computation of the complexity sensitivity indices follows closely that of the complexity computation given in Section 3. The additional requirement here is the computation of the mutual information, which we compute according to

$$I(Q; X_i) = h(Q) + h(X_i) - h(Q, X_i), \tag{13}$$

where the entropies are computed as shown in Figure 10.

3.2.3 Sensitivity Analysis for Reliability Constraint Resource Allocation. Once a system design is of sufficiently low complexity, it may still be the case that several reliability type constraints be violated. That is, constraints of the type shown in Figure 11, where probability of an event (here the event $T \geq 9$) is greater than some allowable threshold ε.

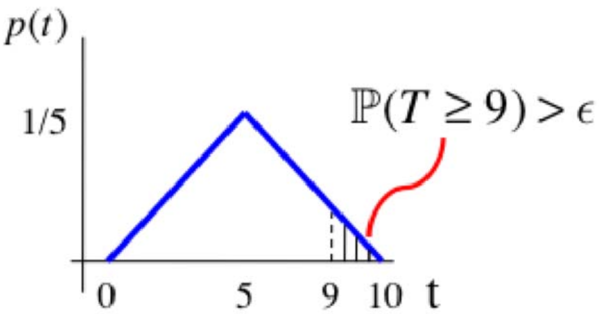


Figure 11: Reliability Constraint Example

For such situations, it is useful to have some means of determining how to allocate resources effectively and efficiently to ensure that the system will meet these types of requirements. For this we have developed a regional sensitivity analysis capability based on Kullback-Liebler divergence, which is given as

$$D_{KL}(P \| F) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{f(x)} dx, \tag{14}$$

where P and F are arbitrary random variables and the support of F contains the support of P. The objective of the method is to identify where to focus efforts (both design and research-based) to reduce failure probabilities. What follows is a simple example of our regional sensitivity analysis methodology.

Consider two independent uniform random variables $X_1 \sim U[0,1]$ and $X_2 \sim U[0,9]$, and their sum $T = X_1 + X_2$, where T is then triangularly distributed. Let's say our "system" fails if $T > 9$ and that we desire our probability of failure to be less than some ε. Then we compute the Kullback-Liebler divergence between the unconditional input distributions and the input distributions conditioned on the failure region. These distributions are shown in Figure 12, where the blue distributions are unconditional and the red distributions are conditioned on the failure event.

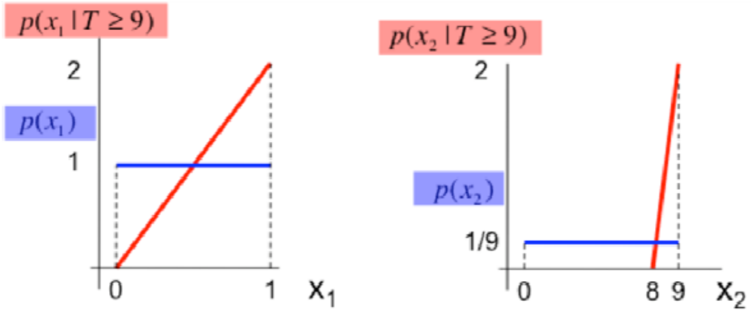


Figure 12: Unconditional and Conditional Distributions

The result is a divergence of 2.4 for X_2 and of 0.2 for X_1. This indicates that X_2 is the main contributor to the failure region, which is as expected given the problem setup.

4.0 RESULTS AND DISCUSSION

In this section we demonstrate our methodology on an RLC circuit and a notional hybrid infantry fighting vehicle.

4.1 RLC Circuit Demonstration

To demonstrate our complexity metric and complexity sensitivity analysis, we consider an RLC circuit design for a high pass filter. The RLC circuit to be analyzed is presented in Figure 13. The circuit contains R, L, and C components with specified uncertainty information. A classical circuit model is given in Figure 14 and the frequency response as calculated with classical circuit analysis is presented in Figure 15.

