# Exploiting Multidimensional Design of Experiments and Kriging Methods: An Application to a Satellite Radar System Tradespace and Orbital Transfer Vehicle Tradespace

Daniel O. Fulcoly*, Nirav B. Shah†, Adam M. Ross‡, and Donna H. Rhodes§
*Massachusetts Institute of Technology, Cambridge, MA, 02139*

**A major benefit to performing tradespace exploration as early as possible and on as broad of a scale as possible is avoiding premature fixation on potentially non-optimal point designs. As the size of a tradespace grows in both the number of variables and the levels for each variable, the number of alternatives, or allowed combinations of various levels of design variables, grows at a combinatorial rate. The Expedited Tradespace Approximation Method (ETAM) was developed in response to this challenge. ETAM leverages intelligent subsampling and interpolation methods, including design of experiments and Kriging Methods, to generate acceptable data for a large tradespace, using fewer computational resources than applying a performance model to every design point. ETAM is applied to two case studies to demonstrate its accuracy and explore its potential for computational savings.**

## Nomenclature

| | | |
|---|---|---|
| $N$ | = | Number of variables |
| $L_i$ | = | Number of levels enumerated for the i$^{th}$ variable |
| $S$ | = | Krigable subset |
| $T$ | = | Training set |
| $K$ | = | Kriging set |
| $H$ | = | Number of single attribute utility functions |
| $A$ | = | Vector of $H$ attributes |
| $U(A)$ | = | Multi-Attribute Utility function |
| $u_h(A_h)$ | = | $h^{th}$ Single-Attribute Utility function |
| $j_h$ | = | $h^{th}$ "small $j$" weight |
| $J$ | = | Normalization constant for a set of small $j$s |
| $M_p$ | = | Propulsion system mass (kg) |
| $m_{p0}$ | = | Propulsion system base mass (kg) |
| $m_{pf}$ | = | Propulsion system mass fraction |
| $M_f$ | = | Fuel mass (kg) |
| $M_b$ | = | Vehicle bus mass (kg) |
| $M_m$ | = | Manipulator mass (kg) |
| $m_{bf}$ | = | Bus mass fraction |
| $DfE$ | = | Design for Evolvability mass penalty (%) |
| $M_d$ | = | Vehicle dry mass (kg) |
| $M_w$ | = | Vehicle wet mass (kg) |
| $c_d$ | = | Dry mass cost ($/kg) |

* Research Assistant, Systems Engineering Advancement Research Initiative, MIT E38-555, 77 Massachusetts Ave.
† Doctoral Student, Aeronautics and Astronautics, MIT E38-555, 77 Massachusetts Ave.
‡ Research Scientist, Engineering Systems Division, MIT E38-574, 77 Massachusetts Ave., AIAA Senior Member.
§ Principal Research Scientist, Engineering Systems Division, MIT E38-572, 77 Massachusetts Ave., AIAA Member.

$c_w$ = Wet mass cost ($/kg)
$C$ = Cost ($)
$\Delta v$ = Delta V (m/s)
$I_{sp}$ = Specific impulse (s)
$D$ = Determinant
$TDF$ = Time Dilation Factor
$n$ = Number of design-epoch pairs in the full tradespace
$t$ = Execution time of the performance model for a single design-epoch pair
$k$ = Ratio of training set size to Krigable subset size
$E$ = Upfront time needed to integrate the tradespace into the ETAM model

# I. Introduction

This paper will cover the motivation for, development and implementation of, and application of the Expedited Tradespace Approximation Method (ETAM) to two case studies. ETAM is a method that leverages design of experiments and interpolation (using Kriging methods) to generate acceptable data for a large tradespace using fewer computational resources than applying a performance model to every design point. For clarity, many of the terms that will be used in this paper are defined up front. A *design* is the collection of particular choices for each design level of each design variable that collectively define a specific design. *Design variables* are the variables, which are within the control of the designer, that are needed to specify a design. Ideally design variables drive value metrics of interest to stakeholders. *Levels* are the allowed values for a design variable or an epoch variable (defined below), to take. The *valid range* is the minimum and maximum values allowed for a design or epoch variable based on physical constraints or limitations of the model. *Enumerated* refers to a design level in the set of values spanning a valid range. For example, if the valid range is [1, 10], the set {1, 4, 7, 10} might be the enumerated levels. These are the possible design choices in the tradespace. *Sampled* refers to the set of particular levels within the enumerated set that will be evaluated using the model (i.e. simulated). An *epoch* is a period of fixed contexts and needs. *Epoch variables* are the variables that encapsulate key system-exogenous uncertainties, which together define an epoch. A *design-epoch pair* is a combination of a single design and single epoch. *Attributes* are stakeholder-defined criteria that reflect how well stakeholder-defined objectives are met for a system. A *simulated* design-epoch pairing is one whose attributes are calculated via a performance model. Simulated designs are those in the sampled subset of the enumerated tradespace. An *unsimulated* design-epoch pairing is one whose attributes will be generated by means other than a performance model. Unsimulated designs are those in the enumerated tradespace but not the sampled subset.

## A. Motivation

A major benefit to performing tradespace exploration as early as possible and on as broad of a scale as possible is avoiding premature fixation on potentially non-optimal point designs. As the size of a tradespace grows in both the number of variables and the levels for each variable, the number of alternatives, or allowed combinations of various levels of design variables, grows at a combinatorial rate. The equation for the full tradespace, with $N$ variables having $L_i$ levels, is

$$Tradespace\_Size = \prod_{i=1}^{N} L_i$$

This growth can be seen in Fig. 1 for a constant $L_i$. In many cases, the calculation of attributes, which enables analysis, for a given design takes a significant amount of computation time. Since analyzing alternatives in terms of attributes requires models that require finite computation time, the total time to fully simulate a tradespace grows in a combinatorial manner along with the size of the tradespace; one could easily enumerate a tradespace that takes prohibitively long to simulate. This is based on the traditional two-step tradespace generation process: (1) enumerate the tradespace and (2) simulate the tradespace. Kriging is a technique for inferring the value of a random variable, in this case an attribute level of an unsimulated design, based on the values of that variable at nearby locations, or simulated design points. When coupled with intelligent sampling techniques (i.e. design of experiments (DOE)), Kriging can be a valuable tool in the tradespace exploration phase of the systems engineering process since it allows for a tradespace to be approximated based on intelligent subsampling. ETAM then changes the tradespace process to a four-step one: (1) enumerate the tradespace, (2) sample the tradespaces, (3) simulate the sampled tradespace, (4) "fill in" the unsampled tradespace using a method of interpolation (e.g. Kriging). For previously simulated tradespaces, ETAM can leverage the existing data to "fill in the blanks" for unsimulated designs.

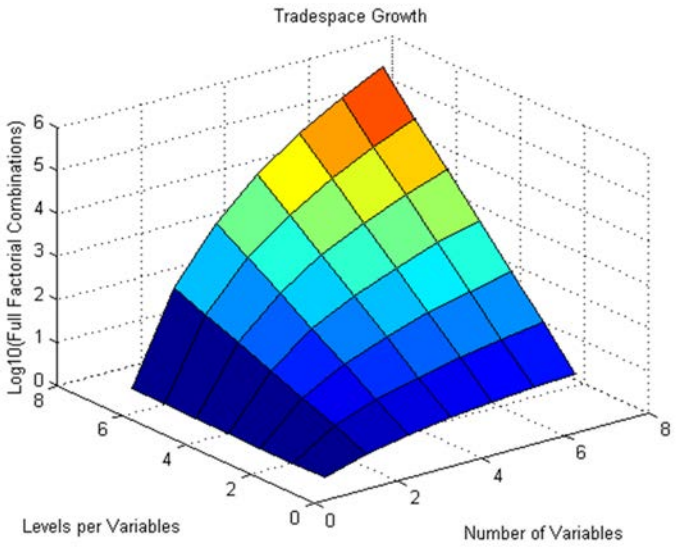


**Figure 1. Tradespace growth assuming all combinations allowed.**

### B. Overview of ETAM

There are three major components to ETAM: isolating an appropriate data set from the full enumeration, sampling to select a training set (e.g. using a DOE approach), and interpolating (e.g. through Kriging) the missing points. Each of these components must be carefully executed in order to ensure that no invalid assumptions are made, inappropriate data is not passed to the Kriging algorithm, and excess computation time is not used. Fig. 2 shows how DOE and Kriging are used to approximate non-simulated Design-Epoch pairs for an appropriately isolated data set.

Modularity is an important property of ETAM. At each arrow, a number of processes can be used. There are many types of DOE that can be used at the first arrow, it is possible that multiple performance models exist for the second arrow, and multiple interpolation routines can be used at the third arrow in place of Kriging.

### C. Existing Methods

Large tradespaces are not a new burden on systems engineers, and therefore the problem of evaluating them has been addressed before[1]. ETAM can be seen as a particular implementation of these approaches in the context of tradespace exploration. Some methods suggest using a combination of model fidelities, a combination of artificial intelligence-based search and optimization engines, or even the combination of DOE and optimization[2,3,4]

The Architecture Enumeration and Evaluation (AEE) process developed by the United Technologies Research Center uses a set of design rules to identify the sparse set of valid architectures in a combinatorially large architecture space. The AEE builds up to complete architecture choices by gradually defining more and more of the architecture. At each partial architecture step, the design rules can be used to validate (to the next step in building the architecture) or eliminate an entire subtree of architectures. This process ensures that each architecture is considered while not actually having to physically evaluate every architecture in the search for all feasible choices[5]. This method is very effective when design rules are known a priori, but ETAM serves to populate a set of designs separate from any value statement. ETAM and AEE could work effectively together, with AEE filtering which designs are valid and handing the valid designs over to ETAM for evaluation.

The method proposed in Ref. 2 calls for the use of a lower fidelity model to examine a full-factorial tradespace and then use of a higher fidelity model to evaluate areas of interest identified by the first model. The authors propose that their method is a "versatile modeling framework which (1) allows a rapid assessment of the broader architectural tradespace, (2) evaluates high-level metrics for comparing competing architectures, (3) models "second-order" couplings between design parameters, and (4) identifies favorable classes of architectures." While this method has

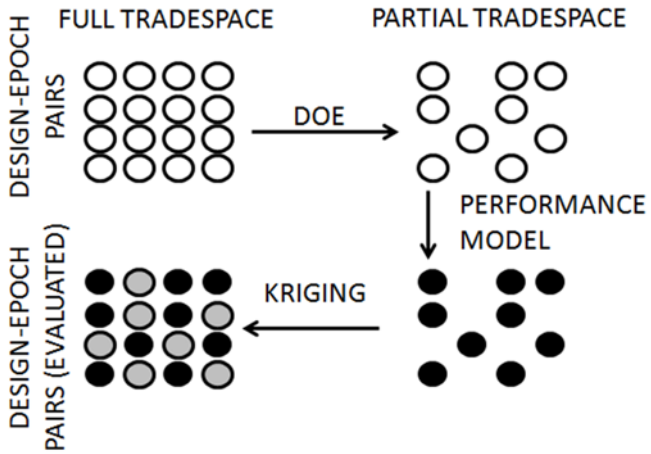


**Figure 2. Overview of the ETAM.**

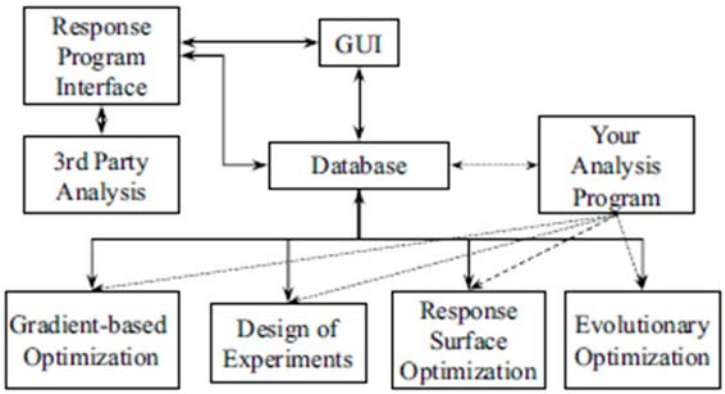


**Figure 3. Overview of general VisualDOC structure[4].**

value when multiple models of varying fidelity are available, ETAM seeks to save computational time in the presence of a single model.