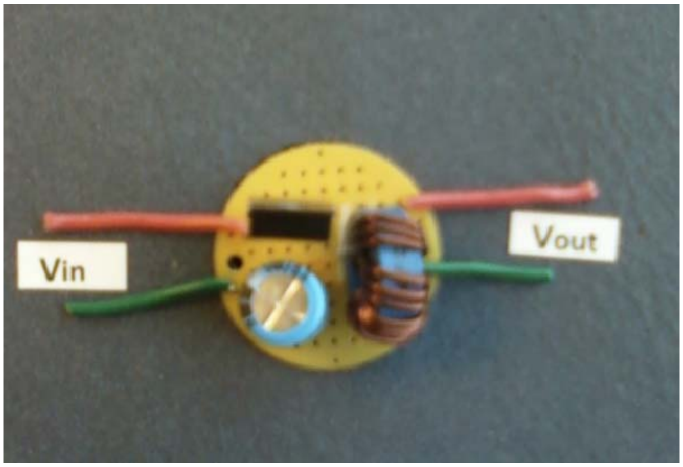


Figure 13: RLC Circuit

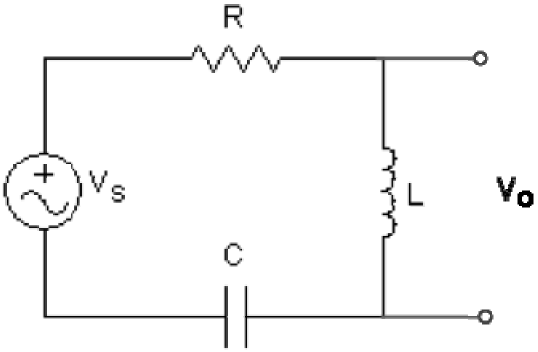


Figure 14: Circuit Model

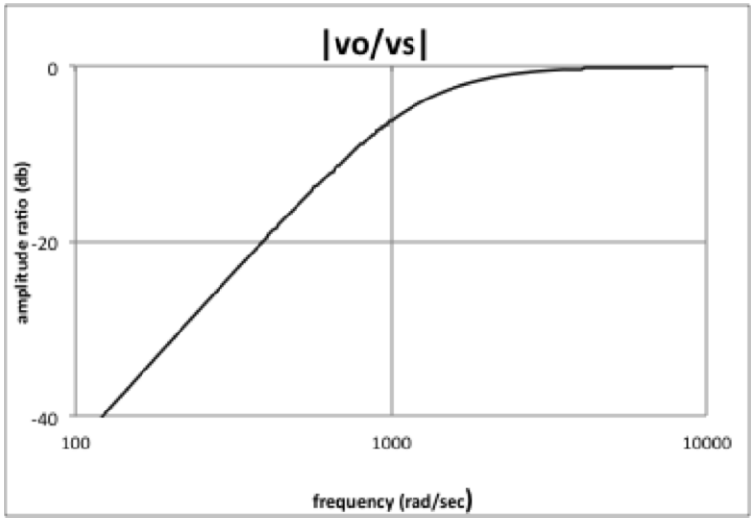
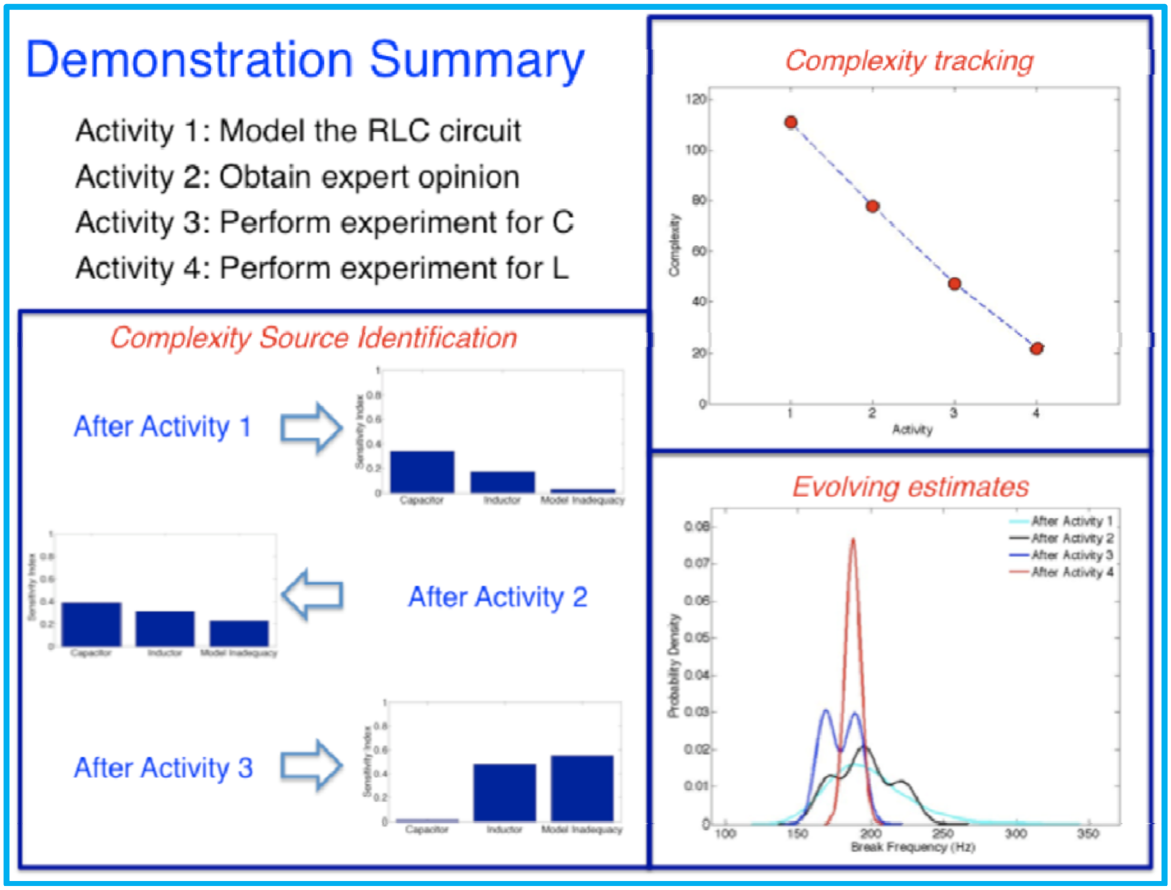