The System Platform for Integrated Design in Real-Time (SPIDR) analyzes large tradespaces by combining artificial intelligence-based search and an optimization engine3. SPIDR is a "constraint-based design synthesis engine" that does not require a utility function or collapsing multiple performance dimensions. For a set of design variables, architecture rules, mission requirements, and optimization metrics, SPIDR returns a "best" design. A single answer such as a "best" design is not always as valuable as having a tradespace to explore in search of many answers. ETAM, unlike SPIDR, does not seek to provide a solution, only to populate a tradespace with approximated attribute values. SPIDR combines exploration and optimization, whereas ETAM serves only to aid in exploration.

The Applied Research Laboratory (ARL) at Penn State developed a method called the ARL tradespace visualizer (ATSV) that provides a multitude of tools for visualizing, navigating, and evaluating complex tradespaces. One of the tools in ATSV involves choosing an "attractor" in the tradespace to simulate new designs "nearby." The ATSV engine uses a differential evolution (DE) algorithm and a response surface model (when quicker than querying the full model) to populate points near the attractor with improving fitness (Normalized Euclidian distance from the attractor). Alternatively, the attractor can be replaced with a preference function to use with DE to find points of increasing preference[6]. The ATSV provides value in a different way than ETAM, specifically in that it combines fitness with the tradespace population problem. ETAM does not seek to include a fitness function. Much of ASTV's value is derived from its excellent visualization ability, a function that ETAM does not include.

VisualDOC is a commercially available tool that, similar to SPIDR, couples exploration, analysis, and optimization[4]. Another focus of VisualDOC is a user interface that allows a user without knowledge of DOE or optimization to still leverage those concepts when exploring a tradespace. An aspect of VisualDOC that ETAM hopes to emulate is the high degree of modularity, as seen in Fig. 3. While in theory VisualDOC could adopt Kriging as one of its methods, at this time the documentation does not list it as a method in use. VisalDOC's modularity could potentially lead to a specific instance of it representing ETAM, but at this time Kriging is not one of the interpolation models available. A long-term goal for ETAM is to have "plug'n'play" modularity with respect to the type of DOE used and the type of interpolation.

Ref. 7 proposes the interactive multiscale-nested clustering and aggregation (iMSNCA) framework to support tradespace exploration for multidimensional data, shown in Fig. 4. The iMSNCA framework "puts design activities in the forefront and emphasizes the role of computational tools in supporting such activities by considering the characteristics of design data." The iMSNCA framework involves a human in the loop between all major steps to aid in steps such as data downsizing, data clustering, viewing graphs, controlling aggregations, and controlling views. While this framework appears to be a valuable tool, it falls into the same category as many of the other methods explored thus far in that it couples tradespace exploration with tradespace simulation.

Ref. 8 previously explored the use of Kriging as a tool in multidisciplinary design optimization. Simpson cites many reasons for using Kriging in place of traditional response surface models, such as the wide range of spatial

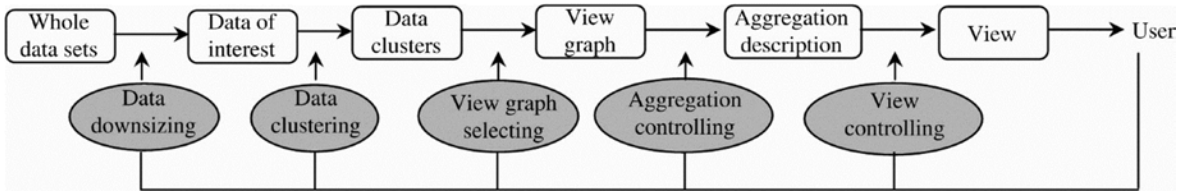


**Figure 4. Overview of iMSNCA Framework[7].**

correlation functions that can be used, the ability to choose between "honoring the data" or "smoothing the data," and the ability to approximate linear and nonlinear functions equally well[8]. Ref. 8 also addressed barriers (circa 2001) to implementing Kriging, most notably the computational complexity with respect to readily available software. This limitation was echoed in Ref. 9, which added the lack of guidance with respect to choosing the appropriate form of the Kriging model as a limitation. By means of application to a small (3 variable) case study, Kriging was viable, slightly more accurate than response surface models, while only adding on minimal additional computational expense[8]. Many topics for future research on Kriging methods are given by Ref. 10 in its review of the state of Kriging meta-modeling: the development of Kriging software, proofs of performance, rules of thumb for application and selection of pilot designs (training set in this paper's notation), stopping rules based on measures of accuracy, application to practical random simulations, consideration of sensitivity analysis for robust optimizations, consideration of multivariate outputs, exploration of the preservation of known simulation I/O properties (e.g. monotonicity), and the use or meta-models other than Kriging. The research in this paper addresses rules of thumb for selection of the training set (pilot designs) and application to multivariate outputs.

Exploring these existing methods reveals that there are many valuable methods for aiding tradespace exploration already in use. This does not mean that ETAM was created in vain; ETAM still plays a unique role. Unlike many of the methods explored, ETAM decouples the processes of tradespace generation and tradespace exploration. Methods like VisualDOC, ATSV, and SPIDR are built around the idea of optimization; the "best" design is often an output. On the other hand, ETAM is a front end to exploration that creates data that might not have been otherwise available, leaving the specific method of analysis up to the analyst. ETAM builds on the existing body of knowledge for applying Kriging to design exploration.

## II. ETAM Implementation

ETAM can be described as a process with 3 steps:

1. Partition the enumerated design-epoch tradespace into appropriate sets for interpolation
2. Select an appropriate training set for each of the isolated data sets
3. Interpolate the remainder of the isolated data sets

A satellite radar case study data set will be used to help illustrate the first two steps for clarity. The Satellite Radar System (SRS) case study contains 8 design variables and 6 epoch variables that specify a constellation of ground-observing radar satellites[11].

### A. Partitioning the Enumerated Tradespace

The dependent variables of interest in tradespace exploration are attributes and cost. These attributes are a function of design variables and epoch variables, the independent variables of tradespace exploration. Variables come in several types: ratio, ordinal, interval, and nominal. These types, as described by Ref. 12 and seen in Table 1, are classified as either Krigable or Non-Krigable depending on whether or not an interpolation scheme such as Kriging can be used to approximate attributes for differing levels of a variable of that type. Since the difference between independent variables is considered in Kriging, only interval and ratio variables are considered Krigable.

**Table 1. Variable types[12]**

| Variable Type | Explanation | Example | Classification |
|---|---|---|---|
| Ratio | The difference between two values is meaningful and there is a definition of zero. | Temperature (K or R), Power, Length, Time | Krigable |
| Interval | The difference between two values is meaningful. | Temperature (F or C), | Krigable |
| Ordinal | Order matters but not the difference between values. | Technology Readiness Level (TRL) | Non-Krigable |
| Nominal | Mutually exclusive, but not necessarily ordered (i.e. categorical). | Binary Variables (On/Off), ID #s | Non-Krigable |

The first step of ETAM, accordingly, is to segregate the design and epoch variables into Krigable and non-Krigable variables. This has been done for the SRS tradespace in Table 2 and Table 3. While all the epoch variables in this case happen to be non-Krigable, it is important to note that this will not always be the case. Any variables that are not inputs to the performance model should not be considered in ETAM, regardless of the variable type.

Including such variables as non-Krigable will unnecessarily increase computation time. Including such variables as Krigable will pass non-unique points to the Kriging routine, detracting from the quality of the fit.

**Table 2. Design variable list for SRS (Krigable variables are italicized)**

| Design Variable | Scale Type | Valid Range | Enumerated Levels | # Levels |
|---|---|---|---|---|
| *Altitude* | *Ratio* | *800 – 1500 [km]* | *800, 1200, 1500 [km]* | *3* |
| Constellation Configuration | Nominal | 1 – 8 [int] | 8 walker IDs | 8 |
| *Antenna Area* | *Ratio* | *10 – 100 [$m^2$]* | *10, 40, 100 [$m^2$]* | *3* |
| *Peak Transmit Power* | *Ratio* | *1.5 – 20 [kW]* | *1.5, 10, 20 [kW]* | *3* |
| *Radar Bandwidth* | *Ratio* | *0.5 – 2 [GHz]* | *0.5, 1, 2 [GHz]* | *3* |
| Communication Downlink | Nominal | Relay or Direct Downlink | Relay, Direct Downlink | 2 |
| Tactical Communication | Nominal | Able or Not Able | Able, Not Able | 2 |
| Maneuver Capability | Ordinal | 1 – 4 [x base fuel] | 1x, 2x, 4x Base Fuel | 3 |

**Table 3. Epoch variable list for SRS (Krigable variables are italicized)**

| Epoch Variable | Scale Type | Valid Range | Enumerated Levels | # Levels |
|---|---|---|---|---|
| Available Radar Technology | Ordinal | 9 – 3 [TRL] | Mature, Medium, Advanced | 3 |
| Communications Infrastructure | Nominal | 0 – 2 [int] | AFSCN, WGS + AFSCN, Third | 3 |
| Target Set | Nominal | 1 – 60 [int] | Lookup Table of 9 Regions and Ops Plans | 9 |
| Collaborative AISR Assets | Nominal | Available or Not Available | Available, Not Available | 2 |
| Threat Environment | Nominal | No Jamming or Hostile Jamming | No Jamming, Hostile Jamming | 2 |
| Mission Priorities | Nominal | SAR <GMTI, SAR=GMTI, SAR<GMTI | SAR < GMTI, SAR = GMTI, SAR < GMTI | 3 |

Once the variable space has been appropriately segregated, statistics about how many times the Kriging routine (second and third steps of ETAM) will be run can be calculated. For each attribute, the Kriging routine will need to be run for each combination of non-Krigable variables. For SRS, this is the product of the number of levels for all the non-Krigable epoch variables and design variables: 93,312 (1.2% of the full factorial design-epoch space). Taking into account that there are 12 attributes in the case study, this means the interpolation routine will actually be run 1,119,744 times.

### B. Selecting the Training Set using DOE

Once an appropriate data set (heretofore referred to as the Krigable subset) has been isolated for Kriging, the data set must be partitioned into a training set and Kriging set. The training set is the set of points whose attributes will be fully simulated and used to create the Kriging matrix, the matrix that is used in the interpolation of unsimulated points. The Kriging set is the set difference between the Krigable subset and the training set. The design and epoch variables of each point in the Kriging set will be used in conjunction with the Kriging matrix to interpolate the corresponding attributes. Using DOE to differentiate between the training set and the Kriging set allows for the intelligent sampling of the Krigable subset.

Remembering the way the Krigable subset was selected, we know that all designs in the set have the same levels for all non-Krigable variables (thus varying only along the Krigable variables). Assuming there are $N$ Krigable variables each having $L_n$ levels, the size of the Krigable subset will be

$$S = \prod_{i=1}^{N} L_i$$

The Krigable subset $S$ will be partitioned into a training set, $T$, and a Kriging set, $K$. Only the $T$ ($<S$) points in the training set, selected by DOE, will need to be simulated, leaving $K = S - T$ points to be interpolated by the Kriging routine. The particular method of DOE will determine which points in $S$ will be in $T$. The method of DOE used for SRS was a Box-Benkhen experimental design[13]. Box-Benkhen designs use three levels for each factor. If a Krigable variable has an odd number of levels, the end points and center point of that factor are the three levels. If a Krigable variable has an even number of levels there are three options: (1) create an additional point in the design space, (2) treat it as a non-Krigable variable, or (3) use a different type of DOE. Furthermore, a Box-Benkhen design is only valid for >2 factors. Table 4 shows the savings for different numbers of Krigable variables. For simplicity in calculating $S$, $T$, and $K$, Table 4 considers all Krigable variables to have only three levels. (It should be noted that Box-Benkhen designs are not the only experimental designs that can be used. The Box-Benkhen design was applicable for SRS because of its 3-level nature. Other designs, such as D-optimal designs, can be used in more general cases.)

**Table 4. Box-Benkhen savings for three-level variables**

| # Krigable Variables ($i$) | Points in $S$ ($S_i$) | Points in $T$ ($T_i$) | Points in $K$ ($K_i$) | Savings ($K/S$) |
|---|---|---|---|---|
| 3 | 27 | 13 | 14 | 52% |
| 4 | 81 | 25 | 56 | 69% |
| 5 | 243 | 41 | 202 | 83% |
| 6 | 729 | 49 | 680 | 93% |
| 7 | 2187 | 57 | 2130 | 97% |

Depending on how much the designer knows about the model and the relationships between Krigable variables and attributes, further savings might be possible. If a Krigable variable has no impact on an attribute the Kriging routine will receive 2 identical attribute values, one for each value of the non-impacting Krigable variable, for each combination of Krigable variables (excluding the non-impacting variable). These identical points are bad for two reasons: (1) it worsens the accuracy of the Kriging interpolation while (2) increasing the computation time on the order of $(T_i - T_{i-1})^2$. For this reason, it is beneficial to make a binary domain mapping matrix (DMM)[14] that identifies which Krigable variables affect which attributes. The binary DMM for SRS is shown in Table 5. Interestingly enough, none of the 12 attributes are a function of all four Krigable variables. It is important to note that the only design variables being considered here are the Krigable variables. Any of the 12 attributes might additionally be a function of the non-Krigable variables. An example of an attribute that is a function only of one variable is shown in Fig. 5 and one that is a function of three variables is shown in Fig. 6.