Figure 15: Frequency response

For demonstration purposes, we consider the break frequency of this circuit as a quantity of interest that we would like to estimate. A summary of the RLC circuit demonstration results is presented in Figure 16. The problem begins with Activity 1, which is the estimation of the complexity of the system with respect to the quantity of interest. We do this by modeling the circuit using classical circuit analysis and estimate the complexity with the computational methodology presented in Section 3.5. The result is a system complexity estimate shown for Activity 1 in the Complexity Tracking plot of Figure 16. Assuming this complexity is too high, we use our complexity sensitivity analysis methodology to determine the key contributors to the complexity among the capacitance, inductance, and model inadequacy (model inadequacy refers to the fact that no model is perfect and is discussed in more detail in Section 8). The capacitance and inductance were found to be key contributors. This information was used to determine that expert advice should be ascertained regarding how uncertainty information regarding capacitors and inductors is presented in specification data. This expert opinion elicitation is the second activity. Once the expert opinion information is obtained, the factor distributions are updated and the complexity estimation and sensitivity analysis calculation are repeated. The reduction in complexity from Activity 1 to Activity 2 is shown in the Complexity Tracking plot and the sensitivity analysis results are shown in the Complexity Source Identification portion of Figure 16. These results reveal that the capacitor is the key contributor to complexity and thus Activity 3 involves an experiment to reveal a more accurate estimate of the capacitance distribution. With this new distribution the complexity and sensitivity analysis calculates are again repeated. The results of this activity are again shown in Figure 16. The sensitivity analysis results clearly identifies the inductance as the key contributor to complexity at this point, thus Activity 4 is an experiment to determine a better estimate of the inductance distribution. We again repeat the complexity calculation. The total result is that these 4 activities reduce the complexity of the system through the evolution of the distribution of the quantity of interest, which is shown in the Evolving Estimates portion of the figure. This complexity was systematically reduced using our sensitivity analysis for guiding resource allocation decisions. It should be noted here that this demonstration was notional in the sense that all numerical values were selected for demonstration purposes. No physical experiments were actually undertaken.



Demonstration Summary

Activity 1: Model the RLC circuit

Activity 2: Obtain expert opinion

Activity 3: Perform experiment for C

Activity 4: Perform experiment for L

Figure 16: RLC Demonstration Summary

4.2 Infantry Fighting Vehicle Demonstration

Figure 17 presents the work breakdown structure for an infantry fighting vehicle design problem we are developing to demonstrate our complexity and sensitivity analysis methodologies upon. The demonstration is to be completed over the next two months.