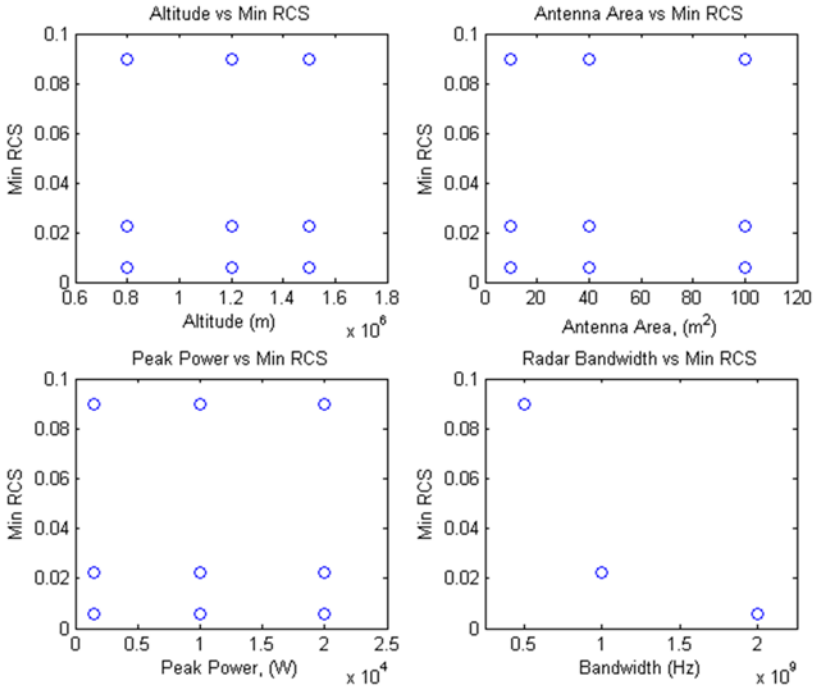


**Figure 5. Dependency diagram between *minimum RCS* and Krigable variables for Epoch 1.**

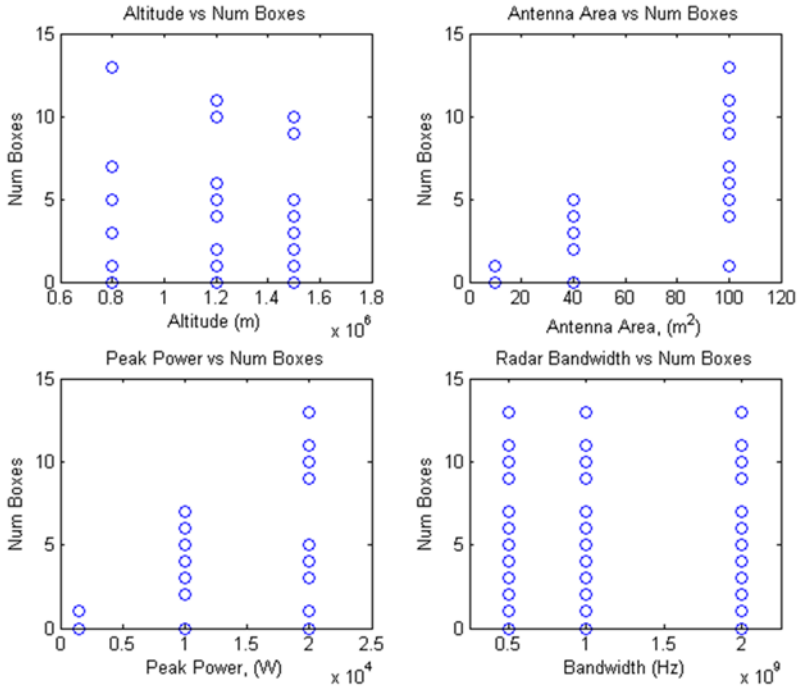


**Figure 6. Dependency diagram between *number of boxes* and Krigable variables for Epoch 1.**

**Table 5. Binary DMM for SRS ('X' indicated dependence)**

| | | Krigable Variables | | | | Number of Krigable Variables | Corresponding Training Set (see Table 6) |
|---|---|---|---|---|---|---|---|
| | | Orbital Altitude | Antenna Area | Band-width | Peak Power | | |
| Attributes | **Min RCS** | | | X | | 1 | A |
| | **Min Detect Velocity** | X | X | | | 2 | B |
| | **Number of Boxes** | X | X | | X | 3 | C |
| | **Target Acquisition Time** | X | X | | | 2 | B |
| | **Track Life** | X | X | | | 2 | B |
| | **Track Latency** | X | X | | X | 3 | C |
| | **Revisit Interval** | X | X | | | 2 | B |
| | **Image Latency** | X | X | | X | 3 | C |
| | **Resolution** | | | X | | 1 | A |
| | **Targets per Pass** | X | X | | X | 3 | C |
| | **Field of Regard** | X | X | | | 2 | B |
| | **Geolocation Accuracy** | X | | | | 1 | D |

For the four attributes dependent on three Krigable variables, *number of boxes*, *track latency*, *image latency*, and *targets per pass* (Training Set C), a three-level Box-Benkhen design was used. Since Box-Benkhen designs do not exist for one- and two-level designs ($T_1$ and $T_2$), a full-factorial design is used for the other eight attributes. Additionally, since the full-factorial experimental designs have three and nine treatments, respectively, they can be used without increasing the number of designs in the training set. Despite the fact that smaller training sets are used, the Kriging set is still determined by the attribute with the largest requisite training set (and accordingly the smallest Kriging set). In cases such as SRS where different size experimental designs are used, it is critical to ensure that each training set $T_i$ is a subset of the next largest training set $T_{i+1}$. Doing so ensures that each attribute will be accounted for in $K_{max(i)}$ since $K_i \subset K_{i-1} \subset K_{i-2} \ldots$. The training sets for each attributes are shown in Table 6.

**Table 6. Training sets for SRS**

| | Krigable Variable Level Index (1, 2, or 3) | | | | Training Set Membership (from Table 5) | | | |
|---|---|---|---|---|---|---|---|---|
| Point ID | Orbital Altitude | Antenna Area | Bandwidth | Peak Power | A ($T_1$) | B ($T_2$) | C ($T_3$) | D ($T_1$) |
| 1 | 1 | 1 | 1 | 2 | X | X | X | X |
| 2 | 1 | 3 | 1 | 2 | | X | X | |
| 3 | 3 | 1 | 3 | 2 | X | X | X | X |
| 4 | 3 | 3 | 1 | 2 | | X | X | |
| 5 | 1 | 2 | 1 | 1 | | X | X | |
| 6 | 1 | 2 | 1 | 3 | | | X | |
| 7 | 3 | 2 | 1 | 1 | | X | X | |
| 8 | 3 | 2 | 1 | 3 | | | X | |
| 9 | 2 | 1 | 2 | 1 | X | X | X | X |
| 10 | 2 | 1 | 1 | 3 | | | X | |
| 11 | 2 | 3 | 1 | 1 | | | X | |
| 12 | 2 | 3 | 1 | 3 | | X | X | |
| 13 | 2 | 2 | 1 | 2 | | X | X | |

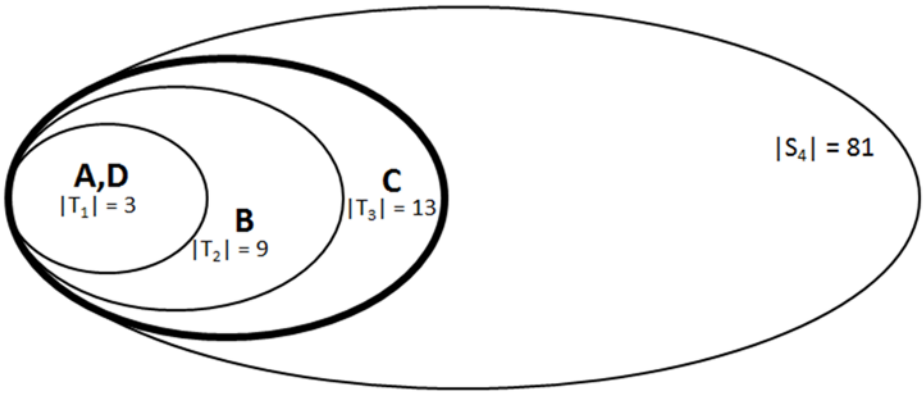


**Figure 7. Set relationships for Satellite Radar System.**

As seen in Table 6, training set *C* is $T_3$, training set *B* is $T_2$, and both training sets *A* and *D* are $T_1$. Note that training set *A* only needed to contain one point with each level for *bandwidth* and training set *D* only needed to contain one point with each level for *orbital altitude*. Since the attributes these training sets are being used for are a function of only one of the Krigable variables, it is not necessary to hold the other Krigable variables constant. A single three point set was sufficient for satisfying both training sets *A* and *D*. The table clearly shows $T_1 \subset T_2 \subset T_3$. The Kriging set then is $K_3$, or all designs (of a 4-factor, 3-level full factorial design) not seen in Table 6.

The relationships between training sets and the Krigable subset for SRS can be seen in Fig. 7. While not illustrated, $K_i = S_4 - T_i$ for *i* = 1, 2, or 3. This visualization reaffirms the set relationship $T_1 \subset T_2 \subset T_3$. Figure 7 only illustrates the set relationships and should not be interpreted as showing the "location" of points in training sets and Kriging sets.

### C. Setting up Kriging and Interpolating "Missing" Points

While many estimation models assume that a value can be approximated as the sum of a polynomial function and an independent, identically distributed random variable representing error, Kriging treats the error as a functional departure from the polynomial[15]. Kriging comes in three varieties: simple, ordinary, and universal. Simple Kriging is used when the variable being Kriged has a constant, known mean. Ordinary Kriging is used when the variable has an unknown but constant mean. Universal Kriging handles variables with unknown, varying means. ETAM assumes that the mean of the Kriged attributes is constant, albeit unknown, in order to use ordinary Kriging.

Ordinary Kriging assumes the value of an attribute at a point can be accounted for by a smooth deterministic function, referred to as *drift*, and random fluctuations from that drift, referred to as the *residual*[16]. In ordinary Kriging, the drift is the unknown mean of the attribute. To treat the drift as systematic (based on location), rather than random, a *variogram* is used. A variogram is a measure of special dependence between the points in the training set and the points to be interpolated. Despite not knowing the mean of the attribute, the Kriging "weights" can still be deduced through minimizing mean squared error. The Kriging weights are the coefficients that will be used to interpolate a value for the attribute at some point based on the value of the attribute at the points in the training set.

The specific computational implementation of ordinary Kriging takes the *1* x $T_i$ vector of simulated attribute values in a training set, the corresponding *i* x $T_i$ matrix of Krigable variable values, and the Krigable variable values of a point to be approximated, and outputs the Kriged approximation for that point[17]. In order to do this, a variogram must be specified. The variogram used was a power law model, chosen for its ability to generate good results while remaining computationally simple[16,18]. The power for the model was set at 1.5 based on recommendations from existing literature[18]. For Krigable variables that had ranges of several orders of magnitude, a logarithmic scaling operation was performed before and after the Kriging coefficients were calculated to improve accuracy[17].

## III. Application to Satellite Radar System

### A. Variable Handling

Recall that the variable handling and DOE setup for the Satellite Radar System (SRS) were described in Sections II.A and II.B to help demonstrate the method. None of the six epoch variables were Krigable and of the eight design variables, only four were Krigable: *orbital altitude, peak transmit power, bandwidth, and antenna area*. Of the 12

attributes, only four were a function of more than two of the Krigable variables: *number of boxes*, *tracking latency*, *image latency*, *and targets per pass*. Because the Box-Benkhen experimental design was not valid for attributes that were a function of one or two Krigable variables, those attributes could be estimated with 100% accuracy from a lookup table created with designs contained in the training set. Each Krigable set contained 81 points and each training set was 13 points.