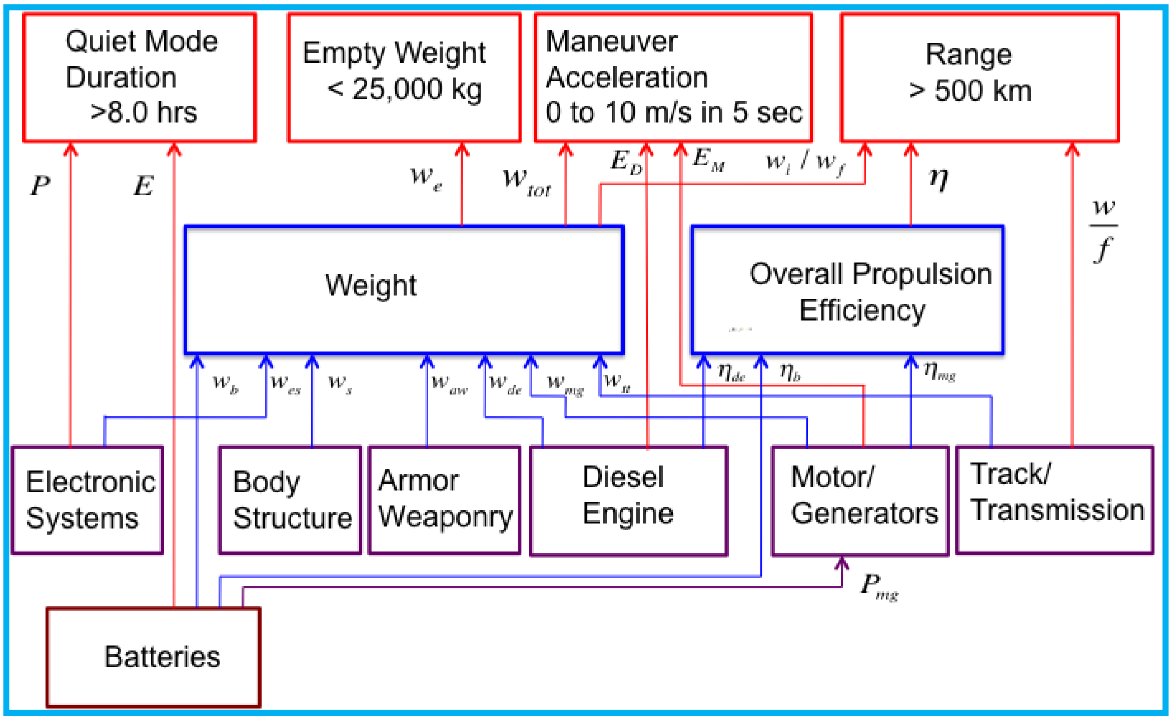


Figure 17: Simplified Work Break Down for an Infantry Fighting Vehicle

4.3 Model Inadequacy

Mathematical models of reality implemented in computer codes contain many different sources of uncertainty. Among these are parameter uncertainty, residual variability, parametric variability, observation error, code uncertainty, and model discrepancy [2]. Following Reference 2, parameter uncertainty relates to uncertainty associated with the values of model inputs; residual variability relates to the variation of a particular process outcome even when the conditions of that process are fully specified, parametric variability results when certain inputs require more detail than is desired (or possible) and are thus left unspecified in the model; observation error involves the use of actual observations in a model calibration process; code uncertainty results when a code is so complex or computationally involved that it may not be possible to execute the code at every possible input configuration of interest, thus there is some additional uncertainty related to regions of the input space that have not been interrogated; and model discrepancy relates to the fact that no model is perfect, and thus some aspects of reality may have been omitted, improperly modeled, or contain unrealistic assumptions. All forms of uncertainty must be quantified to properly estimate complexity. Here we consider model inadequacy (also referred to as model discrepancy), which is an often overlooked source of uncertainty in simulation-based design.

To make the discussion of how we are considering quantifying model discrepancy clear, consider a model, M, that estimates a vector of outputs $\mathbf{z}(\mathbf{d})$, where $\mathbf{d}$ is a vector of design variables. The outputs of $\mathbf{z}(\mathbf{d})$ will have uncertainty associated with them due to the discrepancy of M. We represent this discrepancy as F(**d**), which is a random field on the design space. Our estimate of the true vector of outputs can then be written as y(**d**) = z(**d**) + F(**d**), which is also a random field. To make this more concrete, consider a model with a single design variable and a single output. Figure 18 then shows a Gaussian process representation of model discrepancy of this model as a function of a design variable. The light blue represents the two standard deviation confidence interval of the model discrepancy term and the dark blue line represents the mean value of the model discrepancy. In this notional example we are assumed the model is unbiased, hence the zero value for the mean line. At any given design point, say d* for this one-dimensional example, the model for which this discrepancy is quantified estimates an output of interest z*. At the point d* on the figure, the model discrepancy is represented by a normal distribution shown in red. Since we are using a Gaussian process representation of model discrepancy, at any point in the design space the model discrepancy is normally distributed, which is consistent with a maximum entropy perspective of uncertainty quantification.

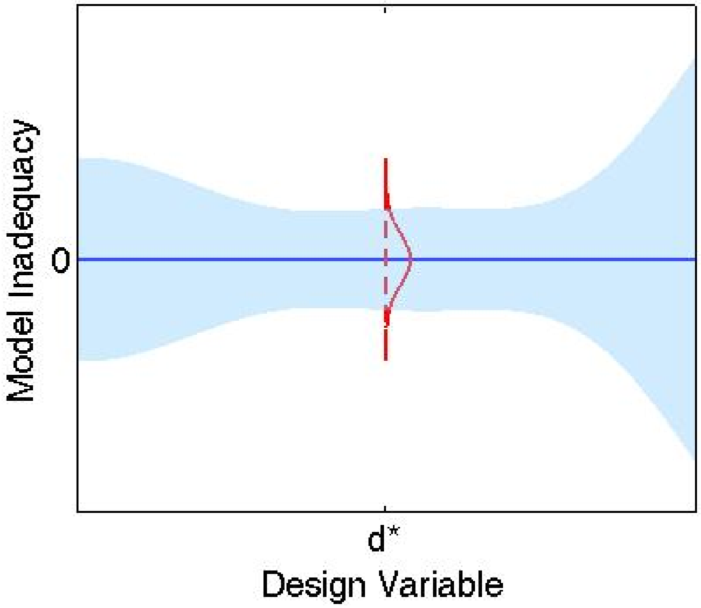


Figure 18: Notional Representation of Model Inadequacy

The term notional has been used here to imply that we have not consulted experts or historical data at this point to formally quantify the model discrepancy. Instead, we have focused on how we will represent that discrepancy in a general way. As noted, this

Gaussian process representation is a natural extension of the maximum entropy representation in the sense that at any point in the design space we have a maximum entropy distribution of the model discrepancy based on information coming from experts, historical data, or information coming from correlation with other nearby design points. We plan to quantify the model discrepancy of several models being used by the Vanderbilt team in the META-X project.

4.4 Implementation of our Methods within Analytical Target Cascading

The next step to applying our work in a design setting is to implement our methodology within a design flow. To that end, we consider the application of methods within a simulation-based multidisciplinary design methodology such as analytical target cascading as the first step.

Analytical target cascading (ATC) is a methodology for the design of multi-disciplinary systems in which the design process is cast as a hierarchical multi-level optimization problem [3,4]. It is particularly useful during the early stages of design in order to partition the overall design into its respective systems, subsystems, and components, identify the connections between the various pieces, and coordinate the exchange of design targets and parameters across the hierarchical structure. Figure 19 shows an example multi-level ATC hierarchy. Each block in the ATC framework corresponds to a physical entity in the vehicle design, but is also an abstract representation of one or more system, subsystem, or component models that are used to solve optimization subproblems at the respective tiers.

The ATC procedure can be summarized in four steps [5]:

1. Specify overall vehicle mission targets
2. Propagate vehicle targets to subsystem and component sub-targets
3. Design vehicle systems, subsystems and components to achieve their respective sub-targets
4. Verify that the resulting design meets overall vehicle mission targets

Atop the design hierarchy is the vehicle level (also known as the supersystem), where the overall vehicle mission targets are enumerated in Step 1; these targets are specified in the vector **T**. Achieving a feasible design for a vehicle that meets all such targets (e.g., range, weight, maneuverability) and satisfies the associated constraints (e.g., cost, schedule) is the goal of the vehicle-level optimization problem. In Step 2, the overall vehicle targets are passed to the second tier as targets for the system-level optimization subproblems; for the simplified ATC schematic shown in Figure 19, this propagation of targets to the two systems is captured by the vectors Rs1U and Rs2U. In Step 3, the system-level analysis models use the local variables xs1 and xs2, as well as target response values Rss1L and Rss2L computed by the subsystem-level analysis models, to solve the system-level optimization subproblem. The computed response values are then cascaded downward as targets Rss1U and Rss2U to the subsystem-level optimization problems, as well as back up to the vehicle level via the vectors Rs1L and Rs2L in order to carry out Step 4: ensuring that the system-level responses are consistent with the overall mission targets T (Figure 19). This “cascade down” and “rebalance up” procedure is characteristic of the ATC decomposition approach. It is an iterative process that is repeated at each block in the ATC hierarchy until a feasible vehicle design is achieved. In addition to the cross-tier exchange of target values through the vectors RL and RU, the various systems, subsystems, and components of the ATC hierarchy may also be connected to one another across multiple levels through linking variables yL and yU, or laterally along one level via linking variables y (Figure 19). In this way, ATC maintains a bidirectional information cascade that coordinates tasks across the different parts of the distributed design problem, and ensures consistency throughout the design process.