### B. Results

The entire SRS tradespace, including 7776 unique designs and 864 unique epochs (108 of the 972 were invalid), was fully simulated for previous trade studies. In this study, the tradespace was subjected to the ETAM to make comparisons of both accuracy and computational time between the two methods (ETAM and full factorial simulation). In cases where the ETAM method called for a design to be simulated, the attributes for the given design-epoch pairing were retrieved from existing data. After the remainder of the attribute values in the tradespace were populated using Kriging, the results were compared to the original study.

**Table 7. Accuracy of Kriged variables**

| Attribute | Mean (Training Set Points) | E[\|Difference\|] | Std[\|Difference\|] | Average Error |
|---|---|---|---|---|
| Num Boxes | 12.674 | 2.574 | 6.666 | n/a |
| Track Latency | 52.545 | 2.935 | 7.599 | 2.81% |
| Image Latency | 393.059 | 9.386 | 16.218 | 2.01% |
| Targets per Pass | 23676.942 | 2548.697 | 9748.730 | 31.51% |

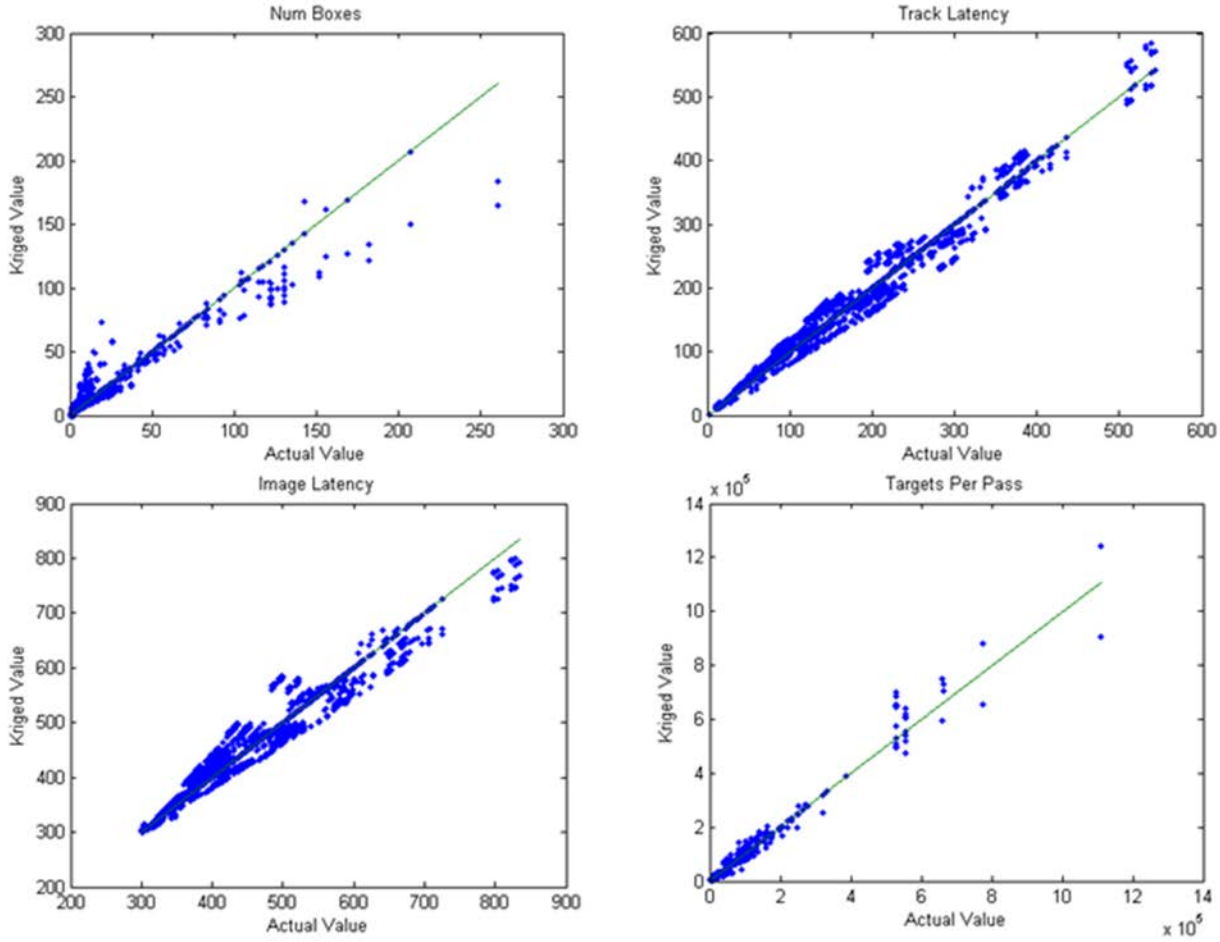


**Figure 8. Correlations of the attributes dependent on >2 Krigable variables.**

The Kriging results of non-simulated points only are seen in Table 7. The mean attribute vales for the training set points are given to put the expected value and standard deviation of the differences into context. For the eight attributes not seen in Table 7 Kriging was 100% accurate. This accuracy is due to the fact that each of these attributes is a function of less than three Krigable variables, as seen in Table 5. Recall that for attributes dependent on less than three Krigable variables, a full-factorial training set of either three or nine points is used. The Krigable variable levels of any interpolated point will match exactly one of the training points. In the case of SRS, the relationship between the design variables and attributes is deterministic, hence the exact match. Accordingly, the results for these eight variables should not be used to justify the accuracy of ETAM. The remaining four attributes range from very accurate (≈2% average error) to much less accurate (>30% average error). Division by zero precluded the calculation of expected error for the *number of boxes* since the range of the actual attribute values included zero.

The charts in Fig. 8 show the correlations between the interpolated values of attributes and the actual values of those attributes.

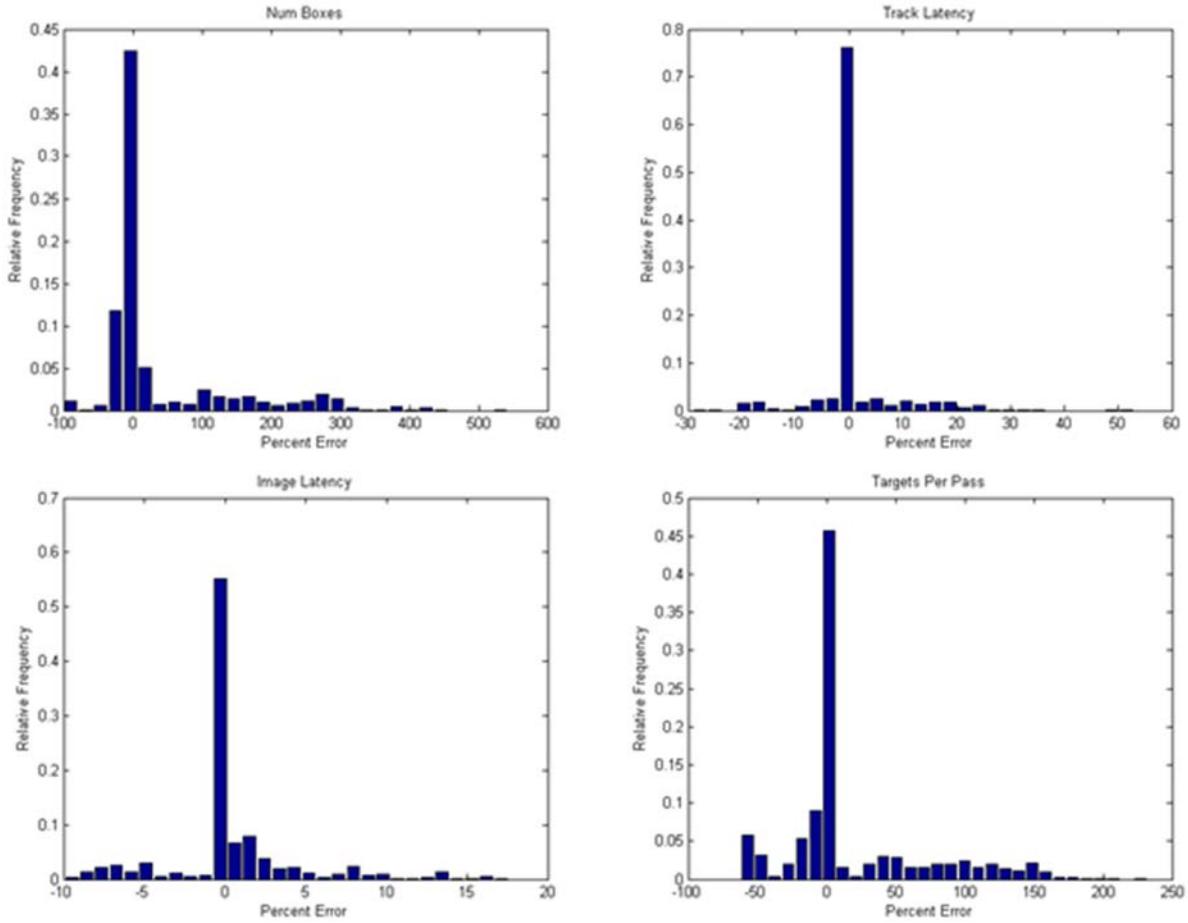


**Figure 9. Error distributions of the attributes dependent on >2 Krigable variables.**

Figure 9 shows the corresponding error distributions. For some attributes, the error seems to be distributed evenly about the actual value. For others, such as *number of boxes* (top-left), the error seems to be more systematic. For *number of boxes*, the Kriged values tend to be higher than the actual value for low actual values. For higher actual values, the Kriged value tends to underestimate the actual value. This due to the fact that there are very few (relative to the all the designs) values for the *number of boxes* attribute above 100 boxes. If the training set for that specific Krigable subset only captured one high value, an instance where the actual value should be high will always be underestimated; even though the design-epoch vector of the point to be interpolated might be closest to the high value point's design-epoch vector, the value will be discounted by the surrounding points of lesser values (discounted by distance). This pattern does not carry over to the *targets per pass* attribute, as seen by the

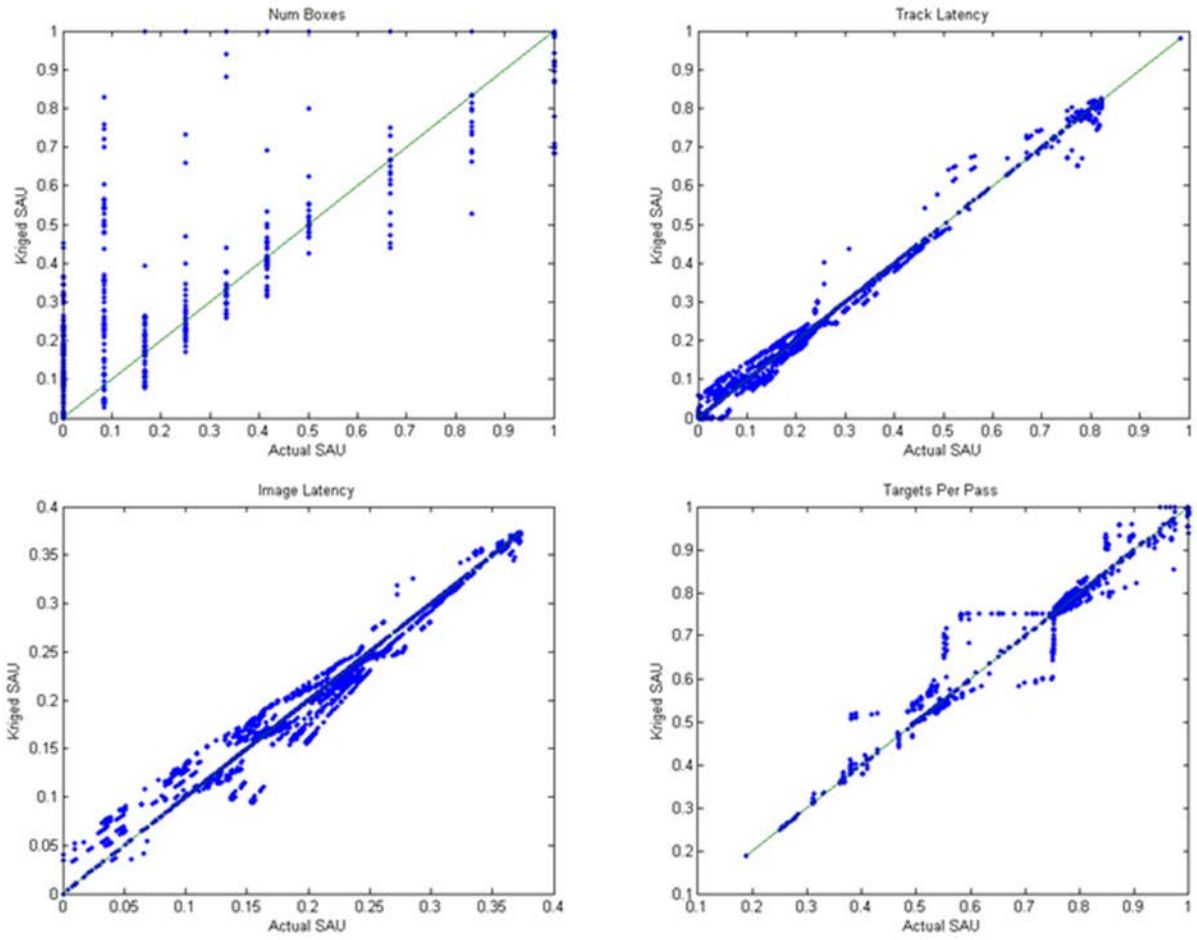


**Figure 10. Correlation between Kriged SAU and actual SAU for Kriged attributes.**

overestimates for higher (and less frequent) values in the lower right figure. Here the large scattering about the actual value is due to the small Kriging errors being propagated through the logarithmic transformation.

In many tradespace studies, multi-attribute utility (MAU) is used to evaluate design options[19,20]. For this study, MAU value is calculated as a weighted sum, according to stakeholder preferences, of single attribute utility (SAU) values. In its most general form, the MAU function is multilinear. In cases where substitution or complementary effects exist, a multiplicative nature is observed. In the case studies explored in this paper, where each attribute contributes independently to utility, the MAU function reduces to a linear sum of SAU values. The SAU is calculated using the attribute value and a monotonic utility function that maps between the attribute and [0, 1]. The MAU, once calculated, allows designers to evaluate designs on a single dimension as opposed to one dimension for each attribute. Figure 10 shows how the errors in each of the Kriged attributes translated to SAU, and Fig. 13 shows how this translated to MAU for SRS.

Unsurprisingly, the SAU values for *image latency* and *track latency* tended to be very accurate. *Image latency* SAU values rarely exceeded 0.05 utiles. *Track latency* values behaved similarly, with occasional larger (0.05-0.15 utiles). The behavior seen in the *number of boxes* (top left) correlation chart is striated because of the discrete nature of the attribute. While other attributes (namely, *targets per pass*) also had discrete ranges, in *number of boxes* the [0, 1] SAU range comes maps to a domain of [1, 10] boxes, explaining 10 values of actual SAU seen in the chart. Looking back to Fig. 8, the Kriging estimates tended to be high for low values of the actual attribute (*number of boxes*). Figure 11 shows the relationship between Kriged values and actual values only on the [0, 1] SAU range. Underestimates were much less common than overestimates for *number of boxes* for the same reason that *targets per pass* tended to underestimate on the high end of the range: Kriging behaves poorly near the limits of the range.

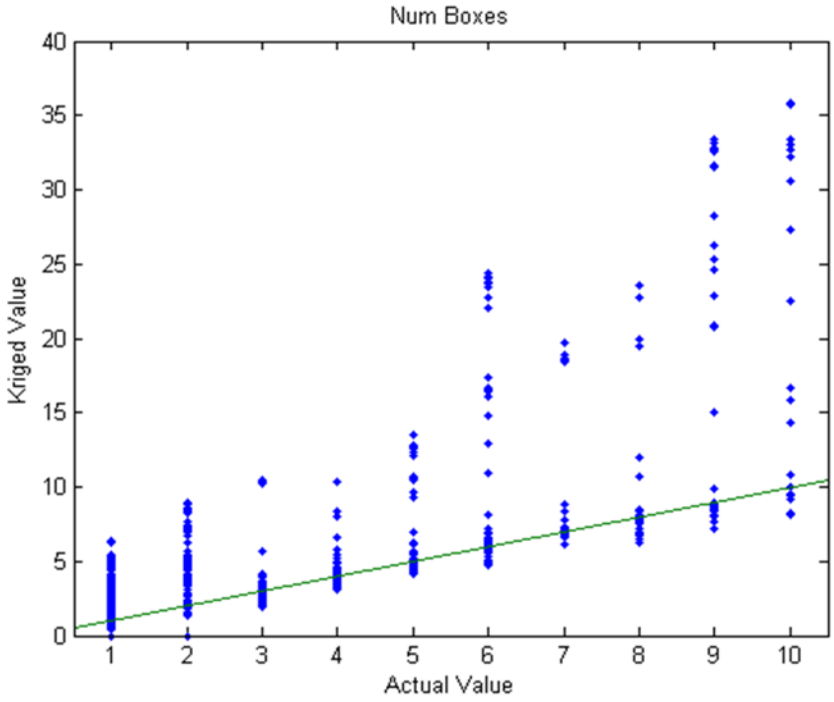


**Figure 11. Kriged values vs. actual values for *number of boxes*, zoomed to [0, 1] SAU range from Fig. 8.**

The correlation pattern seen to the left and below the point (0.75, 0.75) on the *targets per pass* (bottom right of Fig. 10) chart show what happens when there is a significant cusp in the SAU curve. Because the attribute values for *targets per pass* span many orders of magnitude, the piecewise SAU curve, seen in Fig. 12, has discontinuities spaced logarithmically rather than linearly. The horizontal line ending at (0.75, 0.75) on Fig. 10 corresponds to Kriging overestimates that went above the cusp (1,000 targets), whereas the vertical line ending at the same point corresponds to underestimates that went below the cusp.

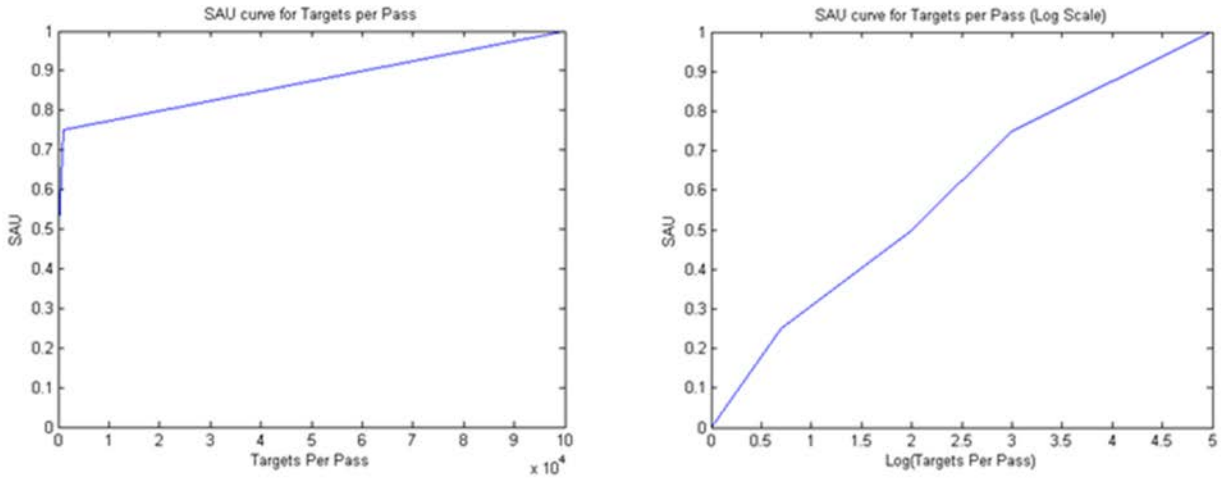


**Figure 12. SAU curve (normal and log scale) for *targets per pass*.**

The MAU values are the final layer in applying ETAM to the SRS case study. ETAM does not directly approximate utility. Rather, utility is calculated from attributes generated in ETAM. Utility is being considered here to show how errors in ETAM can propagate through method of tradespace comparison. Kriging produced attribute values, which were mapped to SAU values through SAU curves, and now the SAU values are mapped to MAU using the multiplicative (as opposed to additive) MAU equation, calculated as

$$JU(A)+1=\prod_{h}^{H}[J\cdot j_h u_h(A_h)+1]$$

where $A$ is the vector of $H$ attributes, $U(A)$ is the MAU function, and $u_h(A_h)$ is the $h^{th}$ SAU function[19]. The small j value, $j_h$, is $U(A)$ when $u_h(A_h) = 1$ and $u_i(A_i) = 0$ (for all $i \neq h$). Both the small k and small u values are a function of epoch. In the case where small $k$s add up to 1, as they do in the SRS case study, the MAU function reduces to a linear sum with the small $k$s as the weights. Big $J$, a normalization constant for a given set of small $j$s, is the solution to

$$J+1=\prod_{h}^{H}[J\cdot j_h+1]$$

This means the results of Kriging have now been passed through two filters exogenous to ETAM. Since the SAU curves and preference sets play a large role in determining the MAU, one must be careful in drawing conclusions about the success of ETAM based on stats concerning MAU (and SAU for that matter). For the specific set of SAU curves and preference sets in the each epoch, Fig. 13 shows how the Kriged MAU values corresponded to actual MAU values for SRS. One of the main reasons Kriging overestimated MAU more than it underestimated MAU was the fact that *number of boxes* was heavily overestimated and happened to be one of the highest weighted attributes (highest small k).

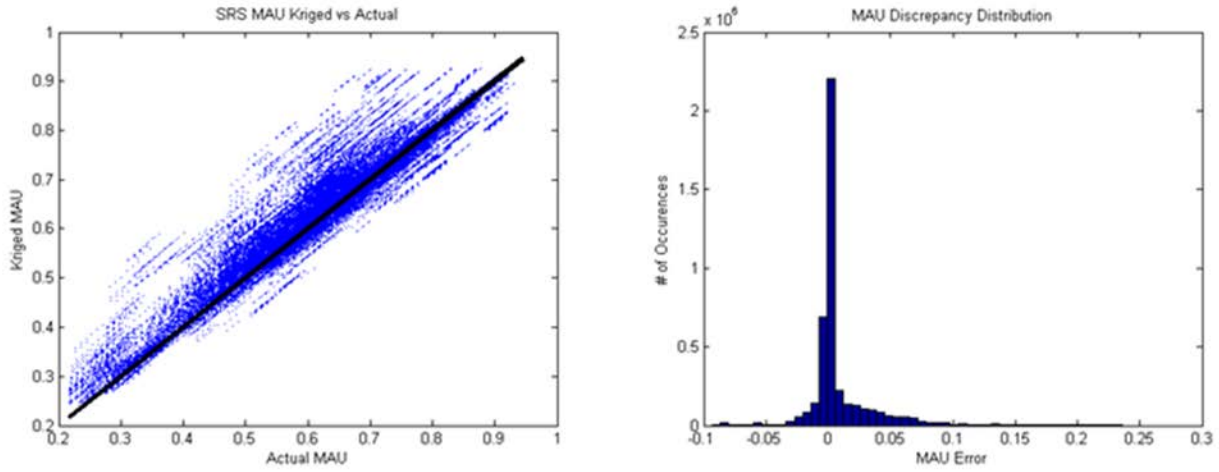


**Figure 13. Correlation between Kriged MAU and actual MAU for Satellite Radar System.**

As mentioned above, analysis of SAU and MAU data must be done carefully to account for the fact that SAU curves and MAU preference sets have an impact. Statistics like percent error and average error can be misleading. For instance, a 3% error for an actual MAU of 0.1 is much different than a 3% error when the actual MAU is 1. Similarly, an error of 0.01 utiles might mean less near the middle of the utility range whereas if it saturates a MAU to 1 it means much more. While percent error and average error might be misleading statistics for SAU and MAU data, a metric that looks only at ordering is more appropriate. The Spearman's Rank Correlation Coefficient[21] was used to characterize the accuracy of DOE-Kriging on the individual SAU values and MAU. The Spearman's rank correlation coefficient measures the correlation between the ranks of a value in two different sets, in this case a design's actual MAU/SAU value and its Kriged MAU/SAU value. For each epoch, a Spearman's coefficient was calculated for the four Kriged attributes' SAU values as well as the MAU. The values were then averaged over all epochs and can be seen in Table 8. Little variance was seen in Spearman coefficients across epochs. The insight gained from the relatively high Spearman's coefficients is that little accuracy is lost with respect to relative performance (a Spearman's coefficient of 1 implies rank is completely preserved). The significance of this test was very strong due to the extremely large sample size.