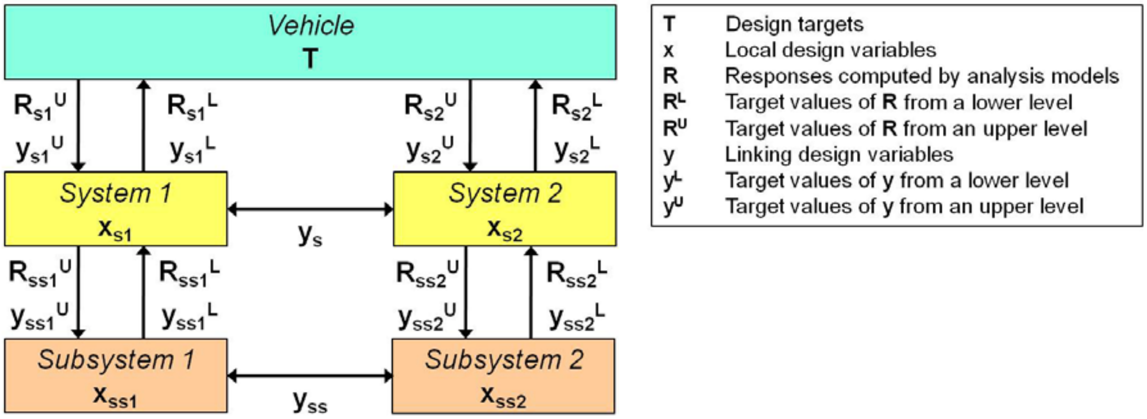


Figure 19: Data Flow for an ATC Design Problem (adapted from [5])

4.4.1 Applications, Advantages, and Limitations. To date, analytical target cascading has been applied to a number of vehicle design problems; for example, for the conceptual design of an aircraft [6], a sport utility vehicle [7], and an advanced technology heavy truck [8]. Beyond the scope of vehicle design, other supersystems studied using the ATC approach have included engineering enterprises [9] and thermal and HVAC design in buildings [10].

There are a number of advantages to the ATC approach for the design of complex systems. The primary motivation for ATC is that the efficiency of a design process greatly improves when the various required design tasks can be performed concurrently [4,5]. In order to achieve this goal, it becomes desirable to decompose a design problem as much as possible into subproblems that can be solved in parallel. A key limitation of concurrent design, however, is that properly accounting for all intersystem interactions can become an intractable problem, leading to increased programmatic risk and cost and schedule overruns. Analytical target cascading, however, provides a rigorous method to coordinate multiple optimization subproblems and exchange design information in systematic manner. By leveraging the idea of distributed design, ATC allows for a complex design process to be partitioned into manageable segments, design tasks to be delegated across multiple teams, and computational throughput to be greatly increased. The main outcome is a significant reduction in design cycle time. In addition, by ensuring at each iteration that the overall design targets are met at all levels of the ATC hierarchy, late-stage design iterations can be avoided, thus saving valuable time and resources [4]. Furthermore, because the target propagation and target matching procedures are integrated into one step (in contrast to, for example, the “all-at-once” approach shown on the left side of Figure 20), the total number of design iterations can also be reduced.

Despite numerous advantages, however, there are nevertheless several limitations associated with ATC. The first class of limitations relates to problem formulation. The multi-level ATC framework lends itself nicely to designs that can be decomposed into a hierarchical structure; failing that, there is limited guidance on how to formulate a problem within the ATC context. This is especially true of complex designs with numerous interfaces and interactions, such that it is difficult to isolate individual systems, subsystems, and components and arrange them in a logical hierarchy. Assuming the design problem can be assimilated into the ATC architecture, a second challenge arises in that appropriate analysis models must be available for each subproblem. This problem has been identified as the main difficulty of ATC, and is particularly observable during the early stages of design, when analysis models are likely to be unavailable [5]. Furthermore, there exists a distinct tradeoff between model fidelity and resource expenditure: sophisticated analysis tools are expensive to develop and to compute, whereas “back-of-the-envelope” calculations may not be sufficient to capture the complexity of the design and the interactions therein [4]. Clearly, it is not adequate to simply have a set of analysis models at hand; selecting models of the appropriate levels of fidelity is an issue that must also be addressed. Finally, in order to leverage the main asset of ATC to coordinate distributed design tasks, great care must also be taken to identify the appropriate responses, linking variables, and local variables that map the design subproblems to their respective analysis models; failure to do so would result in incompatibilities between the abstract models and the physical design that can easily propagate to the various parts of the ATC hierarchy and lead to inaccuracies in the overall design.

A second category of limitations relates to the computational efficiency of ATC. Even though ATC takes advantage of concurrent design and parallel computing, efficiency can still become an issue due to the added cost of coordinating analyses based on different disciplines [5]. Another concern is whether the “cascade down - rebalance up” approach for ATC guarantees convergence to a feasible design. However, this concern has been mostly unfounded in various applications, as the ability of ATC to converge to an optimal solution has been demonstrated through several example problems [4]. Furthermore, it has also been shown that several nested ATC coordination strategies are provably convergent [11]. In fact, the convergence properties of the ATC decomposition approach have made it particularly attractive for system design, especially when the all-at-once design strategy is not available as an option [11].

4.4.2 Application to IFV Design Problem. We are currently working to apply the ATC methodology to a challenge problem involving the design of a hybrid infantry fighting vehicle (IFV) (Figure 17). The main objective of the challenge problem is to demonstrate the validity of ATC to the development of a stochastic process model for the design of complex systems. Furthermore, a second objective is to use the hybrid IFV problem to illustrate how complexity-based sensitivity analysis may be employed at each design iteration to identify key contributors to complexity at every level of the ATC hierarchy. For this challenge problem, we have identified four vehicle-level targets:

1. The hybrid IFV must operate in quiet mode for at least 8 hours
2. The empty weight of the hybrid IFV must not exceed 25,000 kg
3. The hybrid IFV must achieve a maneuver acceleration of 0 to 10 m/s in 5 seconds
4. The hybrid IFV must have a range of at least 500 km

The decomposition of a simplified hybrid IFV design problem into the ATC framework is shown on the left side of Figure 20. The vector T is composed of the four vehicle-level targets listed above, which are cascaded down to two systems and seven subsystems. Each block in the hierarchy represents an analysis model that solves an optimization subproblem to match targets passed from an upper level. The exchange of state variable targets and computed responses between the tiers is shown through the bidirectional arrows, which are color-coded according to their relation to one of the four vehicle-level targets.

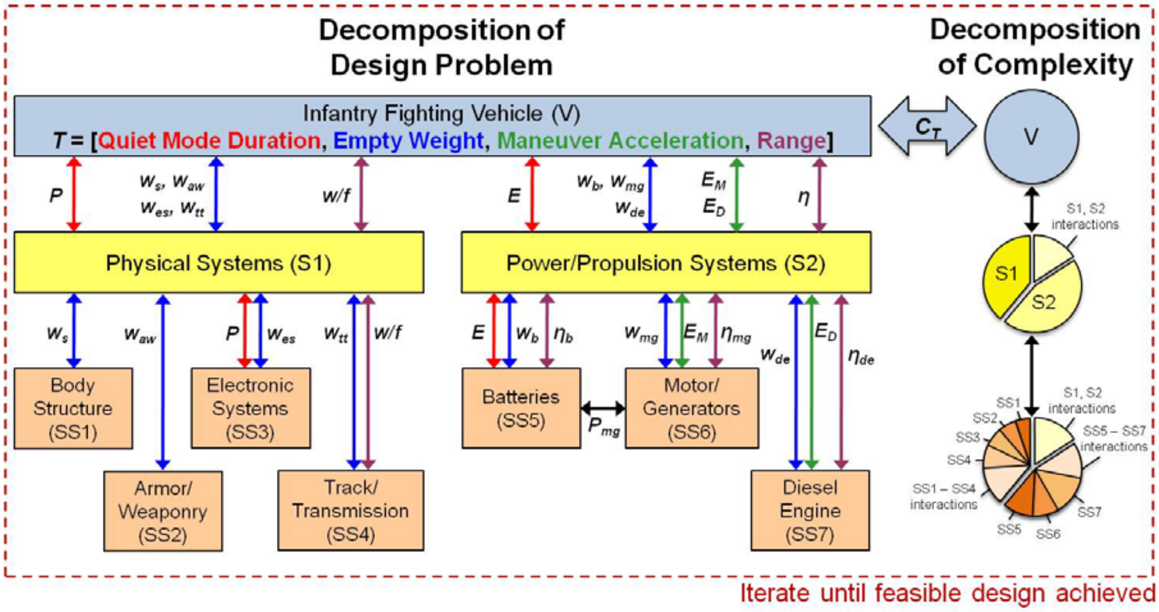


Figure 20: Decomposition of IFV Design Problem using ATC

On the right side of Figure 20 is shown the notational complexity decomposition for the IFV problem. The quantities of interest in this design are the vehicle-level targets **T**. At the top tier, we compute the vehicle-level complexity $\mathbf{C_T}$ with respect to those quantities of interest. Using complexity-based sensitivity analysis, we then decompose the vehicle-level complexity into contributions from the two systems **S1** and **S2**, as well as their interactions. This procedure can then be repeated to uncover how complexity associated with each of the seven subsystems and their interactions contributes to complexity at the system-level and the vehicle-level. We believe that this capability to quantify and apportion complexity provides valuable insights to inform design decisions and guide resource allocation, as well as delivers a rigorous method to manage uncertainty and reduce risk in system design.

5.0 CONCLUSION

We have presented here our work on stochastic process decision methods for cyber-physical design. The emphasis was on our complexity metric development in terms of a definition, a quantification process, as well as on our sensitivity analysis development, which is to be used for informing efforts aimed at complexity reduction.