**Table 8. Spearman's Rank Correlation Coefficients for SAU and MAU values**

| Value | SAU(Number of Boxes) | SAU(Track Latency) | SAU(Image Latency) | SAU(Targets per Pass) | MAU |
|---|---|---|---|---|---|
| Spearman's Coefficient | 0.8705 | 0.9959 | 0.9574 | 0.9738 | 0.9720 |

## IV. Application to Space Tug

The Space Tug data set contains designs for orbital transfer vehicles that can be used for a variety of on-orbit servicing missions such as observation of (potentially hostile) targets, assisting in orbit changes, and removing debris[22]. This data set has been used for studies in changeability and survivability[23,24,25]. Recent work has added context variables to the existing preference curves used to define the epochs[26].

### A. Variable Handling

For Space Tug, the design variables originally introduced are *manipulator mass*, *propulsion system*, and *fuel mass*. An additional variable, design for evolvability (*DfE*), was added for the purposes of this simulation. The *DfE* variable represents the inclusion of design heuristics that make redesign simpler and is treated as a mass penalty. The ranges of values for the design variables are seen in Fig. 11. All of the design variables except for *propulsion system* will be considered Krigable variables. The product of the levels of all Krigable variables for Space Tug is 96.

**Table 9. Space Tug design variable levels[22,26] (Krigable variables are italicized)**

| Design Variable | Scale Type | Valid Range | Enumerated Levels | # Levels |
|---|---|---|---|---|
| *Manipulator Mass* | *Ratio* | *300 – 5000 [kg]* | *300, 1000, 3000, 5000 [kg]* | *4* |
| Propulsion System | Nominal | 1 – 4 [int] | Storable BiPropellant, Cryogenic, Electric, Nuclear | 4 |
| *Fuel Mass* | *Ratio* | *300 – 30000 [kg]* | *30, 100, 300, 600, 1200, 3000, 10000, 30000 [kg]* | *8* |
| *DfE (Mass Penalty)* | *Ratio* | *0 – 20 [%]* | *0, 10, 20 [%]* | *3* |

Only one of the Space Tug epoch variables, *technology level*, will be used for testing ETAM. Technology is either "present level" or "future level" and affects the attribute levels associated with each design variable as well as the cost calculation. The attribute levels are then used to calculate utility. Since *technology level* is a nominal variable, it will be treated as non-Krigable.

The three attributes originally calculated were *capability*, *delta V*, and *response time*. *Capability* is measured as the manipulator mass. *Delta V* is a function of all masses, specific impulse, and mass fraction. The latter two are properties of the *propulsion system* in use, as seen in Table 10. In cases where two values appear, the latter value is used in the future context. *Response time* is either fast or slow and is a function solely of the propulsion and is in the 'Fast?' column of Table 10.

**Table 10. Propulsion system associated constants[22]**

| Propulsion System | $I_{sp}$ (sec) | Base Mass (kg) | Mass Fraction | Fast? |
|---|---|---|---|---|
| Storable Bipropellant | 300 | 0 | 0.12 | Y |
| Cryo | 450/550 | 0 | 0.13 | Y |
| Electric | 3000 | 25 | 0.25/0.3 | N |
| Nuclear | 1500 | 1000/600 | 0.20 | Y |

The *cost* of a design is a function of its dry and wet mass. First, the propulsion system mass ($M_p$) must be calculated using base mass ($m_{p0}$), mass fraction ($m_{pf}$), and the fuel mass ($M_f$)

$$M_p = m_{p0} + m_{pf} \cdot M_f$$

Next, the vehicle bus mass ($M_b$) is calculated using the propulsion system mass, manipulator mass ($M_m$), and bus mass fraction ($m_{bf}$)

$$M_b = M_p + m_{bf} \cdot M_m$$

The bus mass fraction was held constant at 1 for this case study. The mass penalty for DfE is levied in the final mass calculations. The vehicle dry mass ($M_d$) and vehicle wet mass ($M_w$) then are calculated as

$$M_d = (1 + DfE) \cdot (M_b + M_m)$$

$$M_w = M_d + M_f$$

The dry mass cost ($c_d$) of \$150,000/kg and the wet mass cost ($c_w$) of \$15,000/kg in the present context (\$10,000/kg in the future context) is used to calculate cost ($C$). These cost values were adapted from Ref. 22. The future context wet mass cost was proposed by MIT student Matt Fitzgerald.

$$C = c_w \cdot M_w + c_d \cdot M_d$$

*Delta V* is calculated as

$$\Delta v = I_{sp} \ln\left(\frac{M_d + M_w}{M_d}\right)$$

Because response time is a function of only a non-Krigable variable, it will not be considered in this application of ETAM. Using the attribute functions, the binary DMM can be created. The Binary DMM, shown in Table 11, reveals that both attributes are a function of each of the Krigable variables, therefore only one training set will need to be used.

**Table 11. Binary DMM for Space Tug**

| | | **Krigable Variables** | | |
|---|---|---|---|---|
| | | **Manipulator Mass** | **Fuel Mass** | **DfE** |
| **Attributes** | **Cost** | X | X | X |
| | **Delta V** | X | X | X |

A Box-Benkhen design cannot be used for Space Tug without creating new variable levels. Rather than add a *manipulator mass* and *fuel mass*, this situation was used as an opportunity to add in a new DOE method. The DOE method used for this application of ETAM to Space Tug was D-Optimal designs. D-Optimal designs are computer generated experimental designs that are helpful when the variable space is irregular, as is the case with Space Tug's Krigable variables[27]. A D-Optimal design is selected from a set of candidate designs by choosing the design with the largest determinant. The determinant D is calculated as

$$D = \left| X^T X \right|$$

where $X$ is the vector of indices of the different variable levels. There is an element of stochasticity added into the model since there is not explicitly a single design that is D-Optimal. Since the D-Optimal design is generated each time the ETAM is run, different iterations of ETAM will use different D-Optimal experimental designs. There is not always a single combination of treatments that meet the condition for D-optimality. An added criterion of the training set, informed by the findings of ETAM application to SRS, was the inclusion of all eight corner points. Rather than prescribe a training set size a priori, the small size of the Space Tug tradespace was leveraged to study the effect training set size has on accuracy.

### B. Results

Since a fixed training set was not used in the Space Tug application, one chart alone will not capture the results. The first table, Table 12, shows the results with a ratio of training points to Kriged points of 1:2 over 50 full executions of ETAM.

**Table 12. Accuracy of Kriged variables (32 training points) over 50 full executions of ETAM**

| Attribute | Mean (Training Set Points) | E[\|Difference\|] | Std[\|Difference\|] | Average Error | Spearman Rho |
|---|---|---|---|---|---|
| Cost | $976,640,000 | $3,722,500 | $5,368,500 | 0.55% | 0.9999 |
| Delta V | 5.2437 km/s | 0.898 km/s | 1.6964 km/s | 0.29% | 0.9617 |

A first order examination shows that ETAM was accurate within a percent for both cost and delta V. The Spearman rho values, which represent the rank correlation coefficient, reveal that order is very well preserved in cost and less so in delta V. This relationship, for a single execution of ETAM, can be seen in Fig. 14.

The Spearman rho for cost is nearly one, and tended to be so for any number of training points. The Spearman rho values for delta V, on the other hand, were much more sensitive to the size of the training set, as seen in Fig. 15.

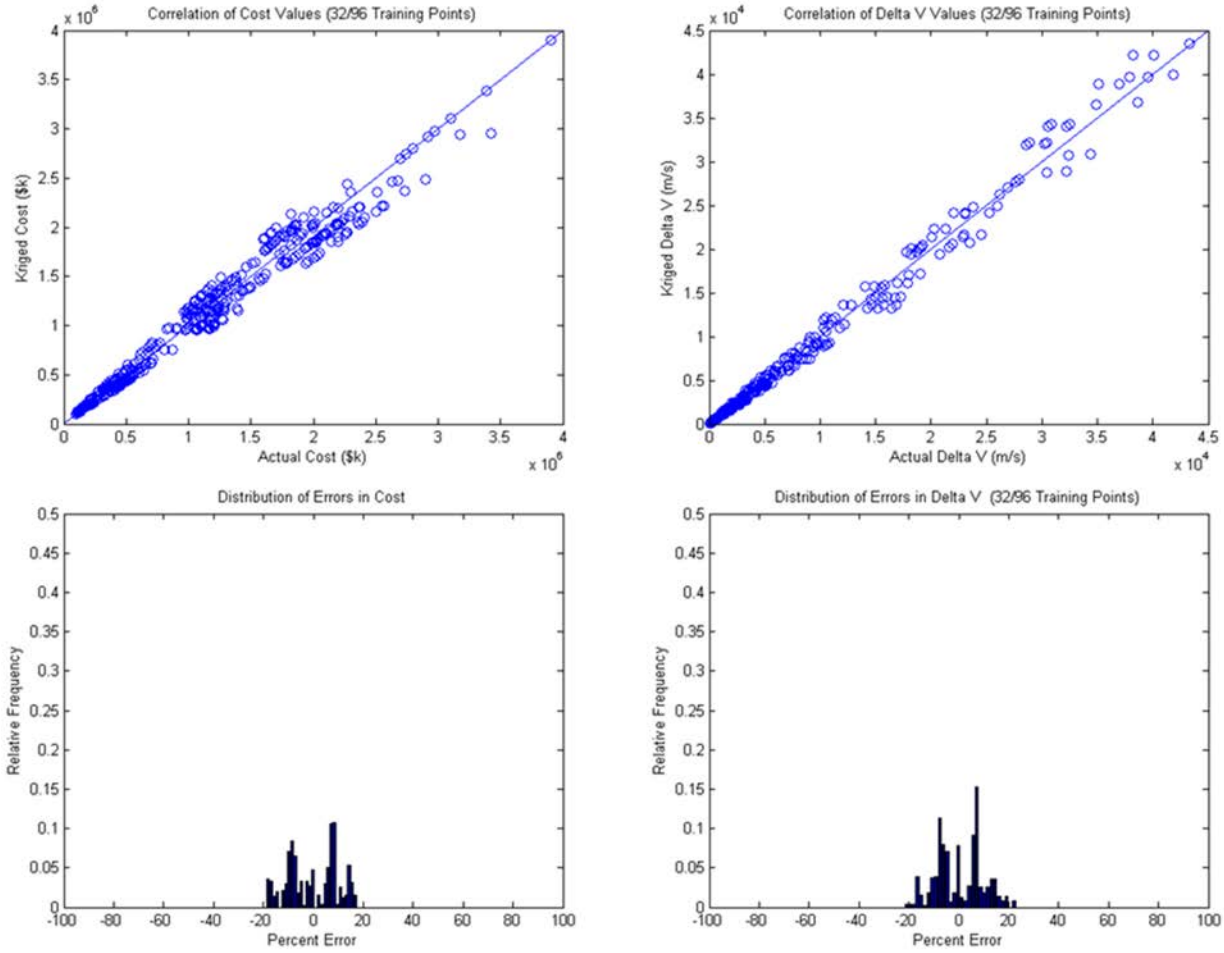


**Figure 14. *Cost* and *delta V* correlations and error distributions for 32 training points**

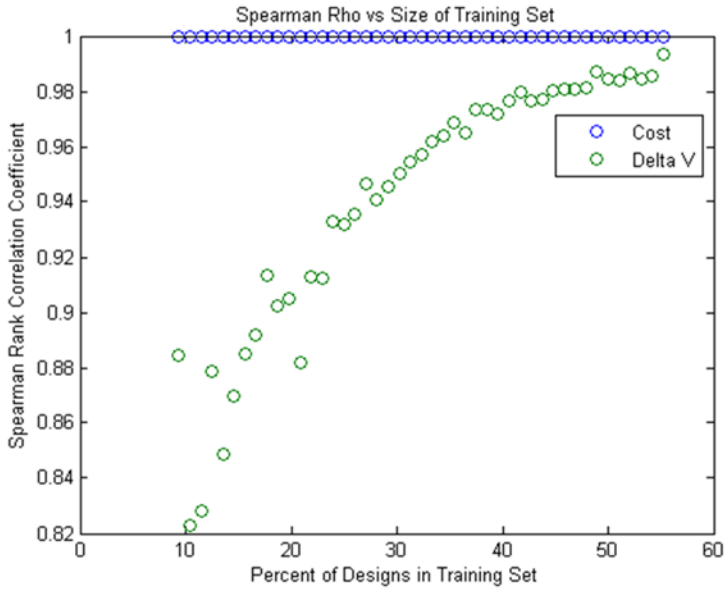


**Figure 15. Spearman Rho vs. training set size (percent of Krigable subset, N = 96)**

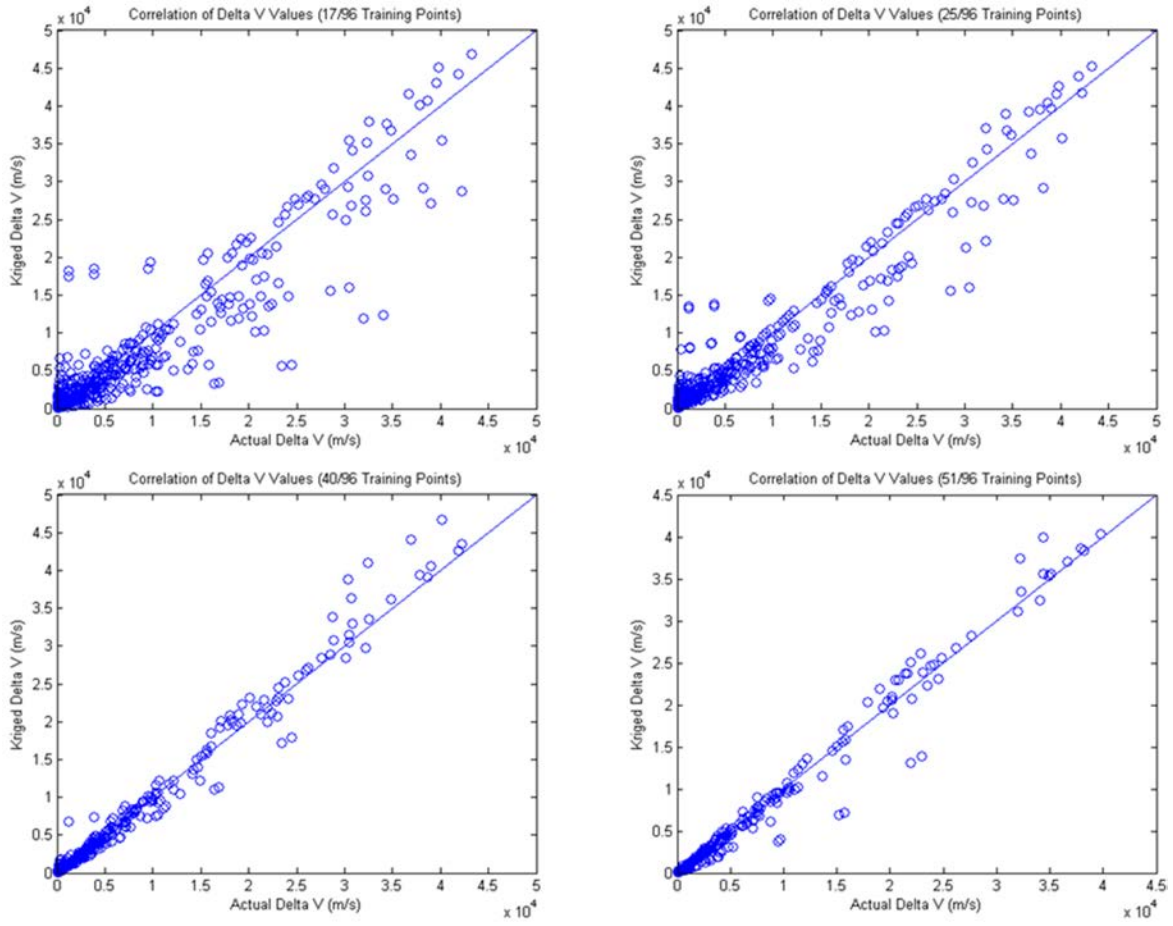


**Figure 16. Correlation of Kriged vs. actual *delta V* values for different training set sizes (Clockwise from top left: 17, 25, 40, 51).**

The values in Fig. 15 are averages of several (3-30 depending on the number of times a specific training set occurred) ETAM trials for a specific training set size. Sample delta V correlations for increasing training set size can be seen in Fig. 16. The accompanying error distributions follow in Fig. 17. As the training set increases in size, the correlation becomes stronger.

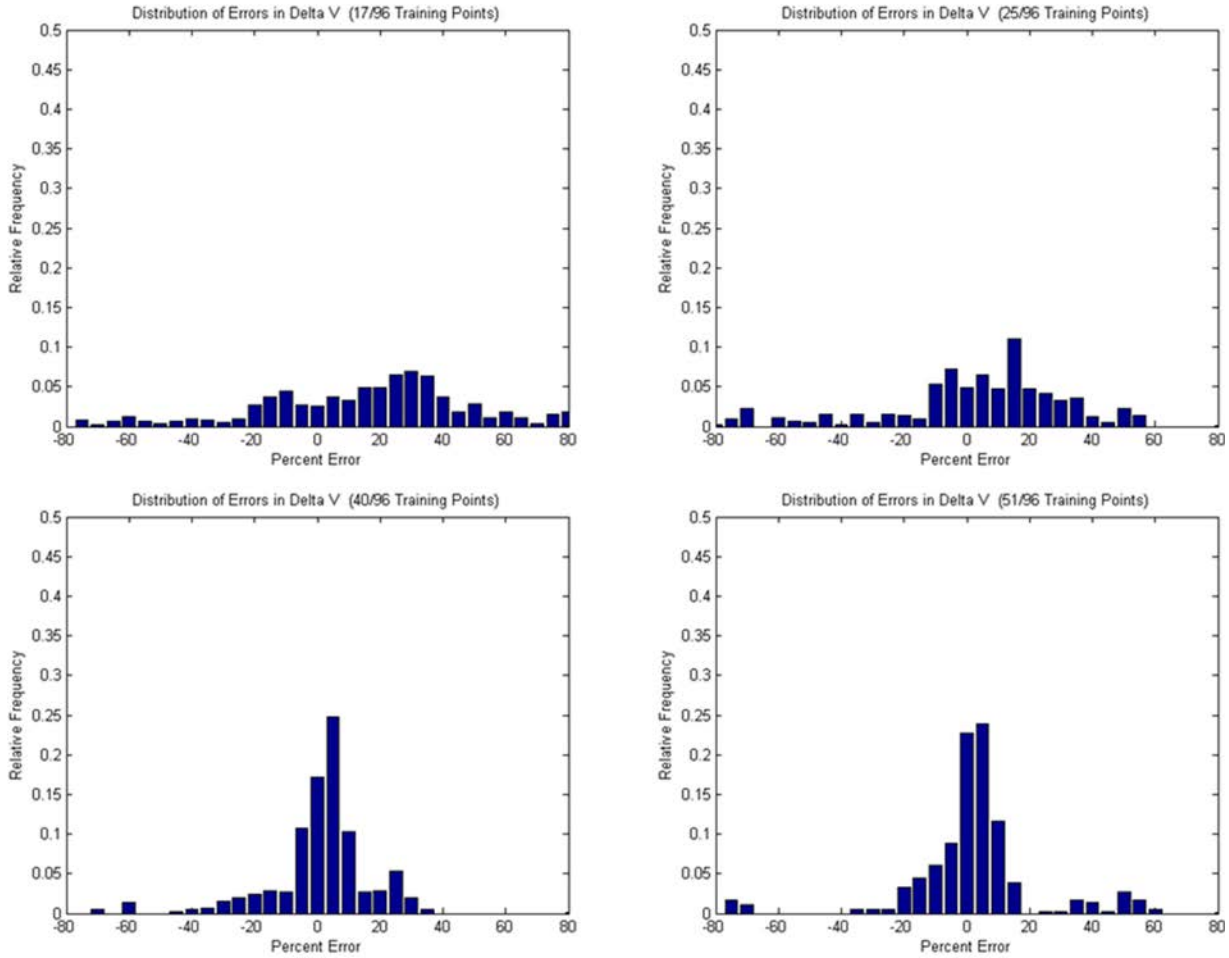


**Figure 17. Error distribution in *delta V* for different training set sizes (Clockwise from top left: 17, 25, 40, 51).**

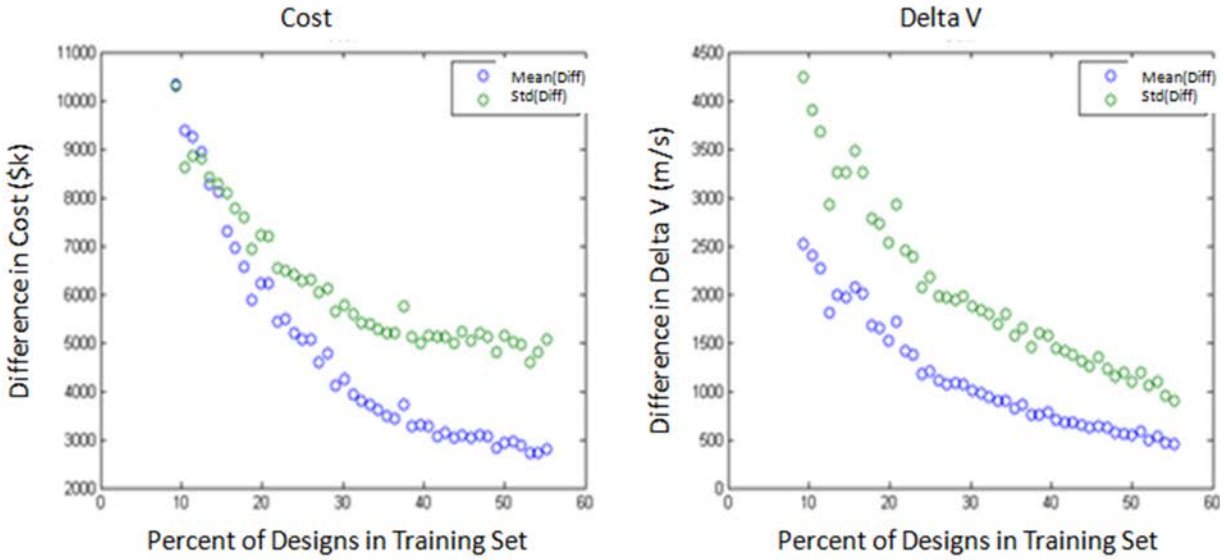


**Figure 18. Mean and standard deviation of difference between actual and Kriged attributes, Space Tug.**

The values for mean error and standard deviation of error also improved as the size of the training set increased, as seen in Fig. 18. Note that the absolute error is being used; the data does not suggest a bias.

The percent error metric allows the differences between actual and Kriged values to be normalized by the actual value of the attribute. The relationship between percent error and the size of the training set is seen in Fig. 19. While percent error improves as the training set increases for both attributes, the improvement is greater for cost.

## V. Discussion

Overall, ETAM was very successful with respect to the research goals. In the SRS study, ETAM proved its ability to generate data for a very large tradespace. In the Space Tug case study, the results were not only very accurate, but the use of D-Optimal DOE demonstrated the substitutability of the DOE module in ETAM. Any valid experimental design can be used, but including the corners of the space in any design will improve results. In SRS, ETAM generated data much faster than the performance model, but at the expense of accuracy. In applying ETAM to Space Tug, the smaller size of the tradespace allowed for the relationship between computational savings (via the training set size) and accuracy to be further explored. Since accuracy increases with training size, a tradespace explorer with a set schedule could maximize accuracy by choosing the maximum training set size that does not exceed that schedule.

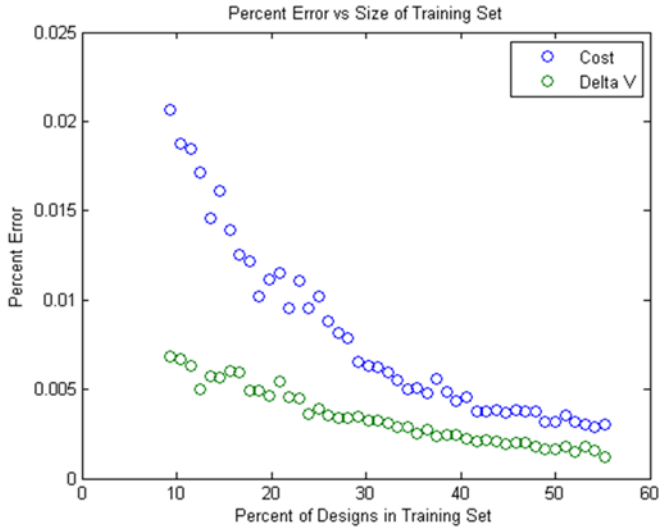


**Figure 19. Percent error in attributes vs. size of training set.**

### A. Considerations for Applying ETAM

1. *Limitations*

The value of ETAM is limited by the variable composition of the tradespace in question. The nature of a tradespace determines the effectiveness of ETAM in terms of time savings. If most of the variables are nominal or ordinal, the limited time savings might not overcome the overhead cost of tailoring the ETAM to handle the tradespace. If most of the variables are ratio or interval types, then the time savings can be quite considerable.