The methods we have developed are very promising in terms of establishing a rigorous procedure for quantifying and subsequently reducing system complexity. However, there are a number of further tasks that must be completed prior to making our methodology generally applicable in the META context. The tasks, which we intend to complete during the META-X effort are as follows:

Task: Further develop complexity metric estimation techniques to efficiently handle multiple quantities of interest and multivariate interactions. We will approach this by incorporating advanced filtering techniques to provide samples for efficient high dimensional integration while maintaining the probabilistic dependence structure among system factors.

Task: Develop a decomposition-based approach to uncertainty propagation through a complex system. We will approach this by using advanced filtering techniques to rapidly coordinate information among subsystems.

Task: Develop a model inadequacy quantification procedure that incorporates historical data and expert opinion. The method should be capable of sequential updates for situations where new data becomes available. We will approach this by creating a Gaussian process model of model inadequacy incorporating techniques from trust region methods from optimization.

Task: Develop a resource allocation procedure based on complexity-based sensitivity analyses and a resource expenditure model. We will approach this by incorporating techniques from statistical decision-theory.

Task: Create open source versions of our complexity metric and complexity-based sensitivity index estimators. We will approach this by considering implementing our methods in C, Python, R, or some other open source language.

6.0 REFERENCES

1. Cover, T. and Joy, A., Elements of information theory, Wiley series in telecommunications Wiley, New York, NY, 1991.

2. Kennedy, M. and O'Hagan, A, "Bayesian Calibration of Computer Models," *J. R. Statist. Soc. B*, Vol. 63, Part 3, pp. 425-464, 2001.

3. Michelena, N., Kim, H.M., and Papalambros, P. "A System Partitioning and Optimization Approach to Target Cascading," *International Conference on Engineering Design*. Aug. 24-26, 1999.

4. Kim, H.M., Michelena, N.F., Papalambros, P.Y., and Jiang, T. "Target Cascading in Optimal System Design," *Transactions of the ASME*, Vol. 125, pp. 474-480, 2003.

5. Kim, H.M. *Target Cascading in Optimal System Design*, Doctoral Dissertation, University of Michigan, 2001.

6. Allison, J., Walsh, D., Kokkolaras, M., Papalambros, P.Y., and Cartmell, M. "Analytical Target Cascading in Aircraft Design," *44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2006-1325, Jan. 9-12, 2006.

7. Kim, H.M., Rideout, D.G., Papalambros, P.Y., and Stein, J.L. "Analytical Target Cascading in Automotive Vehicle Design," *Proceedings of ASME DETC 2001*, DETC2001/DAC-21079, Sept. 9-12, 2001.

8. Louca, L.S., Kokkolaras, M., Delagrammatikas, G.J., Michelena, N.F., Filipi, Z.S., Papalambros, P.Y., and Assanis, D.N. "Analytical Target Cascading for the Design of an Advanced Technology Heavy Truck," *Proceedings of IMECE 2002*, IMECE2002-32860, Nov. 17-22, 2002.

9. Cooper, A.B., Georgiopoulos, P., Kim, H.M., and Papalambros, P.Y. "Analytical Target Setting: An Enterprise Context in Optimal Product Design," *Transactions of the ASME*, Vol. 128, pp. 4-13, 2006.

10. Choudhary, R., Malkawi, A., and Papalambros, P.Y. "Analytic Target Cascading in Simulation-Based Building Design," *Automation in Construction*, Vol. 14, 2005, pp. 551-568.

11. Michelena, N., Park, H., and Papalambros, P.Y. "Convergence Properties of Analytical Target Cascading," *AIAA Journal*, Vol. 41, No. 5, pp. 897-905, 2003.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

ATC	analytical target cascading
CDF	cumulative distribution function
$C(Q)$	complexity of a system with quantity of interest Q
$D(P\|\|F)$	Kullback-Liebler divergence from P to F
$H(Y)$	information entropy of random variable Y
IFV	infantry fighting vehicle
$I(X;Q)$	mutual information between X and Q
$N(\mu,\sigma^2)$	normal distribution with mean μ and variance σ^2
P	probability measure
Q	quantity of interest
RLC	resistance, inductance, capacitance
T	vector of targets
$T(a,b,c)$	triangular distribution with minimum a, maximum b, and mode c
$U[a,b]$	uniform distribution with minimum a, maximum b
V	voltage
$\mathbf{d}$	design variable vector
$h(Q)$	differential entropy of the distribution of the quantity of interest Q
$p(y)$	probability mass function of random variable Y
$y(\mathbf{d})$	true output
$z(\mathbf{d})$	model output
F	sigma field
Ω	sample space
η	complexity-based sensitivity index
λ	probability mass
μ	mean
σ^2	variance