A second limitation of ETAM is that it can only approximate attributes contained within the ranges of values in the training set. If the training set only contains values for Krigable variable *X* between *y* and *z*, the approximation for a point where $X = z + 1$ will be inaccurate because of the nature of Kriging. This can be accounted for by making sure the training set contains the corners (designs where all Krigable variables had a level on either end of their defined range). As the number of levels increases, it can potentially be feasible to include all points on the surface of the *n*-dimensional hypercube (where *n* is the number of Krigable variables. Doing so eliminates interpolation errors that occur due to only having training points on one side of an interpolated point, but can drastically increase the relative size of the training set when the Krigable variables have few levels. This phenomenon can be seen for 2 Krigable variables in Fig. 20. This does not consider additional points that would be in the training set as selected by DOE.

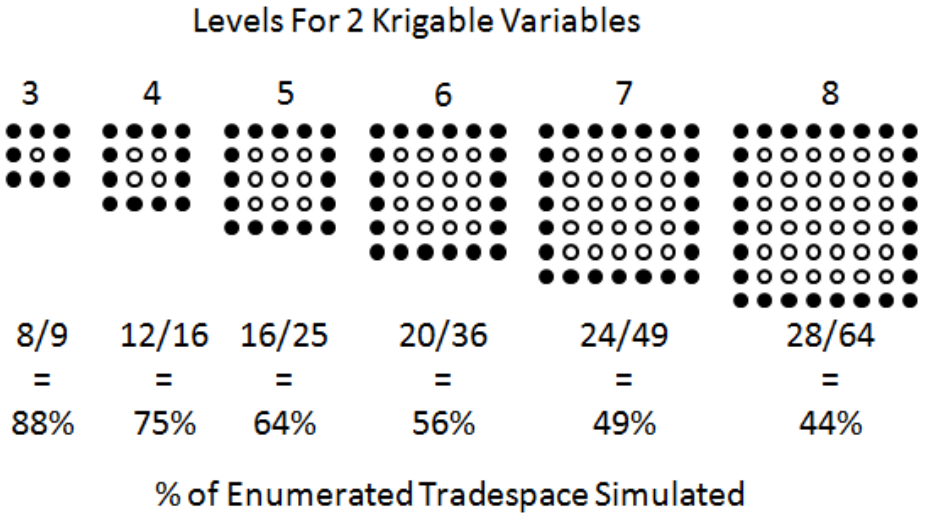


**Figure 20. Training set implications stemming from including the points on the surface of an *n*-dimensional hyper cube (*n* = 2 shown).**

2. *Potential Savings*

The computational time savings from ETAM are based on several variables, some of which might not be known before deciding whether to pursue the use of ETAM. These variables are: the number of design-epoch pairs in the full tradespace (*n*), the execution time of the performance model for a single design-epoch pair (*t*), the ratio of training set to Krigable subset (*k*), and the upfront time needed to integrate the tradespace into the ETAM model (E). The time savings of ETAM, as opposed to full-factorial tradespace simulation, expressed as time dilation factor (TDF), are

$$TDF = \left( \frac{E + k \cdot n \cdot t}{n \cdot t} \right) \cdot 100$$

A TDF of 100% means using ETAM takes the same amount of time as simulating the full tradespace. A TDF below 100% means time was saved by using ETAM and a TDF greater than 100% means ETAM takes longer than simulating the full tradespace would have taken. As the time needed to simulate the full tradespace becomes much greater than the anticipated ETAM setup time ($nt >> E$), the TDF approaches 100**k*. When $E \approx nt$, the TDF is approximately -100*(1+*k*). When $E >> nt$, the TDF approaches 100*(*E*/*nt*). As a general rule, ETAM should only be used if $nt >> E$.

**Table 13. Time Dilation Factor for Different Case Studies**

| Case Study | *E* | *k* | *n* | *t* | Time Dilation Factor |
|---|---|---|---|---|---|
| SRS | 8 hours | 13/81 | 6,718,464 | ≈26s | 16.07% |
| Space Tug | 2 hours | 32/96 | 384 | 0.0015s | 1,250,000% |

The advantage from ETAM in each case study can be seen in Table 13. The *t* value for SRS was estimated based on anecdotal evidence from the original simulation operators based on the following values: 23,778 designs, 972 epochs, ≈10 days to simulate. The very poor time dilation factor for the Space Tug case study is expected since the full tradespace was very small and the performance model was very simple; Space Tug was used to demonstrate the accuracy of ETAM and not to showcase its effectiveness in saving time.

3. *Potential Costs*

The computational savings of ETAM must be weighed against the costs incurred in accuracy. ETAM displayed varying levels of accuracy for the four Kriged attributes in SRS (as seen in Table 7, Fig. 8, Fig. 9, and Table 8) when only simulating 16% of the enumerated tradespace. For the Space Tug case study, it was clearly demonstrated that the accuracy increases with the percent of designs simulated (as seen in Fig. 15, Fig. 16, Fig. 17, Fig. 18, and Fig. 19). Ultimately, it is up to the tradespace explorer to determine the amount of potential accuracy they are willing to sacrifice for computational savings. At this point there is no way to approximate accuracy losses, but application to further case studies might lead towards more insights concerning accuracy losses.

## VI. Conclusions

As tradespace networks grow to become very large, the time needed to simulate the points in these tradespace networks can become prohibitive. In response to this challenge, the ETAM was developed and, through application to two case studies, it was shown that there is indeed a way to make exploring large tradespaces take less time. Intelligent subsampling and interpolation were combined to approximate data that might otherwise take prohibitively long to simulate. The maximum amount of time saved using ETAM is determined by the types of variables in the data set and the number of levels in the Krigable variables (ratio and interval variable types). A tradeoff between time savings and accuracy was demonstrated in the application to the Space Tug case study, but more case studies are needed to quantify this relationship.

## Acknowledgments

The authors gratefully acknowledge funding for this research provided through MIT Systems Engineering Advancement Research Initiative (SEAri, http://seari.mit.edu) and its sponsors.

## References

[1] Forrester, A.I.J., Sobester, A., and Keane, A.J., *Engineering Design via Surrogate Modelling: A Practical Guide*, Progress in Astronautics and Aeronautics, AIAA, Reston, VA, 2008, pp. 1-205.

[2] Howell, D., Uebelhart, S.A., Angeli, G., Miller, D.W., "Large Tradespace Analysis Using Parametric Integrated Models of Extremely Large Ground-based Telescopes," *6th World Congress on Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brazil, June 2005.

[3] Kichkaylo, T., Hoag, L., Lennon, E., Roesler, G., "Highly Efficient Exploration of Large Design Spaces: Fractionated Satellites as an Example of Adaptable Systems," *Proceedings of the 10th Conference on Systems Engineering Research*, St. Louis, MO, March 2012.

[4] Balabanov, V.O., Charpentier, C., Ghosh, D., Quin, G., Vanderplaats, G.N., "VisualDOC: A Software System for General-Purpose Integration and Design Optimization," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, Sept. 2002.

[5] Murray, B.T. et al., "META II: Complex Systems Design and Analysis (CODA)," *United Technologies Research Center*, AFRL-RZ-WP-TR-2011-2102, August 2011.

[6] Stump, G., Lego, S., Yukish, M., Simpson, T.W., and Donndelinger, J.A., "Visual Steering Commands for Trade Space Exploration: User-Guided Sampling with Example," *Journal of Computer and Information Science in Engineering*, Vol. 9, No. 4, December 2009, pp. 1-10.

[7] Zhang, X., Simpson, T., Frecker, M., and Lesieutre, G., "Supporting Knowledge Exploration and Discovery in Multi-dimensional Data with Interactive Multiscale Visualization," *Journal of Engineering Design*, Vol. 23, No. 1, 2012, pp. 23-47.

[8] Simpson, T.W., Mauery, T.M., Korte, J.J., and Mistree, F., "Kriging Models for Global Approximation in Simulation-based Multidisciplinary Design Optimization," *AIAA Journal*, Vol. 39, No. 12, December 2001, pp. 2233-2241.
[9] Martin, J.D., Simpson, T.W., "On the Use of Kriging models to Approximate Deterministic Computer Models," *Proceedings of ASME Computers and Information in Engineering Conference 2004*, Salt Lake City, UT, Sept. 2004.
[10] Kleijnen, J.P.C., "Kriging Metamodeling in Simulation: A Review," *European Journal of Operational Research*, Vol. 192, No. 3, February 2009, pp. 707-716.
[11] Ross, A.M., McManus, H.L., Rhodes, D.H., Hastings, D.E., and Long, A.M., "Responsive Systems Comparison Method: Dynamic Insights into Designing a Satellite Radar System," *AIAA Space 2009*, Pasadena, CA, September 2009.
[12] Siegel, S. "Nonparametric Statistics," *The American Statistician*, Vol. 11, No. 3, 1957, pp. 13-19.
[13] Box, G.E.P., and Benkhen, D. W., "Some New Three Level Designs for the Study of Quantitative Variables". *Technometrics*, Vol. 2, 1960, pp. 455-475.
[14] Danilovic, M. and Browning, T.R., "Managing Complex Product Development Projects with Design Structure Matrices and Domain Mapping Matrices," *International Journal of Project Management*, Vol. 25, No. 3, April 2007, pp. 300-314.
[15] Papalambros, P. Y., and Wilde, D. J., *Principles of Optimal Design*, 2nd ed., Cambridge University Press, Cambridge, 2000, pp. 54-57.
[16] Chiles, J.P., and Delfiner, P., *Geostatistics: Modeling Spatial Uncertainty*, Wiley Interscience, New York, 1999, pp. 150-192.
[17] Fulcoly, D.O., *A Normative Approach to Designing for Evolvability: Methods and Metrics for Considering Evolvability in Systems Engineering*, Master of Science Thesis, Aeronautics and Astronautics, MIT, Cambridge, MA, June 2012.
[18] Press, W.H., Teukolsky, S.A, Vetterling, W.T., and Flannery, B.P., *Numerical Recipes: The Art of Scientific Computing*, 3rd ed., Cambridge University Press, Cambridge, 2007, pp. 135-147.
[19] Keeney, R. L., and Raiffa, H., *Decisions with Multiple Objectives—Preferences and Value Tradeoffs*, Cambridge University Press, Cambridge, 1993, pp. 288-291.
[20] Ross, A.M., *Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration*, Ph.D. Dissertation, Engineering Systems Division, MIT, Cambridge, MA, June 2006.
[21] Spaulding, T.J., *Tools for Evolutionary Acquisition: A Study of Multi-Attribute Tradespace Exploration (MATE) Applied to the Space Based Radar (SBR)*, Master of Science Thesis, Aeronautics and Astronautics, MIT, Cambridge, MA, June 2003.
[22] McManus H., Schuman T., "Understanding the Orbital Transfer Vehicle Trade Space," *AIAA Space 2003*, Long Beach, CA, September 2003.
[23] Ross A.M., Hastings D.E., "Assessing Changeability in Aerospace Systems Architecting and Design Using Dynamic Multi-Attribute Tradespace Exploration," *AIAA Space 2006*, San Jose, CA, September 2006.
[24] Richards M.G., *Multi-attribute Tradespace Exploration for Survivability*, Ph.D., Engineering Systems Division, MIT, Cambridge, MA, June 2009.
[25] Fitzgerald, M.E., Ross, A.M., and Rhodes, D.H., "Assessing Uncertain Benefits: a Valuation Approach for Strategic Changeability (VASC)," *INCOSE International Symposium 2012*, Rome, Italy, July 2012.
[26] Fulcoly, D.O., Ross, A.M., and Rhodes, D.H., "Evaluating System Change Options and Timing using the Epoch Syncopation Framework," *Proceedings of the 10th Conference on Systems Engineering Research*, St. Louis, MO, March 2012.
[27] *NIST/SEMATECH e-Handbook of Statistical Methods*, http://www.itl.nist.gov/div898/handbook/, 2003, accessed 30 March 2012.